The accuracy mirage in the estimation of global irrigation water withdrawals

R code

Arnald Puy

Contents

1 Presentation

This document presents the workflow of the paper "The accuracy mirage in the estimation of global irrigation water withdrawals." The abstract is the following:

We review the main uncertainties in the calculation of irrigation water withdrawals by large-scale models, which are widely used by regulatory agencies to inform global policies on agriculture and climate change. We show that the apparent precision characterizing their outputted values results from neglecting several important ambiguities and is therefore to be reassessed. If a global uncertainty and sensitivity analysis is performed, their estimates rapidly turn into intervals that span several orders of magnitude. This review stresses the importance of properly exploring the ambiguities involved in the estimation of global agricultural water demands, and urges for a thorough debate where the trade-offs between computational complexity, robustness and accuracy are thoroughly assessed.

Global irrigation water withdrawals are usually estimated by slightly varying the following Equation, which is computed at the grid cell level:

$$y = \frac{I_a(ET_c - P)}{E_p}\,, \tag{1}$$

where y is a scalar representing irrigation water withdrawals, I_a the irrigated area, ET_c the crop evapotranspiration, P the precipitation and E_p the irrigation efficiency.

Once all datasets required for the analysis have been acquired from the appropriate sources, the results of the paper should be fully reproducible with the code below in any personal computer. Questions about the code or the computational design should be addressed to A. Puy (apuy@princeton.edu, or arnald.puy@pm.me).

2 Preliminary functions

We start by creating a function to load all required libraries for the analysis in one go, and a custom theme for the figures. We also set a checkpoint to ensure that our code is fully reproducible for anyone anytime.

```
# PRELIMINARY STEPS ------------------------------------------------------------

# Install and load 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(
  "data.table", "tidyverse", "sensobol", "wesanderson",
  "triangle", "scales", "cowplot", "fitdistrplus",
  "parallel", "doParallel", "foreach", "sp", "sf", "Rfast",
  "raster", "rworldmap", "countrycode"))

# Custom theme
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),
          legend.position = "top",
          strip.background = element_rect(fill = "white"))
}

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

checkpoint("2021-02-22",
           R.version ="4.0.3",
           checkpointLocation = getwd())
```

3 Irrigated area (I_a)

Here we plot the differences in irrigated area at the country level reported by the FAO-GMIA, the IWMI-GMIA, the GRIPC, Aquastat and FAOSTAT. The dataset `meier.csv` includes the data produced by Meier, Zabel, and Mauser (2018), which can be found in its Supplementary Materials file.

```
# PLOT UNCERTAINTY IN IRRIGATED AREAS ----------------------------------------------

# Read data compiled by Meier
irrigated.areas <- fread("meier.csv")

# Arrange and drop Oceania and O's
dt <- melt(irrigated.areas, measure.vars = 4:9) %>%
  .[!Continent == "Oceania"] %>%
  .[!value == 0]

# List to plot
continent_list <- list(c("Africa", "Americas"), c("Asia", "Europe"))

# Plot
gg <- list()
for (i in 1:length(continent_list)) {
  gg[[i]] <- ggplot(dt[Continent %in% continent_list[[i]]],
                    aes(reorder(Country, value), value)) +
    geom_point(stat = "identity", aes(color = variable)) +
    scale_y_log10(
      breaks = trans_breaks("log10", function(x) 10^x),
      labels = trans_format("log10", math_format(10^.x))
    ) +
    coord_flip() +
    scale_color_manual(
      name = "Dataset",
      values = c(
        "yellowgreen", "seagreen4", "magenta3",
        "sienna3", "turquoise2", "khaki3"
      )
    ) +
    labs(
      x = "",
      y = "Irrigated area (ha)"
    ) +
    facet_wrap(~Continent, scales = "free_y") +
    theme_AP()
}
```

```
gg[[1]]
```

Figure 1: Irrigated area estimates produced for Africa and the Americas by the FAO-GMIA (Siebert et al. 2013), the IWMI–GIAM (Thenkabail et al. 2009), the GRIPC (Salmon et al. 2015), Meier's map (Meier, Zabel, and Mauser 2018), Aquastat (FAO 2016) and FAOSTAT (FAO 2017). The data has been retrieved from Meier, Zabel, and Mauser (2018)

```
gg[[2]]
```

Figure 2: Irrigated area estimates produced for Asia and Europe by the FAO-GMIA (Siebert et al. 2013), the IWMI–GIAM (Thenkabail et al. 2009), the GRIPC (Salmon et al. 2015), Meier's map (Meier, Zabel, and Mauser 2018), Aquastat (FAO 2016) and FAOSTAT (FAO 2017). The data has been retrieved from Meier, Zabel, and Mauser (2018)

4 The crop evapotranspiration (ET_c)

We recall that ET_c is calculated as

$$ET_c = k_c ET_0 \,, \tag{2}$$

where k_c is the crop coefficient and ET_0 is the reference crop evapotranspiration, usually alfalfa or grass.

4.1 Salt cedar crop coefficients (k_c)

Here we exemplify the uncertainty in the crop coefficient using the data compiled by Nichols et al. (2004) for salt cedar in the Bosque del Apache region, New Mexico. The data has been extracted from their Figure 10 using the free online tool WebPlotDigitizer, which can be found in https://apps.automeris.io/wpd/. We load two datasets and retrieve data only for month 5:

- ‘kc.evolution.cedar.csv‘, which includes the data on the mean k_c values.
- ‘kc.evolution.point.csv‘, which includes the individual k_c data points.

Then we illustrate the oasis effect on k_c values using the data shown by Allen et al. (1998) in their Figure 46, which we extract with the same online tool. Our dataset is `oasis_effect.csv`.

```
# K_C VALUES FOR SALT CEDAR ------------------------------------------------

# Read in dataset
kc_evolution_cedar <- fread("kc_evolution_cedar.csv")
kc_evolution_point <- fread("kc_evolution_point.csv")

# Retrieve minimum and maximmum values for month 5
cedar.min.max <- kc_evolution_point[x > 5 & x < 6]

# Plot
a <- ggplot(kc_evolution_cedar, aes(x = x, y = y)) +
  geom_line() +
  geom_point(
    data = kc_evolution_point, aes(x = x, y = y),
    color = "red", alpha = 0.4
  ) +
  scale_color_discrete(name = "") +
  annotate("text", x = 5.78, y = 0.8, label = "$k_c$ dev") +
  annotate("text", x = 8, y = 1.4, label = "$k_c$ med") +
  annotate("text", x = 9, y = 0.9, label = "$k_c$ late") +
  geom_vline(xintercept = 5, lty = 2) +
  theme_AP() +
  labs(x = "Month", y = "$k_c$")

# Read in data to show oasis effect
oasis <- fread("oasis_effect.csv")

# Plot and merge
b <- ggplot(oasis, aes(x = x, y = y, group = name, linetype = name)) +
  geom_line() +
  scale_linetype_discrete(name = "") +
  theme_AP() +
  labs(x = "Width of irrigation area (m)", y = "") +
```

```r
  theme(legend.position = c(0.6, 0.5))

cowplot::plot_grid(a, b, ncol = 2, labels = "auto", rel_widths = c(0.45, 0.55))
```

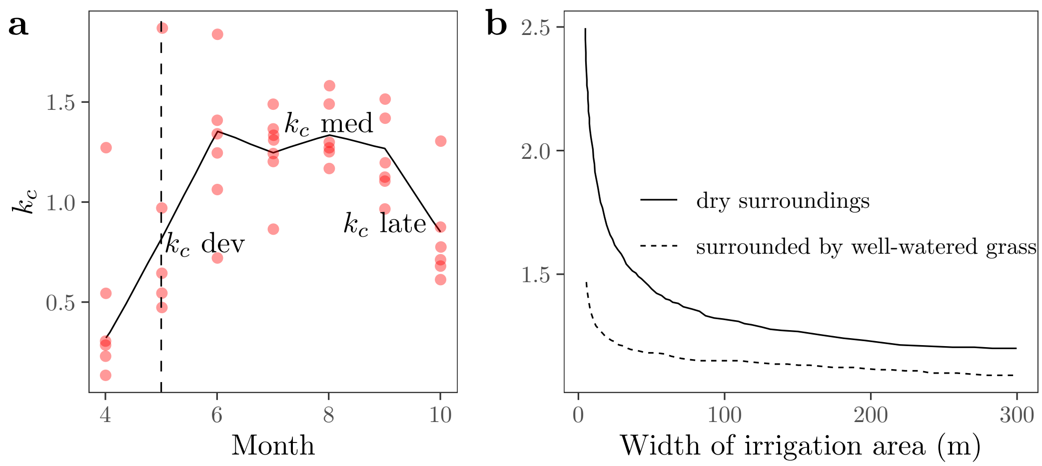
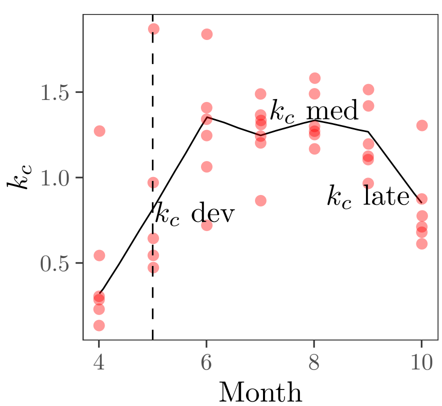


Figure 3: The crop coefficient. a) Evolution of k_c values for salt cedar. The solid black line is the mean k_c, the red dots show the individual measured values (adapted from Figure 10 in Nichols et al. (2004)), and the vertical, dashed black line marks the values selected for the global sensitivity analysis presented in Figure~1c and d. b) Oasis effect on k_c values (adapted from Figure 46 in Allen et al. (1998)).

```r
# PLOT K_C VALUES FOR SALT CEDAR ONLY ------------------------------------------

a
```



4.2 The reference evapotranpiration (ET_0)

We illustrate the effect of uncertainties in the computation of ET_0 using the Priestley-Taylor and the FAO-56 Penman-Monteith formulae. The former reads as follows

$$ET_0 = \alpha \frac{\Delta A}{\Delta + \gamma}\,, \tag{3}$$

whereas the latter, as

$$ET_0 = \frac{0.408 \Delta A + \gamma \frac{900}{T_a + 273} w v}{\Delta + \gamma (1 + 0.34 w)}\,, \tag{4}$$

where A is the net radiation minus the soil heat flux, Δ the gradient of saturated vapour pressure, γ the psychometric constant, α the Priestley-Taylor constant, T_a the mean daily air temperature at 2m, w the average daily wind speed at 2m and v the vapor pressure deficit (kPA). See Allen et al. (1998) for an explanation of the coefficients.

The uncertainty in the inputs is described following Nichols et al. (2004). See Table 1 of the main manuscript for the distributions selected. Our uncertainty and sensitivity analysis is done as explained in the Supplementary Materials file. We set the sample size of the base sample matrices at $N = 2^{12}$ and the number of bootstrap replicas of the Sobol' indices at $R = 10^3$.

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

N <- 2^12
R <- 10^3

# LISTED FUNCTIONS -------------------------------

et0_fun <- list(

  "pt" = function(alpha, delta, gamma, A, k_c = 1)
    k_c * (alpha * ((delta * A) / (gamma + delta))),

  "pm" = function(delta, A, gamma, T_a, w, v, k_c = 1)
    k_c * ((0.408 * delta * A + gamma * (900 / (T_a + 273)) * w * v) /
                    delta + gamma * (1 + 0.34 * w))
)

mat_transform <- function(mat, method, et_c = FALSE) {

  if (method == "pt") {
    mat[, "alpha"] <- qunif(mat[, "alpha"], 1.26 + 1.26 * - 0.1, 1.26 + 1.26 * 0.1)
    mat[, "delta"] <- qunif(mat[, "delta"], 0.21 + 0.21 * - 0.005, 0.21 + 0.21 * 0.005)
    mat[, "gamma"] <- qunif(mat[, "gamma"], 0.059 + 0.059 * - 0.001, 0.059 + 0.059 * 0.001)
    mat[, "A"] <- qunif(mat[, "A"], 350 + 350 * - 0.15, 350 + 350 * 0.15)

  } else if (method == "pm") {
    mat[, "delta"] <- qunif(mat[, "delta"], 0.21 + 0.21 * - 0.005, 0.21 + 0.21 * 0.005)
    mat[, "gamma"] <- qunif(mat[, "gamma"], 0.059 + 0.059 * - 0.001, 0.059 + 0.059 * 0.001)
    mat[, "A"] <- qunif(mat[, "A"], 350 + 350 * - 0.15, 350 + 350 * 0.15)
    mat[, "T_a"] <- qunif(mat[, "T_a"], 27 + 27 * - 0.01, 27 + 27 * 0.01)
    mat[, "w"] <- qunif(mat[, "w"], 2.5 + 2.5 * - 0.05, 2.5 + 2.5 * 0.05)
    mat[, "v"] <- qunif(mat[, "v"], 2.49 + 2.49 * - 0.04, 2.49 + 2.49 * 0.04)
  }

  if (et_c == TRUE) {
    mat[, "k_c"] <- qunif(mat[, "k_c"], min(cedar.min.max$y), max(cedar.min.max$y))
  }
  return(mat)
}
```

```r
params.shared <- c("delta", "gamma", "A")
params.pt <- c(params.shared, "alpha")
params.pm <- c(params.shared, "T_a", "v", "w")

# COMPUTATION OF ET_0 ------------------------------------------------------------

y <- list()

for (i in list("pt", "pm")) {

  if(i == "pt") {

    mat <- sobol_matrices(N = N, params = params.pt)
    mat <- mat_transform(mat = mat, method = i, et_c = FALSE)
    y[[i]] <- et0_fun[[i]](alpha = mat[, "alpha"],
                           delta = mat[, "delta"],
                           gamma = mat[, "gamma"],
                           A = mat[, "A"])

  } else if (i == "pm") {

    mat <- sobol_matrices(N = N, params = params.pm)
    mat <- mat_transform(mat = mat, method = i, et_c = FALSE)
    y[[i]] <- et0_fun[[i]](delta = mat[, "delta"],
                           A = mat[, "A"],
                           gamma = mat[, "gamma"],
                           T_a = mat[, "T_a"],
                           w = mat[, "w"],
                           v = mat[, "v"])
  }
}

# COMPUTATION OF ET_C ------------------------------------------------------------

y.kc <- list()

for(i in list("pt", "pm")) {

  if(i == "pt") {

    mat <- sobol_matrices(N = N, params = c(params.pt, "k_c"))
    mat <- mat_transform(mat = mat, method = i, et_c = TRUE)
    y.kc[[i]] <- et0_fun[[i]](alpha = mat[, "alpha"],
                              delta = mat[, "delta"],
                              gamma = mat[, "gamma"],
                              A = mat[, "A"],
                              k_c = mat[, "k_c"])
```

```r
  } else if (i == "pm") {

    mat <- sobol_matrices(N = N, params = c(params.pm, "k_c"))
    mat <- mat_transform(mat = mat, method = i, et_c = TRUE)
    y.kc[[i]] <- et0_fun[[i]](delta = mat[, "delta"],
                              A = mat[, "A"],
                              gamma = mat[, "gamma"],
                              T_a = mat[, "T_a"],
                              w = mat[, "w"],
                              v = mat[, "v"],
                              k_c = mat[, "k_c"])
  }
}

# FUNCTION TO EXTRACT UNCERTAINTY ------------------------------------------------

extract_unc <- function(out, N = N) {
  value_output <- unlist(lapply(out, function(x) x[1:N]), use.names = FALSE)
  da <- data.table(value_output)[, method:= rep(names(et0_fun), each = N)] %>%
    .[, method:= toupper(method)]
  gg <- ggplot(da, aes(value_output, fill = method)) +
    geom_histogram(alpha = 0.3, color = "black", position = "identity") +
    scale_fill_manual(name = "Method",
                      values = wes_palette(n = 2, name = "Chevalier1")) +
    theme_AP() +
    theme(legend.position = "none")
  return(gg)
}

# PLOT UNCERTAINTY ---------------------------------------------------------------

plots.unc <- lapply(list(y, y.kc), function(x) extract_unc(out = x, N = N))

plots.unc[[1]] <- plots.unc[[1]] + labs(x = "$ET_0$ (mm d$^{-1}$)", y = "Counts") +
  scale_x_continuous(limits = c(0, 800))
plots.unc[[2]] <- plots.unc[[2]] + labs(x = "$ET_c$ (mm d$^{-1}$)", y = "Counts") +
  scale_x_continuous(limits = c(0, 800))

# SOBOL' INDICES -----------------------------------------------------------------

params.pt.plot <- c("$\\Delta$", "$\\gamma$", "$A$", "$\\alpha$")
params.pm.plot <- c("$\\Delta$", "$\\gamma$", "$A$", "$T_a$", "$v$", "$w$")
first <- "jansen"

# ET0 --------------------------

ind <- list()
```

```r
for (i in list("pt", "pm")) {

  if (i == "pt") {
    params <- params.pt.plot

  } else if (i == "pm") {
    params <- params.pm.plot
  }

  ind[[i]] <- sobol_indices(Y = y[[i]], N = N, params = params,
                            first = first, boot = TRUE, R = R)
}

# ETO -------------------------

ind.kc <- list()

for( i in list("pt", "pm")) {

  if (i == "pt") {
    params <- c(params.pt.plot, "$k_c$")

  } else if (i == "pm") {
    params <- c(params.pm.plot, "$k_c$")
  }

  ind.kc[[i]] <- sobol_indices(Y = y.kc[[i]], N = N, params = params,
                               first = first, R = R, boot = TRUE)
}

# FUNCTION TO EXTRACT SOBOL' INDICES ---------------------------------------------

extract_sobol <- function(data) {
  out <- rbindlist(data, idcol = "method") %>%
    .[, method:= toupper(method)] %>%
    .[, parameters:= factor(parameters, levels = c("$\\Delta$", "$\\gamma$",
                                                    "$A$", "$T_a$", "$w$", "$v$",
                                                    "$\\alpha$", "$k_c$"))] %>%
    plot_sobol(.) +
    theme(legend.position = "none") +
    facet_grid(~ method,
               scales = "free_x",
               space = "free_x")
  return(out)
}

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

```r
plots.ind <- lapply(list(ind, ind.kc), function(x) extract_sobol(data = x))
```

```r
# MERGE ALL PLOTS ---------------------------------------------------------------

# Arrange legend
legend_ua <- cowplot::get_legend(plots.unc[[1]] + theme(legend.position = "top"))
legend_sa <- cowplot::get_legend(plots.ind[[1]] + theme(legend.position = "top"))
all_legend <- plot_grid(legend_ua, legend_sa, ncol = 2, rel_heights = c(0.01, 0.01))

# Arrange plots
top <- plot_grid(plots.unc[[1]], plots.ind[[1]], ncol = 2, labels = c("a", "b"))
bottom <- plot_grid(plots.unc[[2]], plots.ind[[2]], ncol = 2, labels = c("c", "d"))
all_plot <- plot_grid(top, bottom, ncol = 1)

# Merge
plot_grid(all_legend, all_plot, ncol = 1, rel_heights = c(0.1, 0.9))
```

5 Irrigation efficiency (E_p)

Here we study the uncertainty ranges of the different empirical datasets used by Döll and Siebert (2002) and Rohwer, Gerten, and Lucht (2007) to produce their E_p estimates.

- The `usa_efficiency.csv` dataset includes the data presented in Table 16 of Solley, Pierce, and Perlman (1998).
- The `fao_1997.csv` dataset includes the data in Table 8, Figure 10 and Figure 11 of FAO (1997).
- The `bos.dt.csv` includes all the data in Bos and Nugteren (1990).

We first plot the data to illustrate the uncertainty ranges.

```r
# PLOT EFFICIENCIES -------------------------------------------------------------

# USA data
usa.dt <- fread("usa_efficiency.csv")
usa.dt <- usa.dt[, Efficiency:= consumptive.use / total.withdrawal]

a <- ggplot(usa.dt, aes(Efficiency)) +
  geom_histogram() +
  geom_vline(xintercept = 0.6, lty = 2) +
  labs(x = "$E_p$", y = "Counts") +
  theme_AP()

# FAO 1997 (Irrigation potential in Africa)
fao_dt <- fread("fao_1997.csv")
fao_dt <- fao_dt[, Efficiency:= Efficiency / 100]

b <- ggplot(fao_dt, aes(Efficiency)) +
  geom_histogram() +
```

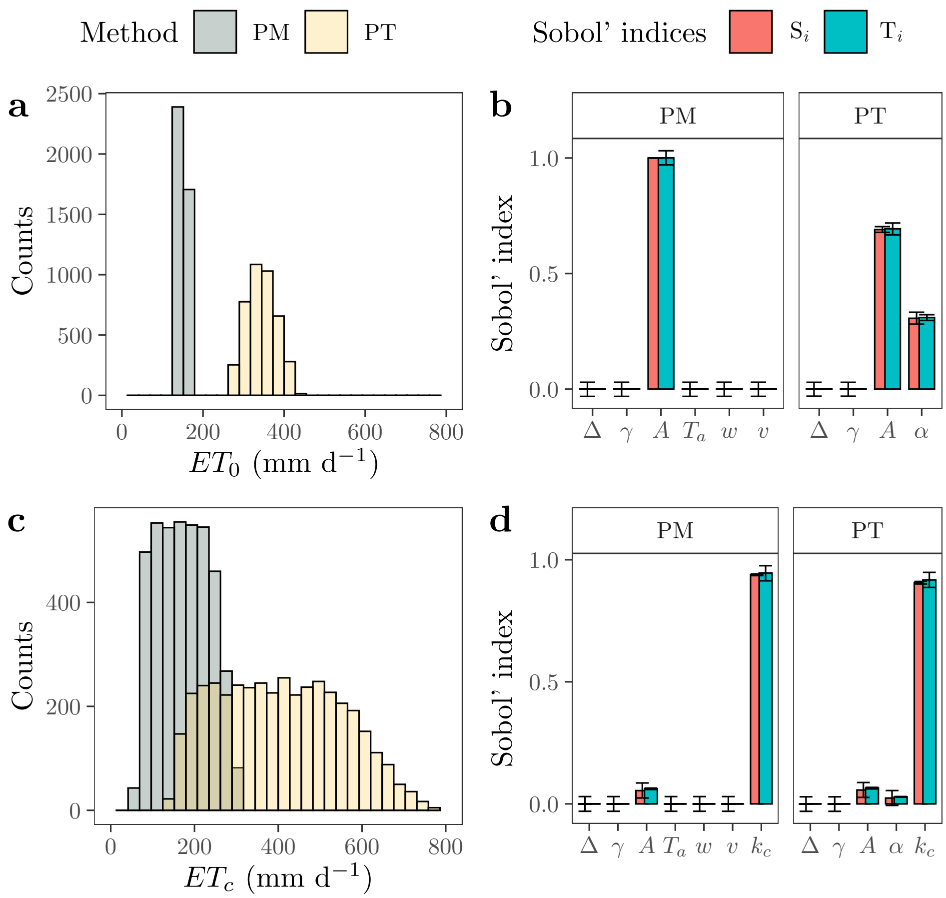


Figure 4: Monte-Carlo based, global uncertainty and sensitivity analysis of the Penman-Monteith (PM) and the Priestley-Taylor (PT) methods. The red bar indicates the first-order effect S_i, i.e. the proportion of variance conveyed by each model input. The blue bar reflects the total-order effect T_i, which includes the first-order effect of the parameter plus the effect derived of its interaction with the rest. The parameters were described with the probability distributions shown in Table~1. See the Supplementary Materials for a technical explanation of global uncertainty and sensitivity analysis. a) Empirical distribution of ET_0. b) Sobol' indices. c) Empirical distribution of ET_c. We described k_c as $U(0.47, 1.86)$ to reproduce the range reported at month 5 by Nichols et al. (2004) for developing salt cedars in the Bosque del Apache, New Mexico (see Figure~3a). d) Sobol' indices.

```r
  labs(x = "$E_p$", y = "") +
  theme_AP()

# Bos and Nugteren data
bos.dt <- fread("bos.dt.csv")
col_names <- colnames(bos.dt)[2:7]
setnames(bos.dt, col_names, paste("$", col_names, "$", sep = ""))

# Create data set with E_a values as defined by Rohwer
dt_e_a <- data.table("Type" = c("Sprinkler", "Surface"),
                     "Value" = c(0.75, 0.6))

c <- ggplot(bos.dt, aes(`$E_a$`)) +
  geom_histogram(bins = 15) +
  geom_vline(data = dt_e_a, aes(xintercept = Value), lty = 2) +
  facet_grid(~ Type) +
  labs(x = "$E_a$", y = "Counts") +
  theme_AP()

# Create data set with E_c values as defined by Rohwer
dt_e_c <- data.table("Type" = c("Surface", "Pressurized"),
                     "Value" = c(0.7, 0.95))

bos.dt.copy <- copy(bos.dt)

d <- bos.dt.copy[, Type:= ifelse(Type == "Surface", "Surface", "Pressurized")] %>%
  ggplot(., aes(`$E_c$`)) +
  geom_histogram(bins = 15) +
  geom_vline(data = dt_e_c, aes(xintercept = Value), lty = 2) +
  facet_grid(~ Type) +
  labs(x = "$E_c$", y = "Counts") +
  theme_AP()

# Merge all plots
top <- cowplot::plot_grid(a, b, ncol = 2, labels = "auto")
bottom <- cowplot::plot_grid(c, d, ncol = 1, labels = c("c", "d"))
cowplot::plot_grid(top, bottom, ncol = 1, rel_heights = c(0.35, 0.65))
```

```r
# EFFICIENCIES AS A FUNCTION OF SCALE ---------------------------------------------

dt.tmp <- bos.dt[, Scale := ifelse(Irrigated_area < 10000,
  "$<10.000$ ha", "$>10.000$ ha"
)] %>%
  na.omit()

melt(dt.tmp, measure.vars = c("$E_c$", "$E_a$", "$E_d$")) %>%
  ggplot(., aes(value)) +
```

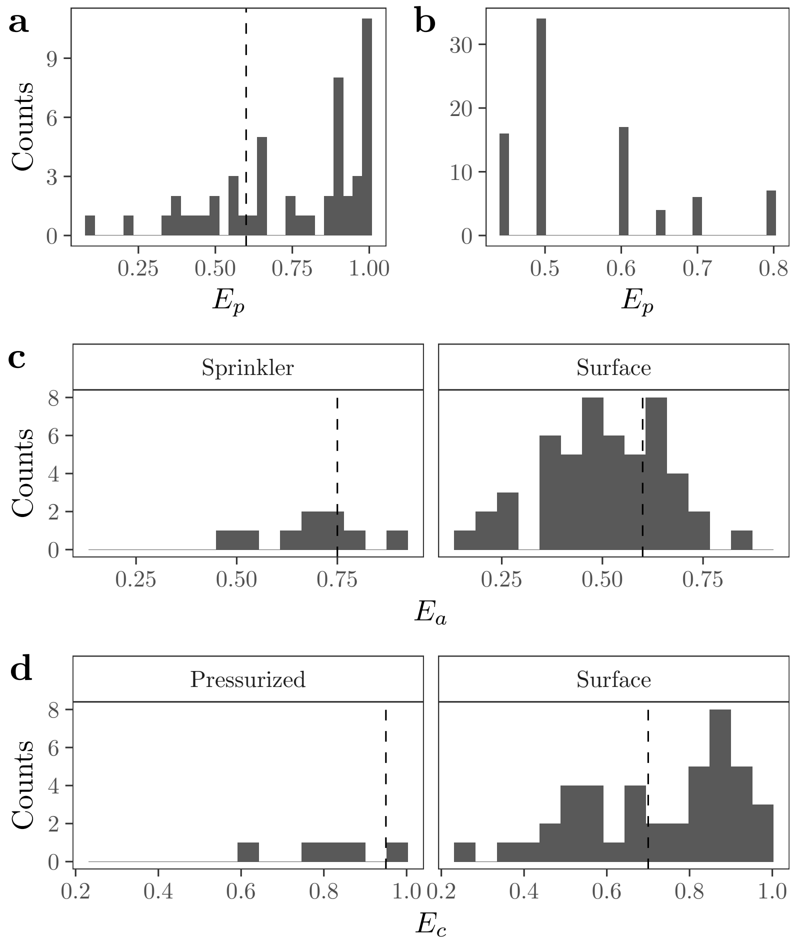


Figure 5: Data on irrigation efficiency. a) Distribution of project efficiencies (E_p) in USA according to Solley, Pierce, and Perlman (1998), calculated as Total water withdrawal / consumptive water use. The vertical dashed line is the E_p value used by Döll and Siebert (2002) to characterize the irrigation efficiency of USA. b) Distribution of project efficiencies (E_p) for Africa reported by FAO (1997). c) Distribution of field application efficiencies (E_a) in surface and sprinkler irrigation systems according to Bos and Nugteren (1990). `Surface` includes furrow, basin and border irrigation systems. The dashed vertical lines mark the E_a point estimates selected by Rohwer, Gerten, and Lucht (2007). d) Distribution of conveyance efficiencies (E_c) in surface and pressurized (sprinkler) irrigation systems according to Bos and Nugteren (1990). The vertical, dashed black lines show the E_c point estimates used by Rohwer, Gerten, and Lucht (2007).

```
  geom_histogram() +
  labs(x = "Value", y = "Counts") +
  facet_grid(Scale ~ variable) +
  theme_AP()
```

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

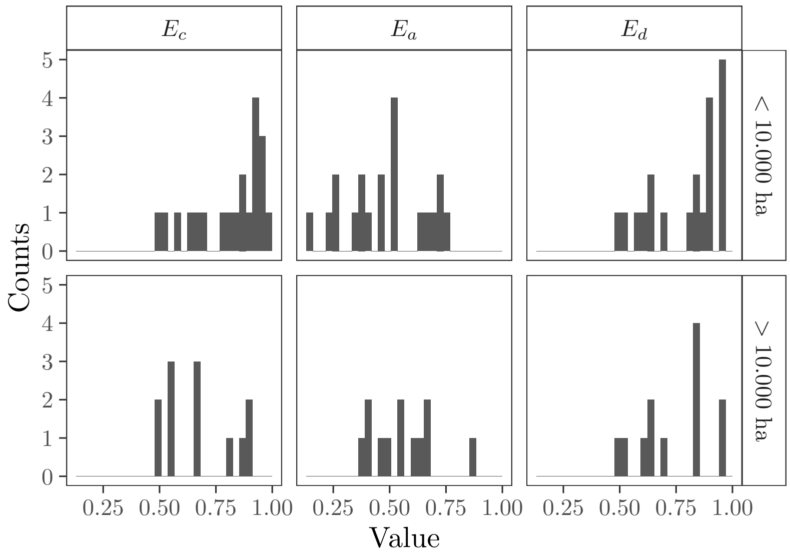


Figure 6: Partial efficiencies as a function of scale according to the data compiled by Bos and Nugteren (1990). E_d stands for distribution efficiency and refers to the water delivery from the source to a specific outlet. E_d is replaced by Rohwer, Gerten, and Lucht (2007) with M_f.

5.1 Rohwer et al. 's factorial design

Here we conduct an uncertainty and sensitivity analysis of Rohwer, Gerten, and Lucht (2007) approach to E_p, which is the following:

$$E_p = E_a E_c M_f \, , \tag{5}$$

% where E_a is the field application efficiency, E_c the conveyance efficiency and M_f a management factor. E_a and E_c are defined by the irrigation technology and M_f by the size of the irrigation system. We first define the size of base sample matrix ($N = 2^{14}$) and the number of bootstrap replicas of the Sobol' indices ($R = 10^3$), and transform each model input to its appropriate probability distributions. We then compare the modeled with the empirical distributions and create a plot where we show that E_p, E_a, M_f are not correlated.

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

N <- 2^14
params <- c("E_a", "E_c", "M_f")
R <- 10^3
```

```
# DEFINE SAMPLE MATRIX ----------------------------------------------------------
```

```r
mat <- sobol_matrices(N = N, params = params, type = "R")

# Truncated Weibull for E_a
shape <- 3.502469
scale <- 0.5444373
minimum <- 0.14
maximum <- 0.87
Fa.weibull <- pweibull(minimum, shape = shape, scale = scale)
Fb.weibull <- pweibull(maximum, shape = shape, scale = scale)
mat[, "E_a"] <- qunif(mat[, "E_a"], Fa.weibull, Fb.weibull)
mat[, "E_a"] <- qweibull(mat[, "E_a"], shape, scale)

# Truncated Beta for E_c
shape1 <- 5.759496
shape2 <- 1.403552
minimum <- 0.26
maximum <- 0.98
Fa.beta <- pbeta(minimum, shape1 = shape1, shape2 = shape2)
Fb.beta <- pbeta(maximum, shape1 = shape1, shape2 = shape2)
mat[, "E_c"] <- qunif(mat[, "E_c"], Fa.beta, Fb.beta)
mat[, "E_c"] <- qbeta(mat[, "E_c"], shape1, shape2)

# Truncated Weibull for M_f
shape <- 6.844793
scale <- 0.8481904
minimum <- 0.5
maximum <- 0.97
Fa.weibull <- pweibull(minimum, shape = shape, scale = scale)
Fb.weibull <- pweibull(maximum, shape = shape, scale = scale)
mat[, "M_f"] <- qunif(mat[, "M_f"], Fa.weibull, Fb.weibull)
mat[, "M_f"] <- qweibull(mat[, "M_f"], shape, scale)
```

```r
# PLOT DISTRIBUTIONS ---------------------------------------------------------------

dt <- data.table(mat)
params_plot <- paste("$", params, "$", sep = "")
dt <- setnames(dt, params, params_plot) %>%
  .[, Data:= "Modeled"]

# Filter empirical datasets
dt.ea <- bos.dt[Type == "Surface", `$E_a$`]
dt.ec <- bos.dt[Type == "Surface", `$E_c$`]
dt.mf <- bos.dt[Irrigated_area < 10000, `$E_d$`]

n <- max(length(dt.ea), length(dt.ec), length(dt.mf))

length(dt.ea) <- n
```

```r
length(dt.ec) <- n
length(dt.mf) <- n

dt.empirical <- cbind(dt.ea, dt.ec, dt.mf) %>%
  data.table() %>%
  .[, Data:= "Empirical"] %>%
  setnames(., colnames(.), colnames(dt))

rbind(dt.empirical, dt[1:N]) %>%
  melt(., measure.vars = params_plot) %>%
  ggplot(., aes(value)) +
  geom_histogram() +
  labs(x = "Value", y = "Counts") +
  scale_x_continuous(breaks = pretty_breaks(n = 4)) +
  facet_grid(Data ~ variable, scales = "free_y") +
  theme_AP()
```

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

```
## Warning: Removed 119 rows containing non-finite values (stat_bin).
```

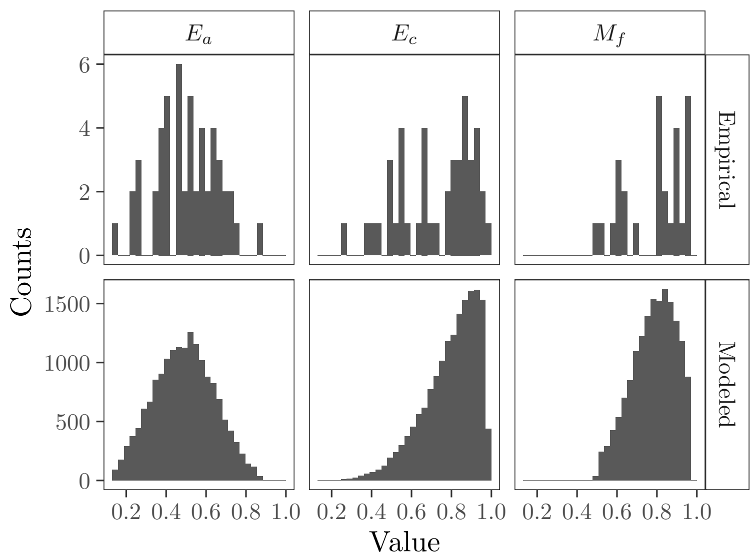


Figure 7: Empirical and modeled distributions for the uncertainty and sensitivity analysis of Equation 5.

```r
# PLOT SCATTERPLOTS TO CHECK FOR CORRELATION BETWEEN E_A, E_C, M_F ---------------

out <- t(combn(colnames(dt.empirical)[1:3], 2))
da <- list()

for (i in 1:nrow(out)) {
  cols <- out[i, ]
  da[[i]] <- cbind(dt.empirical[, ..cols], cols[1], cols[2])
  data.table::setnames(da[[i]], colnames(da[[i]]), c("xvar",
                                                      "yvar", "x", "y"))
```

```
}

output <- data.table::rbindlist(da)

ggplot(output, aes(xvar, yvar)) +
  geom_point() +
  facet_wrap(x ~ y) +
  labs(x = "$x$", y = "$y$") +
  theme_AP()
```

```
## Warning: Removed 165 rows containing missing values (geom_point).
```

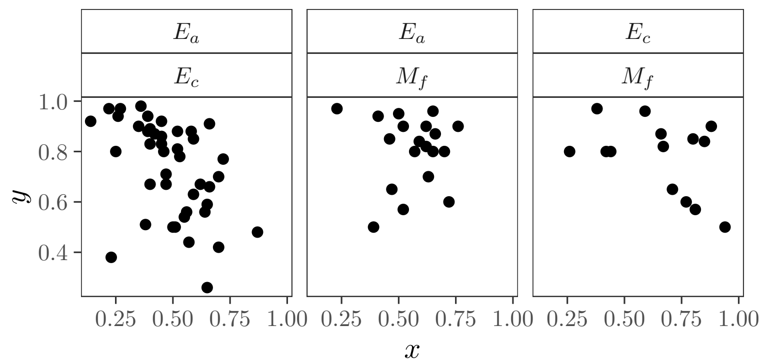


Figure 8: Scatterplots of E_a, E_c, M_f. The topmost and bottomost label facets refer to the x and the y axis respectively.

In the next code snippet we run Equation~5 and compute some statistics on the model output.

```
# DEFINE MODEL ------------------------------------------------------------------

y <- mat[, "E_a"] * mat[, "M_f"] * mat[, "E_c"]

# Some statistics
data.table(y)[, .(min = min(y), max = max(y))]
```

```
##           min       max
## 1: 0.02779241 0.7803253
```

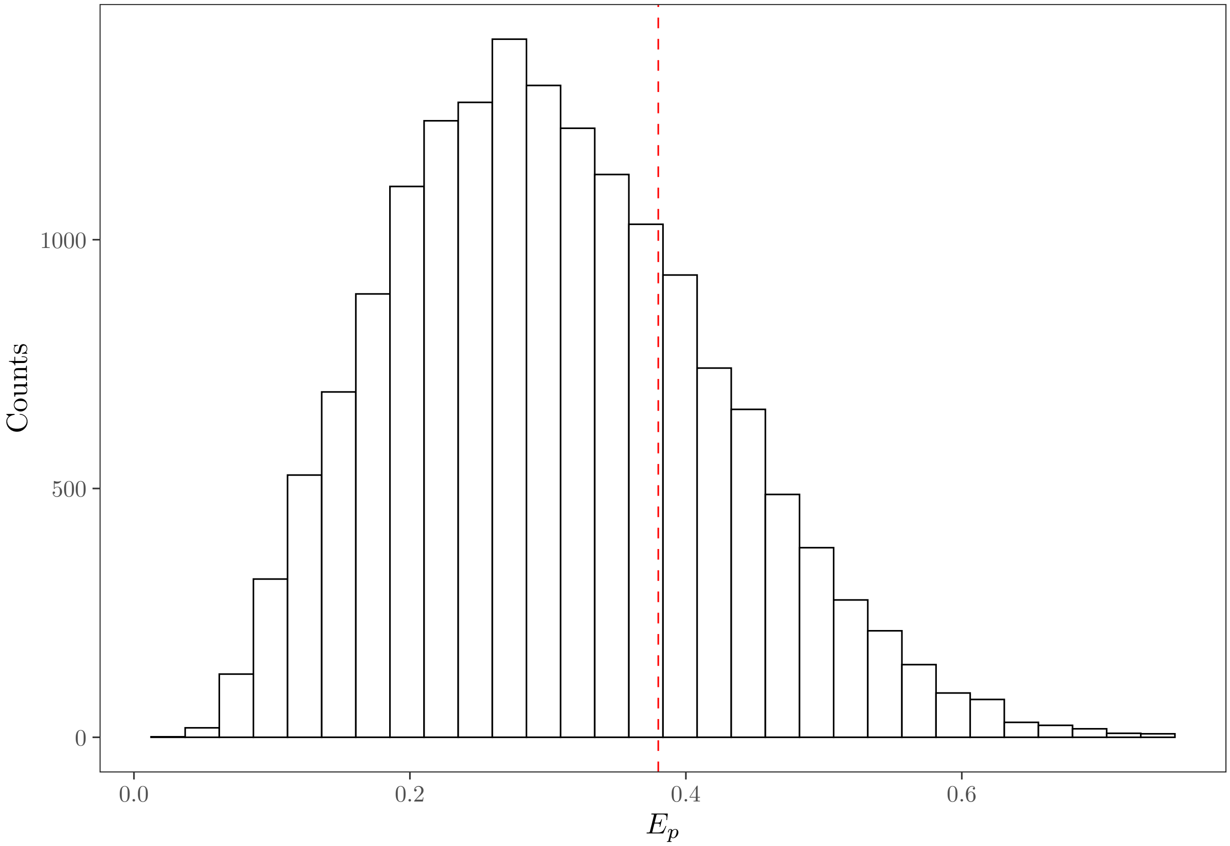
```
quantile(y, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.1091040 0.5562426
```

Here we plot the empirical distribution of E_p:

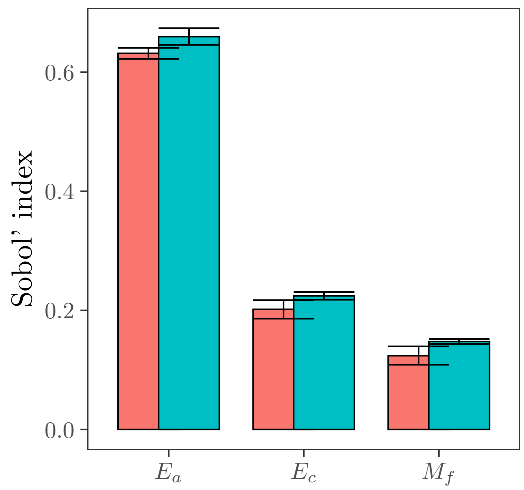
```
# PLOT UNCERTAINTY --------------------------------------------------------------

a <- plot_uncertainty(Y = y, N = N) +
  labs(x = "$E_p$", y = "Counts") +
  geom_vline(xintercept = 0.38, lty = 2, color = "red")
a
```

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



And compute the Sobol' indices below.

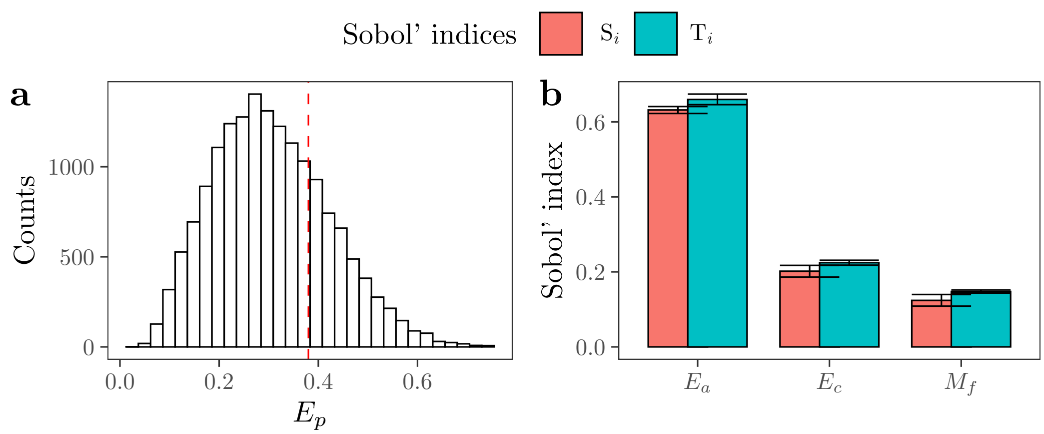
```
# SENSITIVITY ANALYSIS ------------------------------------------------------------

ind <- sobol_indices(
  Y = y, N = N, params = params_plot, R = R, boot = TRUE,
  first = "jansen"
)
b <- plot_sobol(ind) +
  theme(legend.position = "none")
b
```




```
# MERGE PLOTS ---------------------------------------------------------------------

legend <- get_legend(b + theme(legend.position = "top"))
bottom <- cowplot::plot_grid(a, b, ncol = 2, labels = "auto")
```

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

```
cowplot::plot_grid(legend, bottom, ncol = 1, rel_heights = c(0.15, 0.85))
```



6 Assessment of uncertainties

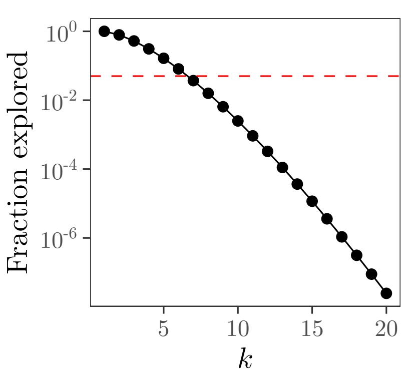
6.1 OAT's perfunctory exploration of the uncertainty space

In the code snippet below we prove that OAT perfunctorily explores the uncertainty space using the general formula for the volume of the hypersphere of radius 1/2 in k dimensions, as in Saltelli and Annoni (2010). The formula reads as follows:

$$r(k) = \frac{\pi^{\frac{k}{2}}}{\Gamma\left(\frac{k}{2}+1\right)}\left(\frac{1}{2}\right)^k \quad (6)$$

We check the fraction explored for $\mathbf{k} = 1, 2, \cdots, 20$

```r
# ASSESS THE FRACTION OF THE UNCERTAIN SPACE EXAMINED BY OAT ----------------------

# Formula: Ratio of the hypercube to the hypersphere
oat_exploration <- function(k) pi^((k) / 2) * (0.5)^(k) / gamma(1 + k / 2)

# Check from 1 to 20 dimensions
out <- sapply(1:20, function(x) oat_exploration(x))

# Plot
data.table(k = 1:20, x = out) %>%
  ggplot(., aes(k, x)) +
  geom_point() +
  scale_y_log10(
    breaks = trans_breaks("log10", function(x) 10^x),
    labels = trans_format("log10", math_format(10^.x))
  ) +
  geom_hline(yintercept = 0.05, lty = 2, color = "red") +
  geom_line() +
  labs(
    y = "Fraction explored",
    x = "$k$"
  ) +
  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
    )
  )
```



6.2 Assesment of uncertainties at the grid cell level

6.3 Uncertainty in the extension of irrigated areas in Texas

```
# FUNCTIONS TO CONVERT LON LAT TO USA STATES AND COUNTRIES -------------------------

# Lon Lat to USA states
# (extracted from https://stackoverflow.com/questions/
# 8751497/latitude-longitude-coordinates-to-state-code-in-r)
lonlat_to_state <- function(pointsDF,
                            states = spData::us_states,
                            name_col = "NAME") {
  ## Convert points data.frame to an sf POINTS object
  pts <- st_as_sf(pointsDF, coords = 1:2, crs = 4326)

  ## Transform spatial data to some planar coordinate system
  ## (e.g. Web Mercator) as required for geometric operations
  states <- st_transform(states, crs = 3857)
  pts <- st_transform(pts, crs = 3857)

  ## Find names of state (if any) intersected by each point
  state_names <- states[[name_col]]
  ii <- as.integer(st_intersects(pts, states))
  state_names[ii]
}

# Lon Lat to countries
coords2country <- function(points) {
  countriesSP <- rworldmap::getMap(resolution = "low")
  pointsSP <- sp::SpatialPoints(points, proj4string = CRS(proj4string(countriesSP)))
  indices <- sp::over(pointsSP, countriesSP)
  indices$ADMIN
}
```

```
# READ IN RASTERS ------------------------------------------------------------------

# Define parallel computing
n_cores <- floor(detectCores() * 0.75)
cl <- makeCluster(n_cores)
registerDoParallel(cl)

# Vector with the name of the files
c("fao_gmia.asc", "meier_map.tif", "iwmi_giam.tif")
```

```
## [1] "fao_gmia.asc"  "meier_map.tif" "iwmi_giam.tif"
```

```
vec_rasters <- c("fao_gmia.asc", "GRIPC_irrigated_area.asc")

# Load rasters and transform to csv in parallel
```

```r
out <- foreach(
  i = 1:length(vec_rasters),
  .packages = c(
    "raster", "data.table", "sf",
    "rworldmap", "sp"
  )
) %dopar% {
  rs <- raster(vec_rasters[i])
  dt <- data.table(rasterToPoints(rs,
    fun = function(r) {
      r > 0
    }
  ))
  states_vector <- lonlat_to_state(dt[, 1:2])
  dt[, states := cbind(states_vector)]
  dt[, country := coords2country(dt[, c("x", "y")])]
  setnames(dt, 3, "area")
}

# Stop parallel cluster
stopCluster(cl)
```

```r
# ARRANGE DATASET -----------------------------------------------------------------

# Read the meier map
meier.map <- fread("meier.map.csv")

# Arrange dataset
names(out) <- c("FAO-GMIA", "GRIPC")
rasters.dt <- rbindlist(out, idcol = "Map")

# Rbind GRIPC, FAO-GMIA and Meier map
all.rasters <- rbind(rasters.dt, meier.map)

# Check differences in global irrigated areas
all.rasters[, sum(area) / 10^6, Map] # Million ha
```

```
##         Map       V1
## 1: FAO-GMIA 307.6357
## 2:    GRIPC 248.5050
## 3:    Meier 368.0746
```

```r
# Export
fwrite(all.rasters, "all.rasters.csv")

# Coordinates of Uvalde, Texas
# 29.209684, -99.786171.
```

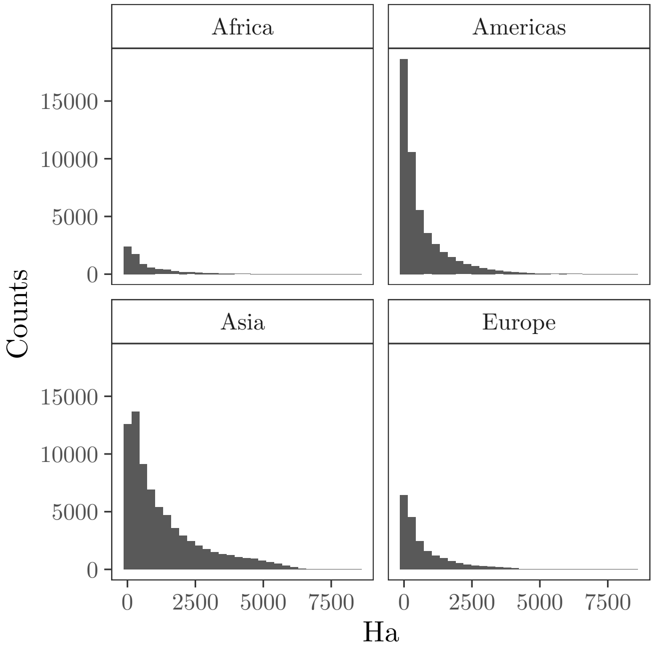
```r
# keep for later: [x< -99.6 & x > -99.8 & y > 29 & y < 29.5]

# meier map: x = -99.70416, y = 29.34583
rasters.merge <- copy(rasters.dt)
rasters.merge <- rasters.merge[, c("x", "y") := round(.SD, 4), .SDcols = c("x", "y")]
rasters.uvalde <- rbind(
  rasters.merge[x == -99.7083 & y == 29.4583],
  meier.map[x >= -99.71 & x <= -99.70 &
    y >= 29.2 & y <= 29.46]
)
```

```r
# CHECK DIFFERENCES AT THE GRID CELL LEVEL PER CONTINENT ---------------------------

rasters.dt <- rasters.dt[, c("x", "y"):= round(.SD, 4), .SDcols = c("x", "y")]
tmp <- merge(rasters.dt[Map == "FAO-GMIA"], rasters.dt[Map == "GRIPC"], by = c("x", "y"))

# Compute absolute difference at the grid level
tmp <- tmp[, abs:= abs(area.x - area.y)]

# Get continent
tmp <- tmp[, continent:= countrycode(tmp[, country.x],
                                     origin = "country.name",
                                     destination = "continent")]
```

```
## Warning in countrycode(tmp[, country.x], origin = "country.name", destination = "continent")
```

```r
# Plot
tmp[!continent == "Oceania"] %>%
ggplot(., aes(abs)) +
  geom_histogram() +
  labs(x = "Ha", y = "Counts") +
  facet_wrap(~continent) +
  theme_AP()
```

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



6.4 Uncertainty in k_c coefficients for wheat in Texas

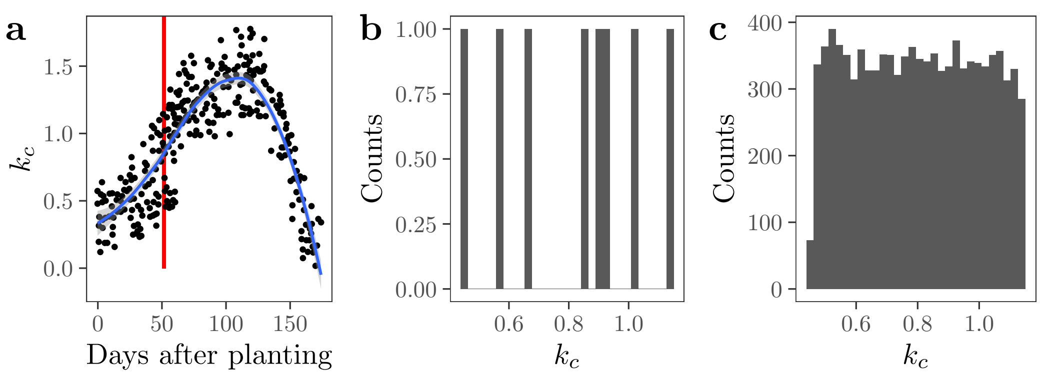
```
# CHECK THE UNCERTAINTY IN KC COEFFICIENTS FOR WHEAT IN TEXAS -----------------

kc_wheat <- fread("kc_wheat_new.csv")

# Define the time frame of the data
min.day <- 50
max.day <- 53

# Retrieve the filtered data
kc_wheat.dt <- kc_wheat[x > min.day & x < max.day][y > 0]

# Uniform distribution
v.final <- runif(10^4, min = min(kc_wheat.dt$y), max = max(kc_wheat.dt$y))
```

```
# PLOT DATA, EMPIRICAL DISTRIBUTION AND MODELED DISTRIBUTION -----------------------

a <- ggplot(kc_wheat, aes(x = x, y = y)) +
  annotate("rect",
    xmin = min.day, xmax = max.day,
    ymin = 0, ymax = Inf, fill = "red"
  ) +
  geom_point(size = 0.6) +
  geom_smooth() +
  labs(x = "Days after planting", y = "$k_c$") +
  theme_AP()

b <- ggplot(kc_wheat.dt, aes(y)) +
```

```
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  labs(x = "$k_c$", y = "Counts") +
  theme_AP()

c <- ggplot(data.table(v.final), aes(v.final)) +
  geom_histogram() +
  labs(x = "$k_c$", y = "Counts") +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  theme_AP()

plot_grid(a, b, c, ncol = 3, labels = "auto")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

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



6.5 Uncertainty and sensitivity analysis

```
# READ IN CLIMATIC DATA FOR UVALDE FOR JANUARY 2007-------------------------------

da <- fread("uvalde_climate_data_january.csv")

# Turn columns into numeric
col_names <- colnames(da)
da[, (col_names):= lapply(.SD, as.numeric), .SDcols = (col_names)]

# Convert Fahrenheit to Celsius
temp_cols <- colnames(da)[da[ , grepl( "Temp", colnames(da))]]
da[, (temp_cols):= lapply(.SD, function(x) (x - 32) / 1.8), .SDcols = (temp_cols)]

# Convert miles per hour to meters per second
wind_cols <- colnames(da)[da[ , grepl( "Wind", colnames(da))]]
da[, (wind_cols):= lapply(.SD, function(x) x / 2.237), .SDcols = (wind_cols)]

# Select days of interest
```

```r
uvalde_days <- da[Day >= 6 & Day <= 7]

# Mean temperature
T_air <- mean(uvalde_days$`Temp.Avg`)
T_air
```

```
## [1] 11
```

```r
# Computation of Vapour pressure (Delta)
# T_air <- 27
e_sat <- 0.6108 * exp((17.27 * T_air) / (T_air + 237.3))
Delta <- (4098 * e_sat) / ((T_air + 237.3) ^ 2)
Delta
```

```
## [1] 0.08725467
```

```r
# Computation of vapour deficit (v)
rel_hum <- mean(uvalde_days$`Hum.Avg`) / 100
v <- e_sat - (e_sat * rel_hum)

# Computation of Psychrometric constant (gamma)
z <- 9.1
lambda <- 2.501 - 0.002361 * T_air
P <- 101.3 * ((293 - 0.0065 * z) / 293) ^ 5.256
Gamma <- 0.0016286 * P / lambda
Gamma
```

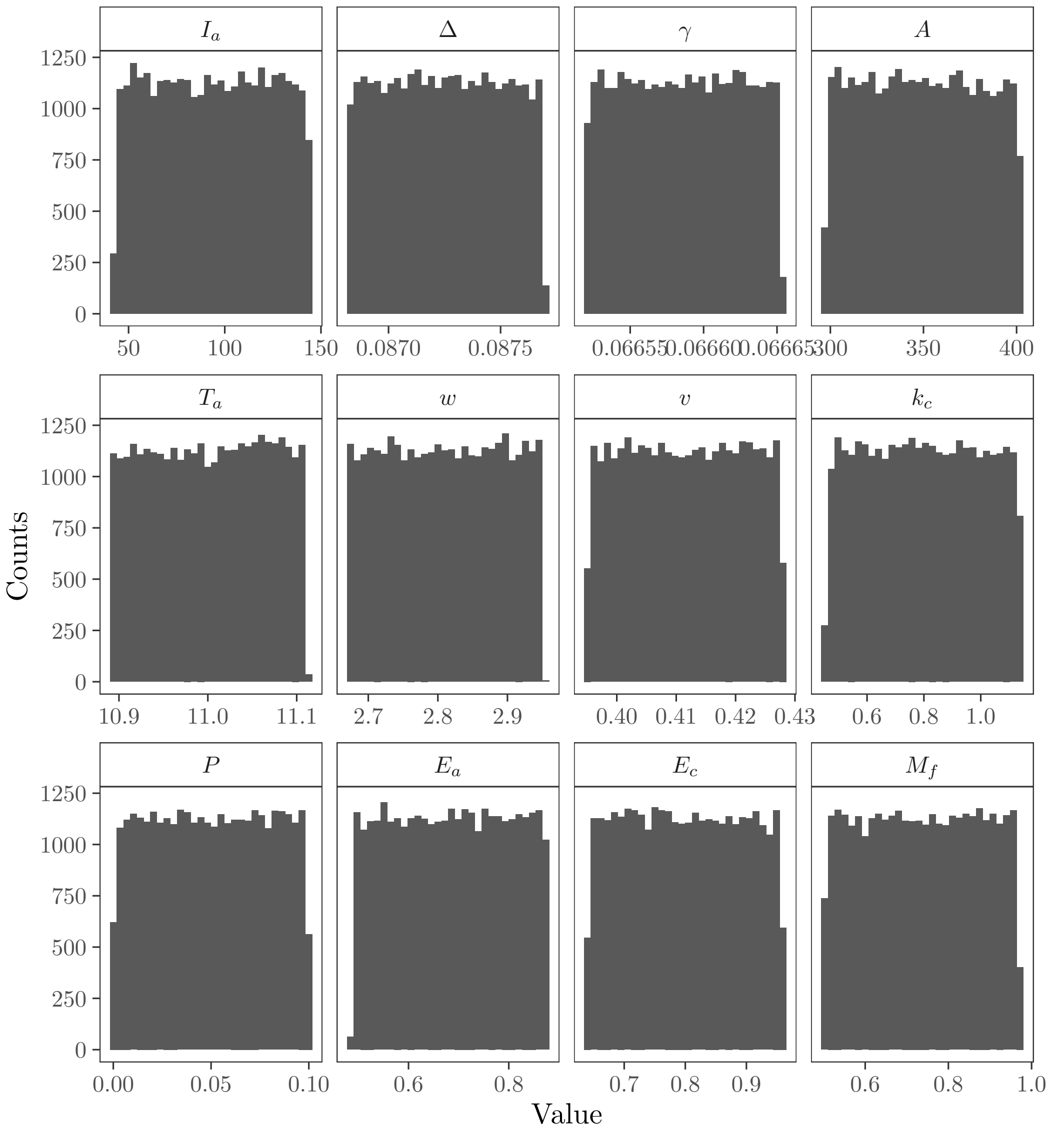
```
## [1] 0.06658597
```

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

N <- 2^15
R <- 10^3
type <- "R"
order <- "second"
params <- c(
  "I_a", "Delta", "gamma", "A", "T_a", "w", "v",
  "k_c", "P", "E_a", "E_c", "M_f"
)

# Vector with the name of the parameters modified for better plotting
params.plot <- c(
  "$I_a$", "$\\Delta$", "$\\gamma$", "$A$", "$T_a$", "$w$",
  "$v$", "$k_c$", "$P$", "$E_a$", "$E_c$", "$M_f$"
)

# I_a (ha)
# P (mm)
# ET_0 (mm)
```

```r
# DEFINE SAMPLE MATRIX AND TRANSFORM TO APPROPRIATE DISTRIBUTIONS -----------------

# Define sampling matrix
mat <- sobol_matrices(N = N, params = params, order = order, type = type)

# Transform to appropriate probability distributions
mat[, "I_a"] <- qunif(mat[, "I_a"], rasters.uvalde[, min(area)], rasters.uvalde[, max(area)])
mat[, "Delta"] <- qunif(mat[, "Delta"], Delta + Delta * -0.005, Delta + Delta * 0.005)
mat[, "gamma"] <- qunif(mat[, "gamma"], Gamma + Gamma * -0.001, Gamma + Gamma * 0.001)
mat[, "A"] <- qunif(mat[, "A"], 350 + 350 * -0.15, 350 + 350 * 0.15)
mat[, "T_a"] <- qunif(mat[, "T_a"], T_air + T_air * -0.01, T_air + T_air * 0.01)
mat[, "w"] <- qunif(mat[, "w"], 2.81 + 2.81 * -0.05, 2.81 + 2.81 * 0.05)
mat[, "v"] <- qunif(mat[, "v"], v + v * -0.04, v + v * 0.04)
mat[, "P"] <- qunif(mat[, "P"], 0, 0.1)
mat[, "M_f"] <- qunif(mat[, "M_f"], 0.5, 0.97)
mat[, "k_c"] <- qunif(mat[, "k_c"], min(kc_wheat.dt$y), max(kc_wheat.dt$y))
mat[, "E_a"] <- qunif(mat[, "E_a"], min(bos.dt[Type == "Sprinkler", `$E_a$`], na.rm = TRUE),
                      max(bos.dt[Type == "Sprinkler", `$E_a$`], na.rm = TRUE))
mat[, "E_c"] <- qunif(mat[, "E_c"], min(bos.dt[Type == "Sprinkler", `$E_c$`], na.rm = TRUE),
                      max(bos.dt[Type == "Sprinkler", `$E_c$`], na.rm = TRUE))
```

```r
# PLOT DISTRIBUTIONS ---------------------------------------------------------

data.table(mat[1:N, ]) %>%
  setnames(., params, params.plot) %>%
  melt(., measure.vars = params.plot) %>%
  ggplot(., aes(value)) +
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  labs(x = "Value", y = "Counts") +
  theme_AP() +
  facet_wrap(~variable, scales = "free_x")
```

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




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

full_model <- function(I_a, Delta, A, gamma, T_a, w, v, k_c, P, E_a, E_c, M_f) {
  out <- (I_a * (k_c * ((0.408 * Delta * A + gamma * (900 / (T_a + 273)) * w * v) /
    Delta + gamma * (1 + 0.34 * w))) - P) / (E_a * E_c * M_f)
  return(out)
}
```

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

y <- full_model(
  I_a = mat[, "I_a"],
  Delta = mat[, "Delta"],
  A = mat[, "A"],
  gamma = mat[, "gamma"],
  T_a = mat[, "T_a"],
```

```r
  w = mat[, "w"],
  v = mat[, "v"],
  k_c = mat[, "k_c"],
  P = mat[, "P"],
  E_a = mat[, "E_a"],
  E_c = mat[, "E_c"],
  M_f = mat[, "M_f"]
)
```

```r
# ASSESS UNCERTAINTIES -------------------------------------------------------

unc <- plot_uncertainty(Y = y, N = N)
```

```r
# ASSESS SENSITIVITIES ------------------------------------------------------------

# Compute sobol' indices
ind <- sobol_indices(
  Y = y, N = N, params = params.plot,
  order = order, boot = TRUE, R = R,
  parallel = "multicore"
)

# Everything is explained by first and second-order effects
ind[sensitivity %in% c("Si", "Sij"), sum(original)]
```

```
## [1] 1.000648
```

```r
# Plot sobol' indices
sobol.plot <- plot_sobol(ind) +
  theme(legend.position = c(0.83, 0.5))
```

```r
# PLOT SECOND-ORDER INDICES -------------------------------------------------------

second.order <- plot_sobol(ind, "second")
```

```r
# PLOT SCATTERPLOTS ---------------------------------------------------------------
```

6.6 One-at-a-time (OAT)

```r
# CONSTRUCT SAMPLE MATRIX ----------------------------------------------------------

A <- mat[1:N, ]
B <- matrix(rep(Rfast::colmeans(A), each = N), nrow = N)

X <- B
for (j in 1:ncol(A)) {
  AB <- B
  AB[, j] <- A[, j]
  X <- rbind(X, AB)
```

```r
}

mat.oat <- X[(N + 1):nrow(X), ]
colnames(mat.oat) <- params
```

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

y.oat <- full_model(
  I_a = mat.oat[, "I_a"],
  Delta = mat.oat[, "Delta"],
  A = mat.oat[, "A"],
  gamma = mat.oat[, "gamma"],
  T_a = mat.oat[, "T_a"],
  w = mat.oat[, "w"],
  v = mat.oat[, "v"],
  k_c = mat.oat[, "k_c"],
  P = mat.oat[, "P"],
  E_a = mat.oat[, "E_a"],
  E_c = mat.oat[, "E_c"],
  M_f = mat.oat[, "M_f"]
)
```

```r
# COMPUTE A SINGLE-POINT ESTIMATE USING MEAN VALUES--------------------------------

vec_means <- colMeans(A)

y.point <- full_model(
  I_a = vec_means[["I_a"]],
  Delta = vec_means[["Delta"]],
  A = vec_means[["A"]],
  gamma = vec_means[["gamma"]],
  T_a = vec_means[["T_a"]],
  w = vec_means[["w"]],
  v = vec_means[["v"]],
  k_c = vec_means[["k_c"]],
  P = vec_means[["P"]],
  E_a = vec_means[["E_a"]],
  E_c = vec_means[["E_c"]],
  M_f = vec_means[["M_f"]]
)

y.point
```

```
## [1] 27075.95
```

6.7 Compare OAT and global sensitivity analysis

```
# ASSESS UNCERTAINTIES ---------------------------------------------------------

unc.oat <- plot_uncertainty(Y = y.oat, N = N)

full.unc <- data.table(cbind(y[1:N], y.oat))
colnames(full.unc) <- c("Global", "OAT")

a <- melt(full.unc, measure.vars = colnames(full.unc)) %>%
  ggplot(., aes(value, fill = variable)) +
  geom_histogram(position = "identity", alpha = 0.3, color = "black") +
  labs(x = "$y$", y = "Counts") +
  scale_x_log10(
    breaks = trans_breaks("log10", function(x) 10^x),
    labels = trans_format("log10", scales::math_format(10^.x))
  ) +
  geom_vline(xintercept = y.point, lty = 2, color = "red", size = 2) +
  scale_fill_manual(values = wes_palette(2, name = "Chevalier1"), name = "Uncertainty analysis"
  theme_AP() +
  theme(legend.position = c(0.2, 0.5))
```

```
# SOME STATISTICS --------------------------------------------------------------

stat.full.unc <- melt(full.unc, measure.vars = c("Global", "OAT"))
stat.full.unc[, .(min = min(value), max = max(value)), variable]
```

```
##    variable       min       max
## 1:   Global  4407.492 121728.35
## 2:      OAT 12403.700  41762.57
```

```
# Quantiles
probs.quantile <- c(0, 0.025, 0.1, 0.5, 0.9, 0.975, 1)
stat.full.unc[, .(value = quantile(value, probs = probs.quantile)), variable] %>%
  .[, quantile := rep(probs.quantile, 2)] %>%
  dcast(., variable ~ quantile, value.var = "value") %>%
  print()
```

```
##    variable         0    0.025    0.1      0.5      0.9    0.975         1
## 1:   Global  4407.492  9145.979 13024.1 25966.80 49904.26 67348.10 121728.35
## 2:      OAT 12403.700 17976.290 23111.1 27075.95 32100.00 37331.91  41762.57
```

```
# Sum of first and second-order indices
ind[sensitivity == "Si", sum(original)]
```

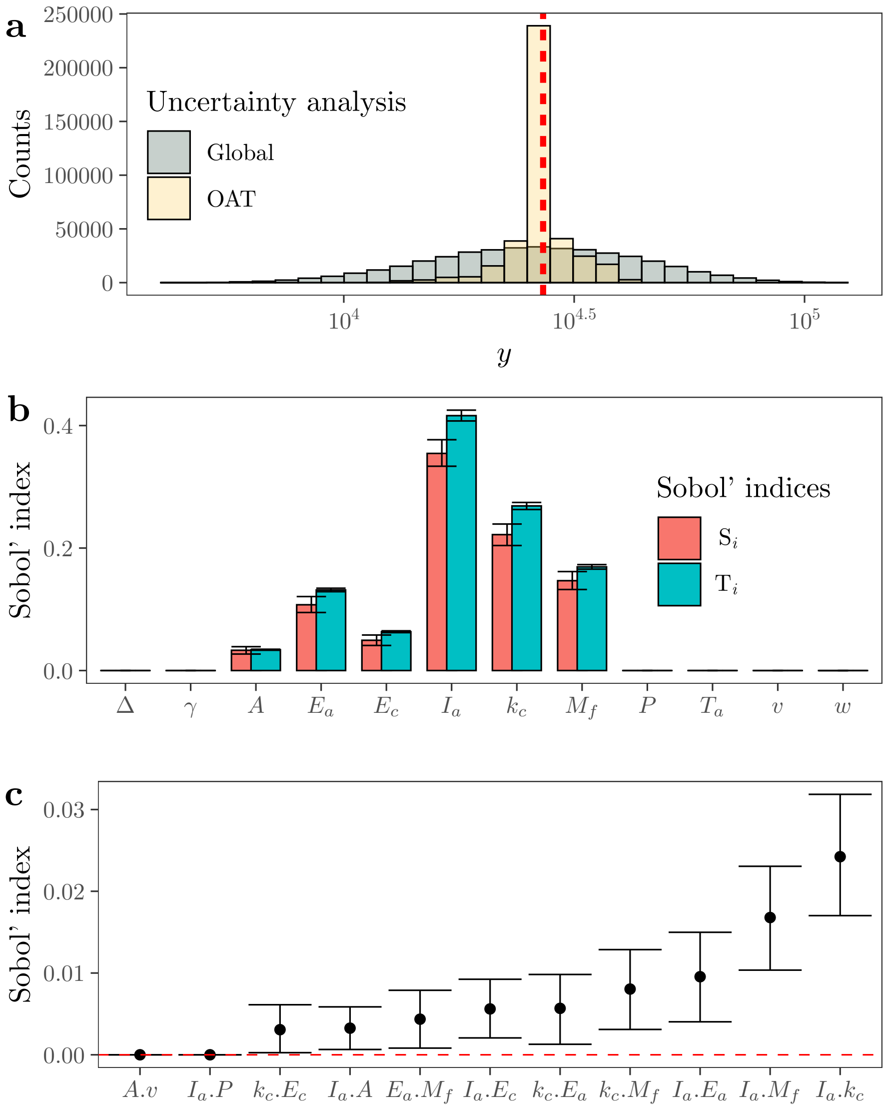
```
## [1] 0.9131286
```

```
ind[sensitivity %in% c("Si", "Sij"), sum(original)]
```

```
## [1] 1.000648
```

```r
# MERGE UNCERTAINTY AND SOBOL' INDICES ----------------------------------------

plot_grid(a, sobol.plot, second.order, ncol = 1, labels = "auto")
```

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



Allen, Richard G., Luis S. Pereira, D. Raes, and M. Smith. 1998. *Crop evapotranspiration (guidelines for computing crop water requirements)*. Vol. 56. Rome: FAO. https://doi.org/10.1016/j.eja.2010.12.001.

———. 1998. *Crop evapotranspiration (guidelines for computing crop water requirements)*. Vol. 56. Rome: FAO. https://doi.org/10.1016/j.eja.2010.12.001.

Bos, M. G., and J Nugteren. 1990. "On Irrigation Efficiencies." Wageningen: International Institute for Land Recamation; Improvement / ILRI. http://www1.frm.utn.edu.ar/laboratorio_hidraulica/Biblioteca_Virtual/Publicaciones%20de%20ILRI/On%20irrigation%20efficiencies.pdf.

———. 1990. "On Irrigation Efficiencies." Wageningen: International Institute for Land Recamation;

Improvement / ILRI. http://www1.frm.utn.edu.ar/laboratorio_hidraulica/Biblioteca_Virtual/Publicaciones%20de%20ILRI/On%20irrigation%20efficiencies.pdf.

Döll, P., and S. Siebert. 2002. "Global modeling of irrigation water requirements." *Water Resources Research* 38 (4): 8-1-8-10. https://doi.org/10.1029/2001WR000355.

———. 2002. "Global modeling of irrigation water requirements." *Water Resources Research* 38 (4): 8-1-8-10. https://doi.org/10.1029/2001WR000355.

FAO. 1997. "Irrigation potential in Africa. A basin approach. FAO Land and Water Bulletin 4." Rome: Food; Agriculture Organization of the United Nations. http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf.

———. 2016. "AQUASTAT website." Food; Agriculture Organization of the United Nations. http://www.fao.org/nr/water/aquastat/didyouknow/index3.stm.

———. 2017. "FAOSTAT database." Rome. http://www.fao.org/faostat/en/.

Meier, J., F. Zabel, and W. Mauser. 2018. "A global approach to estimate irrigated areas. A comparison between different data and statistics." *Hydrology and Earth System Sciences* 22 (2): 1119–33. https://doi.org/10.5194/hess-22-1119-2018.

Nichols, J., W. Eichinger, D. I. Cooper, J. H. Prueger, L. E. Hipps, C. M. U. Neale, and A. S. Bawazir. 2004. "Comparison of evaporation estimation methods for a riparian area. Final report." 436. Vol. 436. Iowa City: IIHR - Hydroscience & Engineering, College of Engineering, University of Iowa.

Rohwer, Janine, Dieter Gerten, and Wolfgang Lucht. 2007. "Development of functional irrigation types for improved global crop modelling." *PIK Report*, no. 104: 1–61.

Salmon, J. M., M. A. Friedl, S. Frolking, D. Wisser, and E. M. Douglas. 2015. "Global rain-fed, irrigated, and paddy croplands: A new high resolution map derived from remote sensing, crop inventories and climate data." *International Journal of Applied Earth Observation and Geoinformation* 38: 321–34. https://doi.org/10.1016/j.jag.2015.01.014.

Saltelli, A., and P. Annoni. 2010. "How to avoid a perfunctory sensitivity analysis." *Environmental Modelling and Software* 25 (12): 1508–17. https://doi.org/10.1016/j.envsoft.2010.04.012.

———. 2010. "How to avoid a perfunctory sensitivity analysis." *Environmental Modelling and Software* 25 (12): 1508–17. https://doi.org/10.1016/j.envsoft.2010.04.012.

Siebert, S., V. Henrich, K. Frenken, and J. Burke. 2013. "Update of the digital global map of irrigation areas to version 5." Rome: Rheinische Friedrich-Wilhelms-University; Food; Agriculture Organization of the United Nations.

Solley, Wayne B., Robert R. Pierce, and How Perlman. 1998. "Estimated Use of Water in the United States in 1995." US Geological Survey Circular 1200. http://www.usgs.gov/default.asp.

Thenkabail, P. S., C. M. Biradar, P. Noojipady, V. Dheeravath, Y. Li, M. Velpuri, M. Gumma, et al. 2009. "Global irrigated area map (GIAM), derived from remote sensing, for the end of the last millennium." *International Journal of Remote Sensing* 30 (14): 3679–3733. https://doi.org/10.1080/01431160802698919.