Global irrigation water demands biased by unreliable irrigation efficiencies

R code

Arnald Puy

Contents

```r
# 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("data.table", "tidyverse", "sensobol", "wesanderson",
               "cowplot", "parallel", "foreach", "doParallel",
               "countrycode", "ggridges", "scales", "overlapping",
               "sp", "rworldmap", "ncdf4"))

# Create 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"),
          plot.margin = margin(t = 0, r = 0.3, b = 0, l = 0.3, unit ="cm"))
}

# Set checkpoint

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

checkpoint("2021-08-02",
           R.version ="4.0.3",
           checkpointLocation = getwd())
```

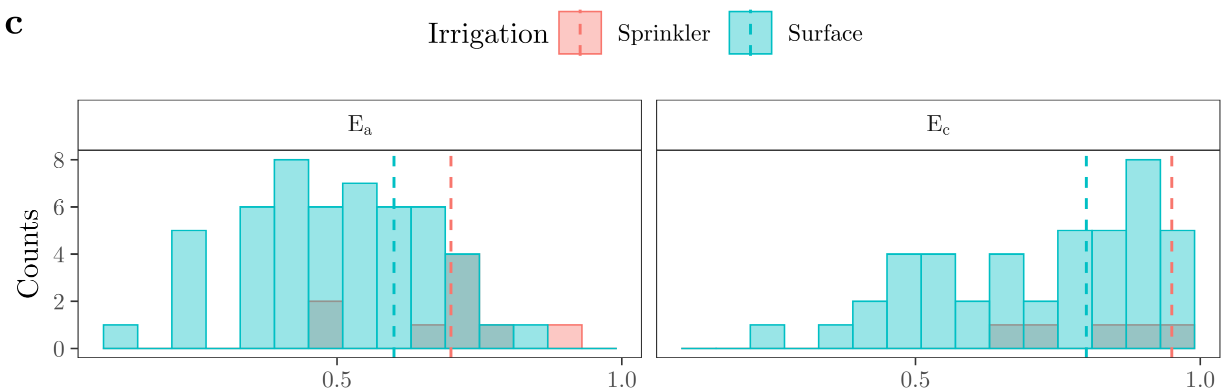
1 Read in data

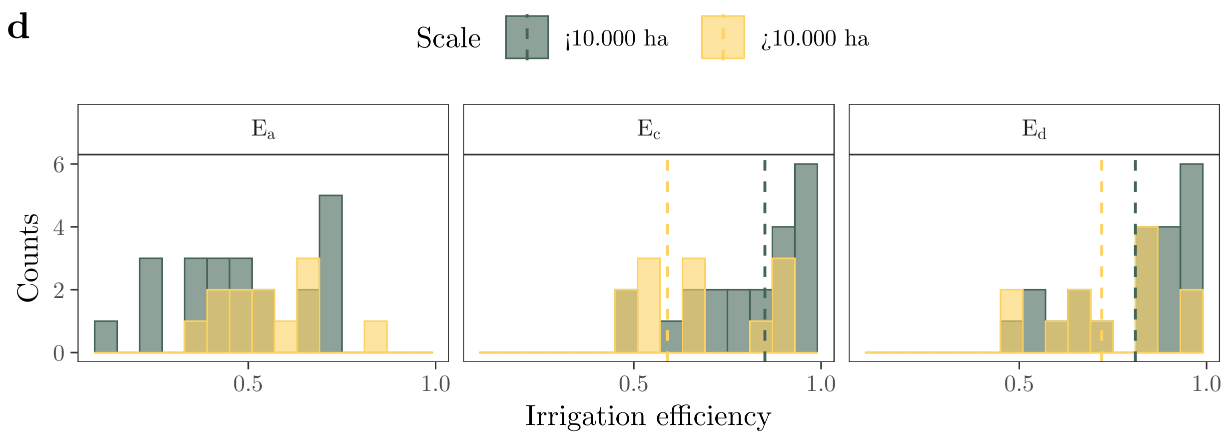
```
# READ IN DATA ---------------------------------------------------------------------

# Rohwer data
rohwer <- fread("rohwer_data_all.csv")
rohwer[rohwer == ""] <- NA
rohwer <- rohwer[, Large_fraction:= Large_fraction / 100]

# Jager data
jager <- fread("jager_data.csv")
jager.list <- split(jager, jager$Country)

# Bos data
bos <- fread("bos_data.csv")
bos <- bos[, Scale := ifelse(Irrigated_area < 10000, "<10.000 ha", ">10.000 ha")]

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

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

# Create data set with E_a values as defined by Rohwer
bos.rohwer.ea <- data.table("Irrigation" = c("Surface", "Sprinkler"),
                            "Value" = c(0.6, 0.7),
                            "variable" = "E[a]")

# Create data set with E_c values as defined by Rohwer
bos.rohwer.ec <- data.table("Irrigation" = c("Surface", "Sprinkler"),
                            "Value" = c(0.8, 0.95),
                            "variable" = "E[c]")

bos.rohwer.all <- rbind(bos.rohwer.ec, bos.rohwer.ea)

# As a function of scale
bos.rohwer.mf.ec <- data.table("Scale" = c("<10.000 ha", ">10.000 ha"),
                               "Value" = c(0.85, 0.59),
                               "variable" = "E[c]")

bos.rohwer.mf.ed <- data.table("Scale" = c("<10.000 ha", ">10.000 ha"),
                               "Value" = c(0.81, 0.72),
                               "variable" = "E[d]")

bos.rohwer.mf.all <- rbind(bos.rohwer.mf.ec, bos.rohwer.mf.ed)
```

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

bos2 <- copy(bos)
bos2 <- setnames(bos2, c("E_a", "E_c", "E_d"), c("E[a]", "E[c]", "E[d]"))

# Field and conveyance efficiency ----------------

a <- bos2 %>%
  melt(., measure.vars = c("E[a]", "E[c]")) %>%
  ggplot(., aes(value, fill = Irrigation, color = Irrigation)) +
  geom_histogram(position = "identity", alpha = 0.4, bins = 15) +
  facet_wrap(~variable, labeller = label_parsed) +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_vline(data = bos.rohwer.all, aes(xintercept = Value,
                                        color = Irrigation,
                                        group = variable),
             lty = 2,
             size = 1) +
  labs(x = "", y = "Counts") +
  theme_AP()

# As a function of scale ----------------------

b <- melt(bos2, measure.vars = c("E[c]", "E[a]", "E[d]")) %>%
  na.omit() %>%
  ggplot(., aes(value, fill = Scale, color = Scale)) +
  geom_histogram(bins = 15, position = "identity", alpha = 0.6) +
  labs(x = "Irrigation efficiency", y = "Counts") +
  facet_wrap(~ variable, labeller = label_parsed) +
  geom_vline(data = bos.rohwer.mf.all, aes(xintercept = Value,
                                          color = Scale,
                                          group = variable),
             lty = 2,
             size = 1) +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  scale_color_manual(values = wes_palette(2, name = "Chevalier1"),
                     name = "Scale",
                     labels = c("<10.000 ha", ">10.000 ha")) +
  scale_fill_manual(values = wes_palette(2, name = "Chevalier1"),
                    name = "Scale",
                    labels = c("<10.000 ha", ">10.000 ha")) +
  theme_AP()

bottom <- plot_grid(a, b, ncol = 1, labels = c("c", "d"))
```

```
## Warning: Removed 74 rows containing non-finite values (stat_bin).
```

```
bottom
```

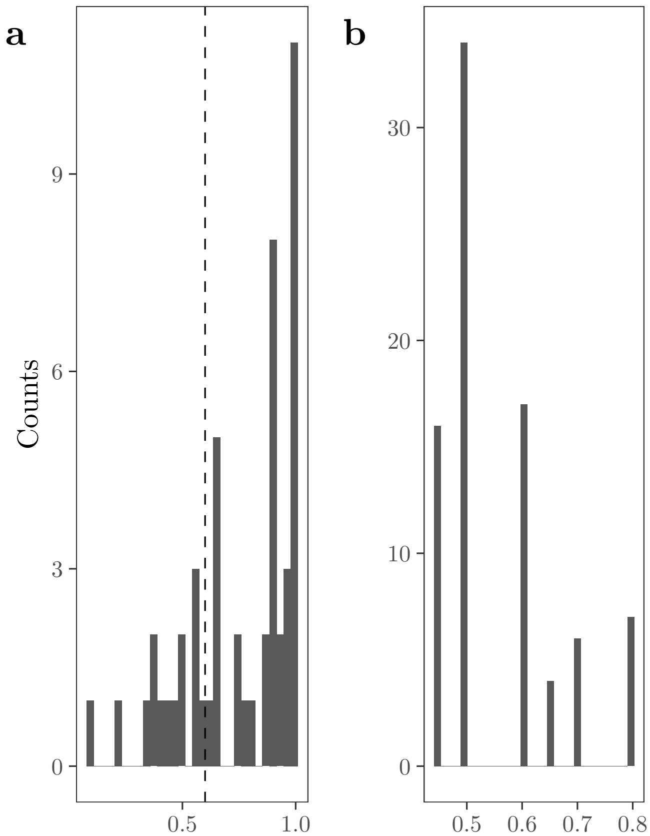




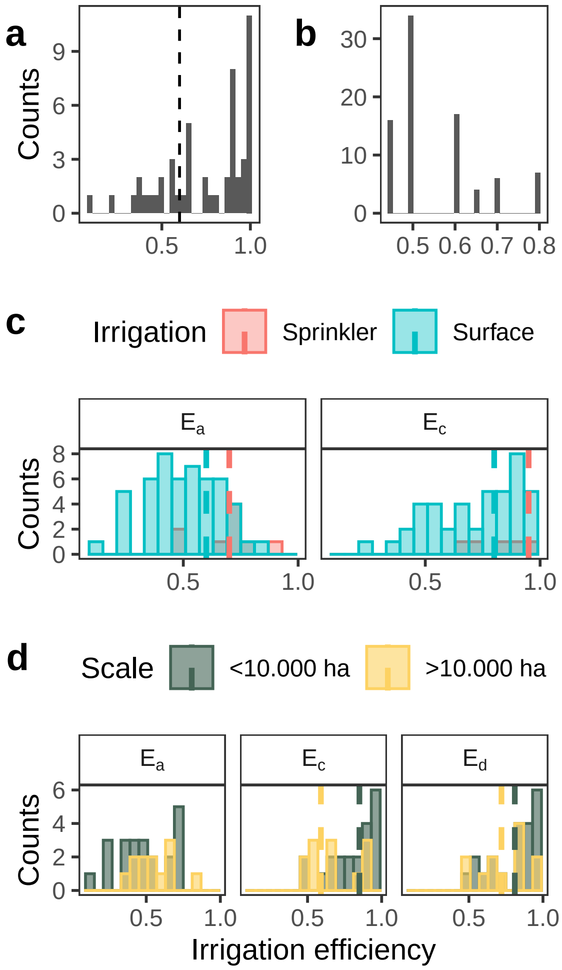

```
# PLOT USA AND AFRICA ----------------------------------------------------------

c1 <- ggplot(usa.dt, aes(Efficiency)) +
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_vline(xintercept = 0.6, lty = 2) +
  labs(x = "", y = "Counts") +
  theme_AP()

d1 <- ggplot(fao_dt, aes(Efficiency)) +
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  labs(x = "", y = "") +
  theme_AP()

top <- cowplot::plot_grid(c1, d1, ncol = 2, labels = "auto")
```

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

## Warning: Removed 3 rows containing non-finite values (stat_bin).

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

```
top
```




```
# PLOT MERGED ------------------------------------------------------------------

plot_grid(top, bottom, ncol = 1, rel_heights = c(0.3, 0.7))
```



2 The model

2.1 Function to create sample matrix

```
# CREATE FUNCTION TO DESIGN SAMPLE MATRIX -------------------------------------------

params_algo <- list(
  "Surface" = c("Ea_surf", "Ec_surf", "m", "r_L", "X1", "X2"),
  "Sprinkler" = c("Ea_sprinkler", "Ec_sprinkler", "m"),
  "Micro" = c("Ea_micro", "Ec_micro", "m"),
  "Mixed" = c("Ea_surf", "Ea_sprinkler", "Ec_surf", "Ec_sprinkler", "m", "r_L", "X1", "X2"),
  "Jager" = c("Ea_surf", "Ea_sprinkler", "Ec_surf", "Ec_sprinkler",
              "Ea_micro", "Ec_micro", "m", "r_L", "X1", "X2")
)

params_fun <- function(IFT) {
  out <- params_algo[[IFT]]
  return(out)
}
```

```r
sample_matrix_fun <- function(IFT) {
  params <- params_fun(IFT = IFT)
  mat <- sensobol::sobol_matrices(N = N, params = params)
  out <- list(params, mat)
  names(out) <- c("parameters", "matrix")
  return(out)
}
```

2.2 Define distributions

```r
# DEFINE TRUNCATED DISTRIBUTIONS ---------------------------------------------------

# EA SURFACE ---------

Ea.surface <- bos[Irrigation == "Surface"][, .(min = min(E_a, na.rm = TRUE),
                                               max = max(E_a, na.rm = TRUE))]
shape <- 3.502469
scale <- 0.5444373
minimum <- Ea.surface$min
maximum <- Ea.surface$max
weibull_dist <- sapply(c(minimum, maximum), function(x)
  pweibull(x, shape = shape, scale = scale))

# EC SURFACE --------

Ec.surface <- bos[Irrigation == "Surface"][, .(min = min(E_c, na.rm = TRUE),
                                               max = max(E_c, na.rm = TRUE))]
shape1 <- 5.759496
shape2 <- 1.403552
minimum.beta <- Ec.surface$min
maximum.beta <- Ec.surface$max
beta_dist <- sapply(c(minimum.beta, maximum.beta), function(x)
  pbeta(x, shape1 = shape1, shape2 = shape2))

# EA SPRINKLER --------

Ea.sprinkler <- bos[Irrigation == "Sprinkler"][, .(min = min(E_a, na.rm = TRUE),
                                               max = max(E_a, na.rm = TRUE))]
shape.spr <- 6.9913711
scale.spr <- 0.7451178
minimum.spr <- Ea.sprinkler$min
maximum.spr <- Ea.sprinkler$max
weibull_dist_spr <- sapply(c(minimum.spr, maximum.spr), function(x)
  pweibull(x, shape = shape.spr, scale = scale.spr))

# MANAGEMENT FACTOR (m) -----
```

```r
shape1.m <- 5.759496
shape2.m <- 1.403552
minimum.m <- 0.65
maximum.m <- 1
beta_dist.m <- sapply(c(minimum.m, maximum.m), function(x)
  pbeta(x, shape1 = shape1.m, shape2 = shape2.m))
```

```r
# FUNCTION TO TRANSFORM TO APPROPRIATE DISTRIBUTIONS -------------------------------

distributions_fun <- list(

  # SURFACE IRRIGATION
  # ---------------------------

  "Ea_surf" = function(x) {

    out <- qunif(x, weibull_dist[[1]], weibull_dist[[2]])
    out <- qweibull(out, shape, scale)
  },

  "Ec_surf" = function(x)  {

    out <- qunif(x, beta_dist[[1]], beta_dist[[2]])
    out <- qbeta(out, shape1, shape2)
  },

  # SPRINKLER IRRIGATION
  # --------------------------

  "Ea_sprinkler" = function(x) {

    out <- qunif(x, weibull_dist_spr[[1]], weibull_dist_spr[[2]])
    out <- qweibull(out, shape.spr, scale.spr)
  },

  "Ec_sprinkler" = function(x) qunif(x, 0.64, 0.96),

  # MICRO (DRIP) IRRIGATION
  # --------------------------

  "Ea_micro" = function(x) out <- qunif(x, 0.75, 0.95),
  "Ec_micro" = function(x) out <- qunif(x, 0.9, 0.95),

  # PROPORTION LARGE
  # -------------------------
  "Proportion_large" = function(x) x,
```

```r
  # MANAGEMENT FACTOR
  # -------------------------

  "m" = function(x) {
    out <- qunif(x, beta_dist.m[[1]], beta_dist.m[[2]])
    out <- qbeta(out, shape1.m, shape2.m)
    },

  # REDUCTION IN MANAGEMENT FACTOR DUE TO LARGE-SCALE
  # ---------------------------
  "r_L" = function(x) qunif(x, 0, 0.5),

  # IRRIGATED AREA DATASET
  # ---------------------------

  "X1" = function(x) floor(x * (4 - 1 + 1)) + 1,

  # THRESHOLD FOR LARGE-SCALE IRRIGATED AREAS
  # ----------------------------

  "X2" = function(x) floor(x * (5 - 1 + 1)) + 1
)
```

2.3 Uncertainty in the proportion of large-scale irrigated areas

```r
# DEFINE THE UNCERTAINTY IN THE LARGE FRACTION AT THE COUNTRY LEVEL ----------------

rohwer.frac <- rohwer[, .(Country, Large_fraction)]
rohwer.frac[, `:=` (min = Large_fraction, max = Large_fraction + 0.1)]

countries.list <- split(rohwer.frac, seq(nrow(rohwer.frac)))
names(countries.list) <- rohwer$Country

n.rows <- nrow(rohwer)
triggers.dt <- rohwer[rep(seq_len(n.rows), 4 * 5)][, .(Country, IFT)] %>%
  .[, X1:= rep(1:4, each = n.rows, times = 5)] %>%
  .[, X2:= rep(1:5, each = n.rows, times = 4)] %>%
  .[, Proportion_large:= runif(n.rows * 4 * 5)] %>%
  .[, index:= paste(Country, X1, X2, sep = ".")]

triggers.dt <- setkey(triggers.dt, index)
```

2.4 Function to create sample matrix and transfrom to appropriate distributions

```r
# FULL ALGORITHM TO CREATE SAMPLE MATRIX ----------------------------------------
```

```
full_sample_matrix <- function(IFT, Country) {
  tmp <- sample_matrix_fun(IFT = IFT)
  mat <- tmp[["matrix"]]
  temp <- colnames(mat)
  mat <- sapply(seq_along(temp), function(x) distributions_fun[[temp[x]]](mat[, x]))
  colnames(mat) <- temp
  countries.frac <- countries.list[[Country]]
  out <- list(tmp$parameters, mat)
  names(out) <- c("parameters", "matrix")
  return(out)
}
```

2.5 Run the model

```
# FULL MODEL ----------------------------------------------------------------------

full_model <- function(IFT, Country, sample.size, R) {

  country.differences <- setdiff(rohwer$Country, jager$Country)
  tmp <- full_sample_matrix(IFT = IFT, Country = Country)
  mat <- tmp$matrix

  if(IFT == "Surface" | IFT == "Mixed" | IFT == "Jager") {
    X1 <- mat[, "X1"]
    X2 <- mat[, "X2"]
    index <- paste(Country, X1, X2, sep = ".")
    Proportion_large <- triggers.dt[index][, Proportion_large]
  }

  if(IFT == "Surface") {

    Mf <- mat[, "m"] - mat[, "r_L"] * Proportion_large
    y <- mat[, "Ea_surf"] * mat[, "Ec_surf"] * Mf

  } else if(IFT == "Sprinkler") {

    Mf <- mat[, "m"]
    y <- mat[, "Ea_sprinkler"] * mat[, "Ec_sprinkler"] * Mf

  } else if(IFT == "Mixed") {

    Mf.surf <- mat[, "m"] - mat[, "r_L"] * Proportion_large
    y.surf <- mat[, "Ea_surf"] * mat[, "Ec_surf"] * Mf.surf

    Mf.sprink <- mat[, "m"]
    y.sprink <- mat[, "Ea_sprinkler"] * mat[, "Ec_sprinkler"] * Mf.sprink
```

```
    y <- 0.5 * y.surf + mat[, "r_L"] * y.sprink

  } else if(IFT == "Micro") {

    Mf <- mat[, "m"]
    y <- mat[, "Ea_micro"] * mat[, "Ec_micro"] * Mf

  } else if(IFT == "Jager") {

    if(Country %in% country.differences == TRUE) {
      next
    }

    Mf.surf <- mat[, "m"] - mat[, "r_L"] * Proportion_large
    y.surf <- mat[, "Ea_surf"] * mat[, "Ec_surf"] * Mf.surf

    Mf.spr <- mat[, "m"]
    y.spr <- mat[, "Ea_sprinkler"] * mat[, "Ec_sprinkler"] * Mf.spr

    Mf.micro <- mat[, "m"]
    y.micro <- mat[, "Ea_micro"] * mat[, "Ec_micro"] * Mf.micro

    y <- jager.list[[Country]]$Surface_fraction * y.surf +
      jager.list[[Country]]$Sprinkler_fraction * y.spr +
      jager.list[[Country]]$Drip_fraction * y.micro

  }

  ind <- sobol_indices(N = sample.size, Y = y, params = tmp$parameters,
                       boot = TRUE, R = R)
  out <- list(y, ind)
  names(out) <- c("output", "indices")
  return(out)
}
```

2.6 Define settings

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

N <- 2^14
R <- 10^2
```

2.7 Run model

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

new.rohwer <- rohwer[Country %in% jager$Country][, IFT:= "Jager"]
```

```r
all.dt <- list(rohwer, new.rohwer)

y <- list()
for(j in 1:length(all.dt)) {
  y[[j]] <- mclapply(1:nrow(all.dt[[j]]), function(x)
    full_model(IFT = all.dt[[j]][[x, "IFT"]],
               Country = all.dt[[j]][[x, "Country"]],
               sample.size = N,
               R = R),
    mc.cores = detectCores() * 0.75)
}
```

2.8 Extract model output

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

names(y) <- c("Rohwer et al. 2007", "Jägermeyr et al. 2015")

output <- tmp <- list()
for(i in names(y)) {
  output[[i]] <- lapply(y[[i]], function(x) x[["output"]][1:(2 * N)])

  if(i == "Rohwer et al. 2007") {

    names(output[[i]]) <- rohwer$Country

  } else if(i == "Jägermeyr et al. 2015") {

    names(output[[i]]) <- new.rohwer$Country

  }
  tmp[[i]] <- lapply(output[[i]], data.table) %>%
    rbindlist(., idcol = "Country")

  if(i == "Rohwer et al. 2007") {

    tmp[[i]] <- merge(tmp[[i]], rohwer[, .(Country, IFT)], all.x = TRUE) %>%
      .[, IFT:= factor(IFT, levels = c("Surface", "Sprinkler", "Micro", "Mixed"))]

  } else if(i == "Jägermeyr et al. 2015") {

    tmp[[i]] <- tmp[[i]][, IFT:= "Jager"]

  }

  tmp[[i]] <- tmp[[i]][, Continent:= countrycode(tmp[[i]][, Country],
                                                 origin = "country.name",
```

```r
                                              destination = "continent")]
}
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,

## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

```r
uncertainty.dt <- rbindlist(tmp, idcol = "Approach")
uncertainty.dt <- uncertainty.dt[, Study:= ifelse(IFT == "Jager",
                                                 "The proportion of IFTs is known",
                                                 "The proportion of IFTs is not known")]
```

```r
# EXPORT UNCERTAINTY IN IRRIGATION EFFICIENCY ------------------------------------

fwrite(uncertainty.dt, "uncertainty.dt.csv")
```

```r
# COMPUTE RANGES ------------------------------------------------------------------

calc <- uncertainty.dt[, .(min = min(V1), max = max(V1)), .(Continent, Country)] %>%
  .[, .(range = max - min), .(Continent, Country)] %>%
  .[order(range)]

print(calc, n = Inf)
```

```
##      Continent                  Country     range
##   1:      Asia                   Cyprus 0.4679420
##   2:      Asia     United Arab Emirates 0.4900561
##   3:      Asia                   Israel 0.4944147
##   4:      Asia                   Jordan 0.5149145
##   5:    Europe                  Austria 0.6014893
##   6:    Europe                  Belgium 0.6014893
##   7:      <NA>                 Byelarus 0.6014893
##   8:    Europe                  Denmark 0.6014893
##   9:    Europe                  Finland 0.6014893
##  10:    Europe                  Germany 0.6014893
##  11:    Europe                   Greece 0.6014893
##  12:    Europe                  Hungary 0.6014893
##  13:    Europe                   Latvia 0.6014893
##  14:    Europe                Lithuania 0.6014893
##  15:    Europe               Luxembourg 0.6014893
##  16:    Africa                   Malawi 0.6014893
##  17:    Europe                 Slovakia 0.6014893
##  18:    Europe                   Sweden 0.6014893
##  19:    Europe              Switzerland 0.6014893
##  20:    Europe                  Ukraine 0.6147176
##  21:    Europe           Czech Republic 0.6477047
##  22:    Europe                   Russia 0.6622102
##  23:    Europe                  Croatia 0.6645827
##  24:    Europe           United Kingdom 0.6680857
```

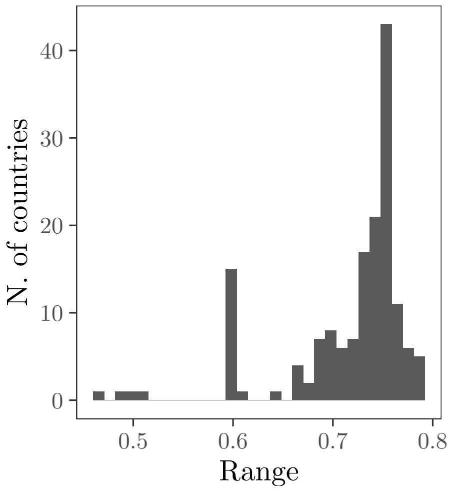
```
##  25:   Europe                Romania 0.6684038
##  26:   Africa              Swaziland 0.6796764
##  27: Americas                   Cuba 0.6796800
##  28:   Africa                  Benin 0.6838233
##  29: Americas          United States 0.6859862
##  30:     Asia                 Brunei 0.6879111
##  31:     Asia                Lebanon 0.6892404
##  32:   Europe            Netherlands 0.6896682
##  33:     Asia           Saudi Arabia 0.6898575
##  34:     Asia             Kazakhstan 0.6920644
##  35:   Africa             Mozambique 0.6939102
##  36: Americas                 Brazil 0.6959479
##  37:   Europe                 France 0.6976147
##  38:   Africa                 Zambia 0.6985666
##  39:   Africa               Zimbabwe 0.6985699
##  40:   Africa                Namibia 0.7002246
##  41:   Africa           South Africa 0.7025134
##  42:   Africa                Tunisia 0.7025813
##  43:     Asia                  Qatar 0.7036842
##  44:     Asia                 Kuwait 0.7046397
##  45: Americas                 Canada 0.7077460
##  46:   Africa            Ivory Coast 0.7091386
##  47:   Europe                  Italy 0.7108935
##  48:   Africa           Burkina Faso 0.7115694
##  49:   Europe               Bulgaria 0.7147026
##  50:   Africa                Algeria 0.7182193
##  51:   Europe                  Spain 0.7190056
##  52: Americas          French Guiana 0.7191150
##  53:  Oceania              Australia 0.7218235
##  54:   Africa                Lesotho 0.7226186
##  55:     Asia                  Japan 0.7247774
##  56: Americas                Uruguay 0.7264713
##  57:     Asia                   Iraq 0.7277116
##  58:     Asia             Azerbaijan 0.7284226
##  59:   Africa                  Libya 0.7299519
##  60:   Africa                  Zaire 0.7306228
##  61:   Africa               Botswana 0.7322712
##  62:   Africa                  Gabon 0.7325094
##  63: Americas               Paraguay 0.7327599
##  64:     Asia                 Turkey 0.7330840
##  65:   Africa                 Uganda 0.7335162
##  66: Americas                Jamaica 0.7340006
##  67:     Asia                  Burma 0.7344695
##  68: Americas              Guatemala 0.7352743
##  69:   Africa               Tanzania 0.7353794
##  70:  Oceania       Papua New Guinea 0.7357704
##  71:     Asia                  Nepal 0.7364300
##  72:   Africa                  Congo 0.7364368
```

```
##  73:  Americas                  Ecuador 0.7379232
##  74:    Africa                    Kenya 0.7381012
##  75:  Americas                    Haiti 0.7389654
##  76:    Africa                     Chad 0.7398257
##  77:    Africa Central African Republic 0.7406275
##  78:  Americas                 Colombia 0.7406412
##  79:    Africa                   Gambia 0.7410900
##  80:  Americas              Puerto Rico 0.7411452
##  81:    Africa           Western Sahara 0.7412732
##  82:    Europe                 Portugal 0.7424699
##  83:  Americas              El Salvador 0.7424882
##  84:      Asia                    India 0.7430188
##  85:    Africa                 Cameroon 0.7430382
##  86:      Asia             Turkmenistan 0.7431978
##  87:    Africa                    Egypt 0.7434191
##  88:      Asia                    Syria 0.7437367
##  89:    Africa               Madagascar 0.7453378
##  90:  Americas               Costa Rica 0.7460634
##  91:      Asia                    Yemen 0.7464134
##  92:    Africa        Equatorial Guinea 0.7472551
##  93:    Europe                   Serbia 0.7476467
##  94:    Africa                  Liberia 0.7479554
##  95:  Americas                  Bolivia 0.7480052
##  96:    Africa                  Somalia 0.7480674
##  97:      Asia                     Laos 0.7481253
##  98:    Africa                 Ethiopia 0.7482251
##  99:    Africa                   Rwanda 0.7482431
## 100:      Asia               Kyrgyzstan 0.7484823
## 101:      Asia                   Bhutan 0.7487014
## 102:    Europe                   Norway 0.7488203
## 103:    Europe                   Poland 0.7495878
## 104:    Africa                     Mali 0.7501681
## 105:      Asia                Sri Lanka 0.7502131
## 106:      Asia                 Mongolia 0.7502820
## 107:  Americas                 Trinidad 0.7505246
## 108:    Africa            Guinea-Bissau 0.7507230
## 109:    Africa                   Guinea 0.7508012
## 110:      Asia                    China 0.7508179
## 111:    Africa               Mauritania 0.7508221
## 112:      Asia                Indonesia 0.7509219
## 113:      Asia                  Armenia 0.7511245
## 114:    Europe   Bosnia and Herzegovina 0.7513991
## 115:    Africa                  Morocco 0.7514743
## 116:      Asia                     Iran 0.7514788
## 117:  Americas                 Suriname 0.7515058
## 118:  Americas                     Peru 0.7516554
## 119:  Americas                Argentina 0.7521091
## 120:  Americas                    Chile 0.7522434
```

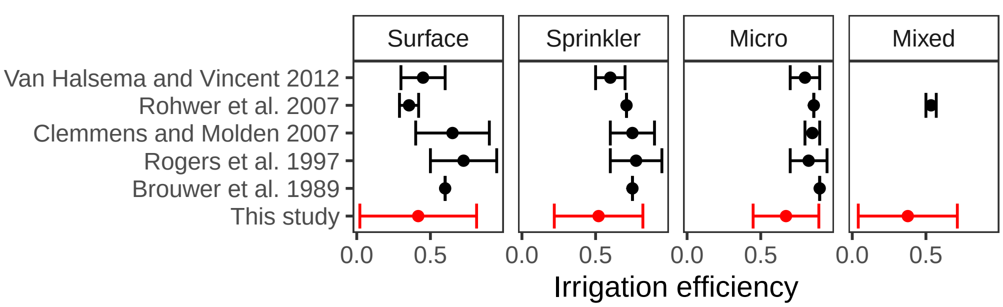
```
## 121:    Africa            Sierra Leone 0.7525826
## 122:    Africa                   Niger 0.7526578
## 123:    Africa                  Angola 0.7530253
## 124:    Africa                 Burundi 0.7531444
## 125:  Americas                  Belize 0.7533846
## 126:      Asia             South Korea 0.7544530
## 127:  Americas                  Mexico 0.7544977
## 128:      Asia                Malaysia 0.7545882
## 129:    Africa                Djibouti 0.7546681
## 130:  Americas                Honduras 0.7550064
## 131:    Africa                    Togo 0.7550732
## 132:    Africa                 Eritrea 0.7555949
## 133:    Europe                 Moldova 0.7563886
## 134:      Asia                Cambodia 0.7569788
## 135:    Africa                 Senegal 0.7574388
## 136:    Europe               Macedonia 0.7578519
## 137:    Africa                 Nigeria 0.7589399
## 138:    Africa                   Sudan 0.7593515
## 139:      Asia                Thailand 0.7607690
## 140:      Asia                Pakistan 0.7615949
## 141:  Americas               Nicaragua 0.7622110
## 142:      Asia                 Vietnam 0.7641254
## 143:  Americas               Venezuela 0.7645259
## 144:  Americas      Dominican Republic 0.7645690
## 145:    Europe                 Albania 0.7647019
## 146:      Asia              Uzbekistan 0.7672137
## 147:      Asia              Tajikistan 0.7680949
## 148:      Asia                 Georgia 0.7709388
## 149:      Asia              Bangladesh 0.7714909
## 150:  Americas                  Guyana 0.7752999
## 151:      Asia             Philippines 0.7759418
## 152:  Americas                  Panama 0.7774470
## 153:    Europe                Slovenia 0.7784899
## 154:      Asia             North Korea 0.7820092
## 155:    Africa                   Ghana 0.7850875
## 156:      Asia             Afghanistan 0.7865884
## 157:      Asia                    Oman 0.7878713
## 158:   Oceania             New Zealand 0.7892427
##       Continent                 Country     range
```

```r
ggplot(calc, aes(range)) +
  geom_histogram() +
  labs(x = "Range", y = "N. of countries") +
  theme_AP()
```

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




```
# COMPARE RANGES ---------------------------------------------------------------

ranges_empirical <- uncertainty.dt[, .(higher = max(V1), lower = min(V1)), IFT] %>%
  .[, Study:= "This study"]%>%
  .[!IFT == "Jager"]

ranges_efficiencies <- fread("ranges_efficiencies.csv")

rbind(ranges_empirical, ranges_efficiencies)[, mean.value:= (higher + lower) / 2] %>%
  .[, Study:= factor(Study, levels = c("This study",
                                       "Brouwer et al. 1989",
                                       "Rogers et al. 1997",
                                       "Clemmens and Molden 2007",
                                       "Rohwer et al. 2007",
                                       "Van Halsema and Vincent 2012"))] %>%
  na.omit() %>%
  ggplot(., aes(mean.value, Study, color = ifelse(Study == "This study", "red", "black"))) +
  geom_point() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_errorbar(aes(xmin = lower, xmax = higher)) +
  scale_color_identity() +
  facet_wrap(~IFT, ncol = 4) +
  labs(x = "Irrigation efficiency", y = "") +
  theme_AP()
```



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

dd <- uncertainty.dt[!Continent == "Oceania"][Study == "The proportion of IFTs is not known"] %
  split(., .$Continent, drop = TRUE)

overlap.dt <- lapply(dd, function(x) split(x, x$IFT, drop = TRUE)) %>%
  lapply(., function(x) lapply(x, function(y) y[, V1])) %>%
  lapply(., function(x) overlap(x)$OV)

overlap.dt
```

```
## $Africa
## Surface-Sprinkler     Surface-Mixed   Sprinkler-Mixed
##         0.2341197         0.7482128         0.1513037
##
## $Americas
## Surface-Mixed
##     0.7622839
##
## $Asia
## Surface-Micro Surface-Mixed   Micro-Mixed
##    0.04502982    0.72015229    0.01232303
##
## $Europe
## Surface-Sprinkler     Surface-Mixed   Sprinkler-Mixed
##         0.2404475         0.7435392         0.1536190
```

```r
ff <- uncertainty.dt[!Continent == "Oceania"] %>%
  .[Country %in% intersect(rohwer[, Country], jager[, Country])] %>%
  split(., .$Country, drop = TRUE) %>%
  lapply(., function(x) split(x, x$Approach, drop = TRUE)) %>%
  lapply(., function(x) lapply(x, function(y) y[, V1])) %>%
  lapply(., function(x) overlap(x)$OV) %>%
  lapply(., data.table) %>%
  rbindlist(., idcol = "Country") %>%
  .[, Continent:= countrycode(.[, Country],
                             origin = "country.name",
                             destination = "continent")]

list_continents <- list(c("Africa", "Asia"), c("Americas", "Europe"))

# PLOT OVERLAP ------------

dd <- list()
for(i in 1:length(list_continents)) {
  dd[[i]] <- ff[Continent %in% list_continents[[i]]] %>%
    ggplot(., aes(reorder(Country, V1), V1)) +
    geom_point() +
```

```
    scale_color_discrete(name = "GM") +
    labs(y = "Overlap", x = "") +
    facet_wrap(~Continent, scales = "free_y") +
    coord_flip() +
    theme_AP()
}

dd
```

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


```
##
## [[2]]
```

3 Uncertainty analysis

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

gg <- list()
for (i in 1:length(list_continents)) {
  gg[[i]] <- ggplot(uncertainty.dt[Continent %in% list_continents[[i]]],
                    aes(x = V1, y = fct_reorder(Country, V1), fill = Study)) +
    geom_density_ridges(scale = 2, alpha = 0.3) +
    labs(x = "Irrigation efficiency", y = "") +
    facet_wrap(~Continent, scales = "free") +
    scale_x_continuous(breaks = pretty_breaks(n = 3),
```

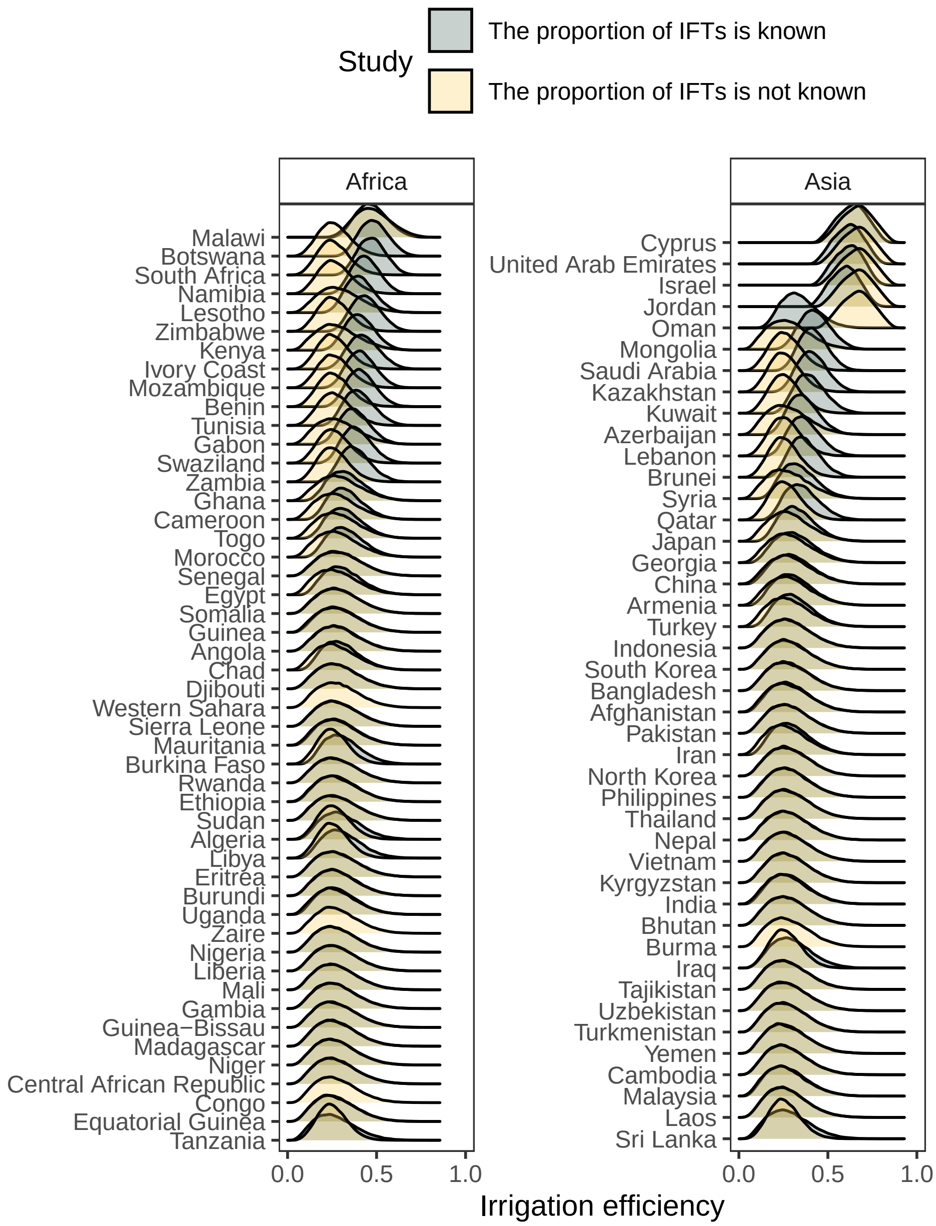
```r
                        limits = c(0, 1)) +
    scale_fill_manual(values = wes_palette("Chevalier1")) +
    theme_AP() +
    theme(legend.position = "top") +
    guides(fill = guide_legend(nrow = 2, byrow = TRUE))
}
```

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

gg
```

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

```
## Picking joint bandwidth of 0.0117
```

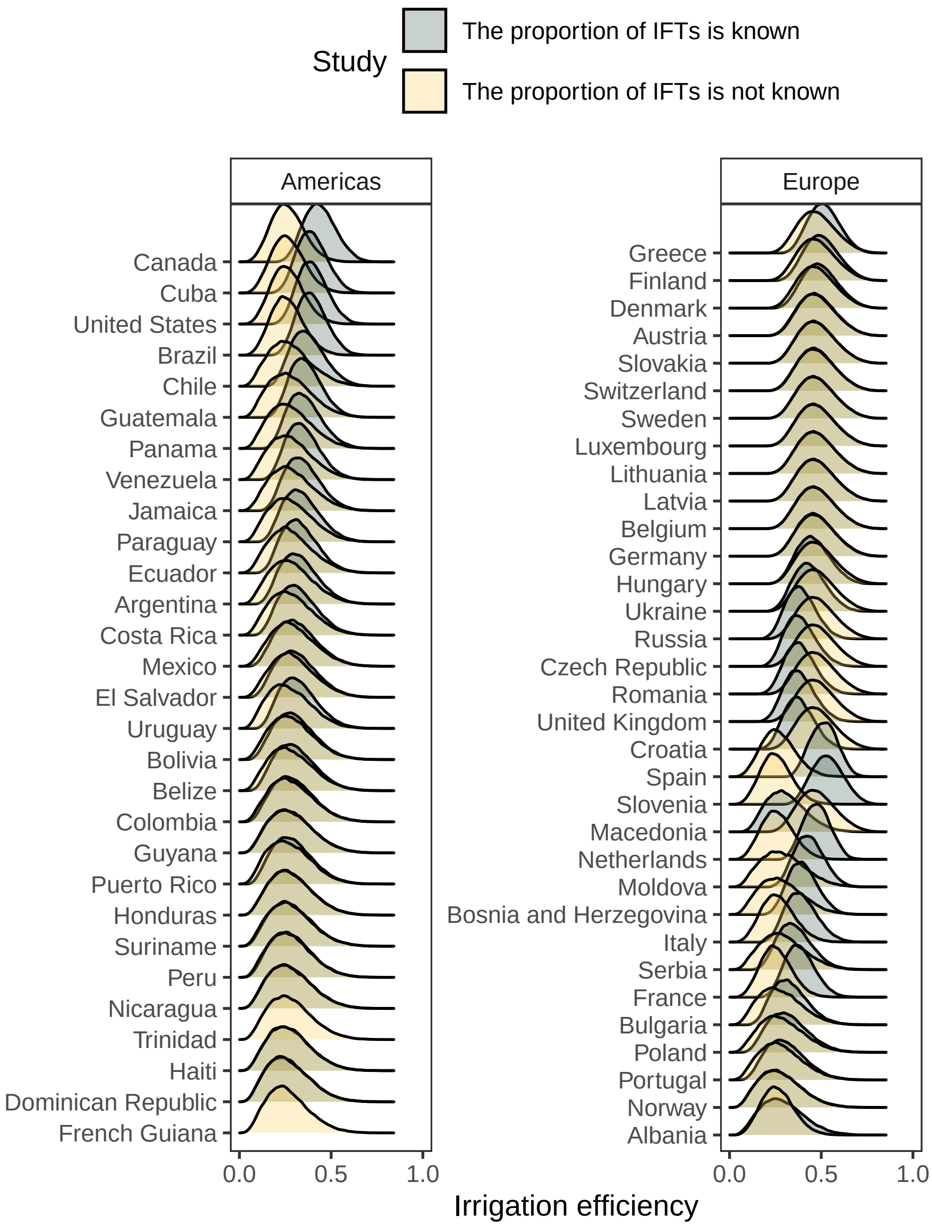
```
## Picking joint bandwidth of 0.0118
```




```
##
## [[2]]
```

```
## Picking joint bandwidth of 0.012
```

```
## Picking joint bandwidth of 0.0108
```

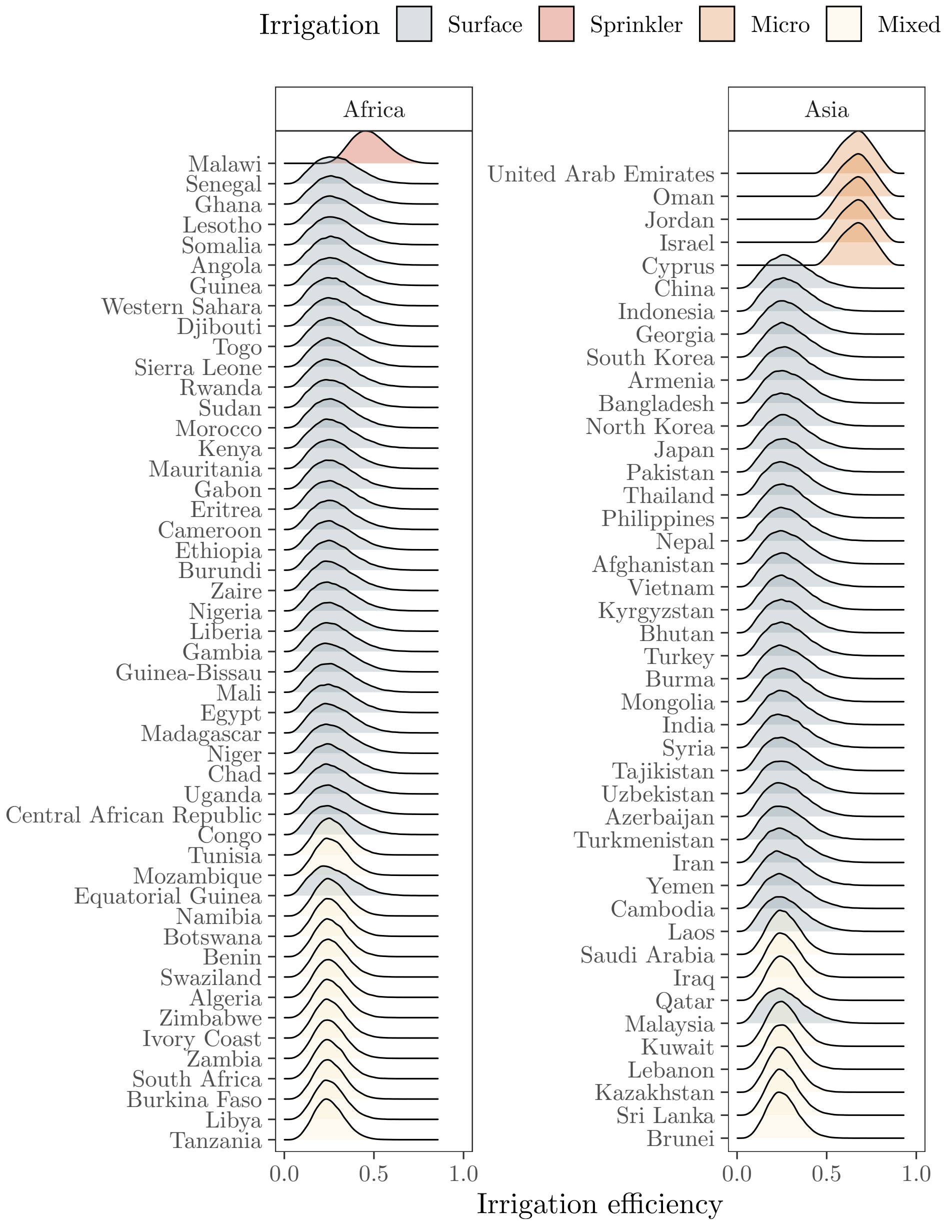



```
# PLOT UNCERTAINTY IN EACH IRRIGATION TECHNOLOGY -----------------------------------

gg <- list()

for(i in 1:length(list_continents)) {
  gg[[i]] <- uncertainty.dt[Approach == "Rohwer et al. 2007"][Continent %in% list_continents[[i
    ggplot(.,aes(x = V1, y = fct_reorder(Country, V1), fill = IFT)) +
    geom_density_ridges(scale = 2, alpha = 0.3) +
    labs(x = "Irrigation efficiency", y = "") +
    facet_wrap(~Continent, scales = "free") +
    scale_x_continuous(breaks = pretty_breaks(n = 3),
                       limits = c(0, 1)) +
    theme_AP() +
```

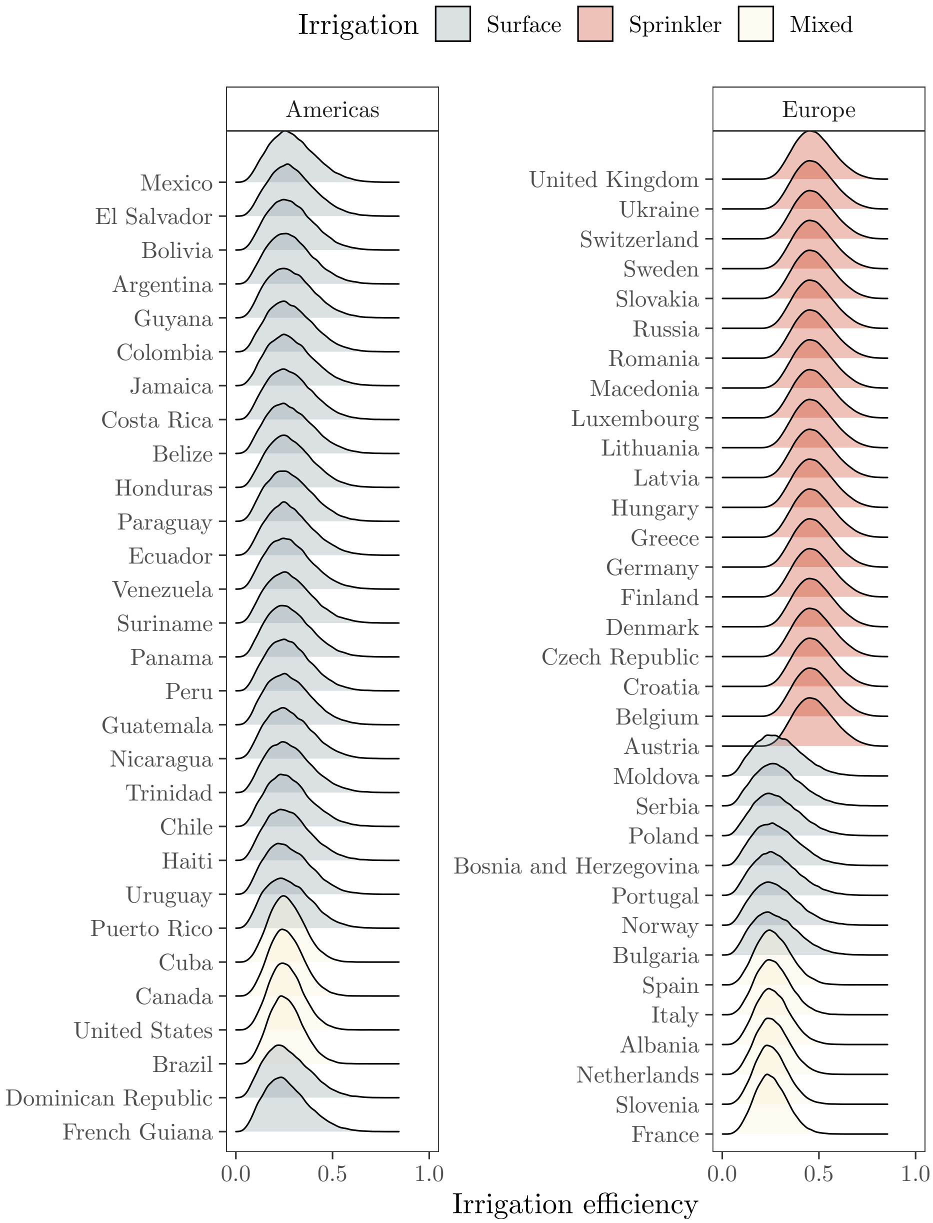
```
    theme(legend.position = "top")
  if(i == 1) {
    gg[[i]] <- gg[[i]] +
      scale_fill_manual(values = c("#899DA4", "#C93312", "#DC863B", "#FAEFD1"),
                        labels = c("Surface", "Sprinkler", "Micro", "Mixed"),
                        name = "Irrigation")
  } else if(i == 2) {
    gg[[i]] <- gg[[i]] +
      scale_fill_manual(values = c("#899DA4", "#C93312", "#FAEFD1"),
                        labels = c("Surface", "Sprinkler", "Mixed"),
                        name = "Irrigation")
  }
}

gg
```

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

```
## Picking joint bandwidth of 0.012
## Picking joint bandwidth of 0.012
```




```
##
## [[2]]

## Picking joint bandwidth of 0.0126

## Picking joint bandwidth of 0.0113
```



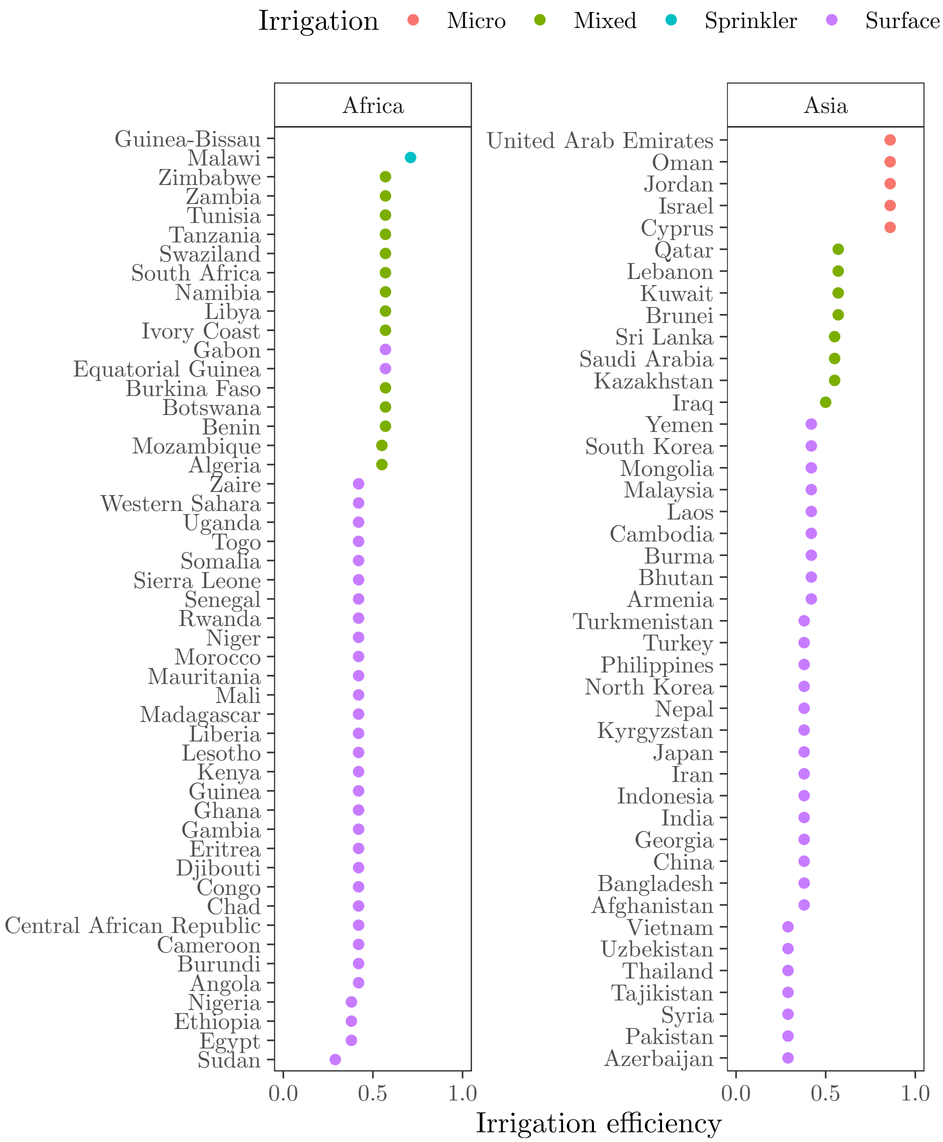

```
# PLOT ROHWER ET AL.'S IRRIGATION EFFICIENCY VALUES --------------------------------

rohwer[, Continent:= countrycode(rohwer[, Country], origin = "country.name",
                                 destination = "continent")]
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

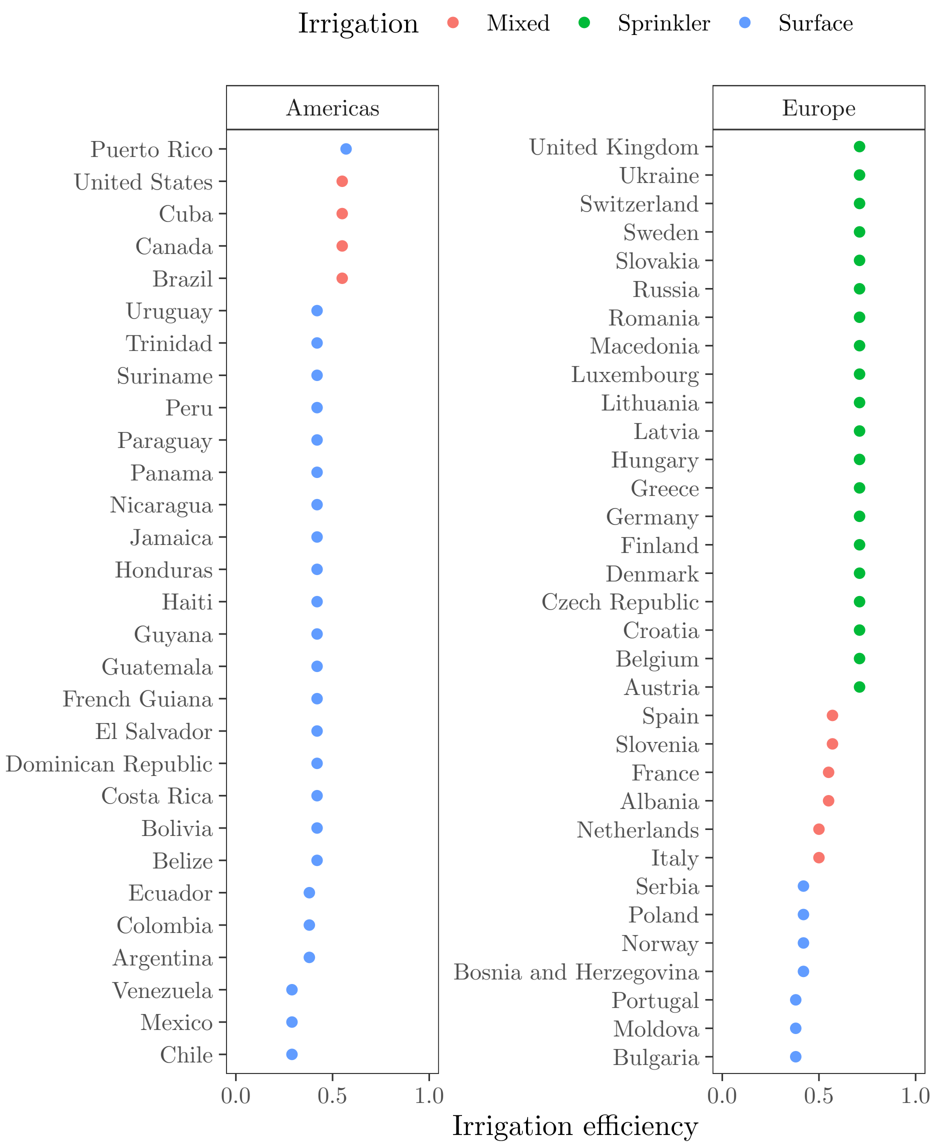
```
dd <- list()
for (i in 1:length(list_continents)) {
  dd[[i]] <- ggplot(rohwer[Continent %in% list_continents[[i]]],
                    aes(x = Ep, y = fct_reorder(Country, Ep), color = IFT)) +
    geom_point() +
    labs(x = "Irrigation efficiency", y = "") +
    scale_x_continuous(breaks = pretty_breaks(n = 3),
```

```
                        limits = c(0, 1)) +
    facet_wrap(~Continent, scales = "free") +
    scale_color_discrete(name = "Irrigation") +
    theme_AP()
}

dd
```

```
## [[1]]

## Warning: Removed 1 rows containing missing values (geom_point).
```

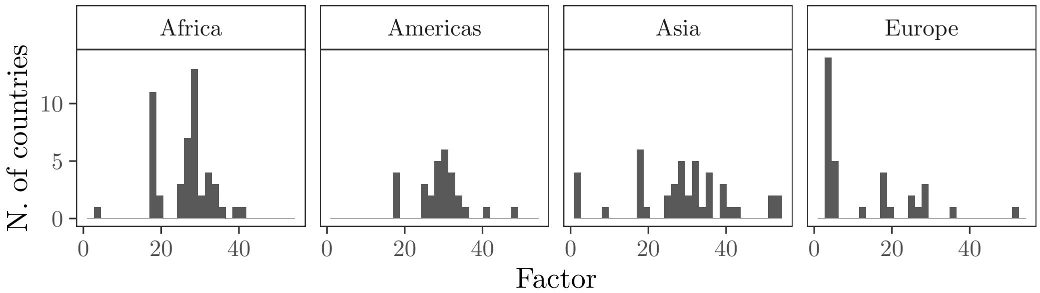



```
##
## [[2]]
```




```
# CALCULATE THE UNCERTAINTY IN THE RANGES -------------------------------------------

selection_continents <- c("Africa", "Asia", "Americas", "Europe")

factor_unc <- uncertainty.dt[, .(min = min(V1), max = max(V1)), .(Continent, Country)] %>%
  .[Continent %in% selection_continents] %>%
  .[, factor:= max / min]

ggplot(factor_unc, aes(factor)) +
  geom_histogram() +
  facet_wrap(~Continent, ncol = 4) +
  labs(x = "Factor", y = "N. of countries") +
  theme_AP()
```

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




```r
# Number of countries whose irrigation water withdrawals fluctuate a factor of x
# due to uncertainty in irrigation efficiency
factor_unc %>%
  .[, factor:= floor(max / min)] %>%
  .[, .(number.countries = .N), factor] %>%
  .[order(factor)] %>%
  print()
```

```
##     factor number.countries
##  1:      2                4
##  2:      3               15
##  3:      4                1
##  4:      5                4
##  5:      8                1
##  6:     12                1
##  7:     17               10
##  8:     18               18
##  9:     20                1
## 10:     24                6
## 11:     25                4
## 12:     26                8
## 13:     27               11
## 14:     28               14
## 15:     29               10
## 16:     30                5
## 17:     31                8
## 18:     32                6
## 19:     33                4
## 20:     34                3
## 21:     35                5
## 22:     36                1
## 23:     38                1
## 24:     39                3
## 25:     40                2
## 26:     41                1
## 27:     42                1
## 28:     47                1
## 29:     51                1
## 30:     52                3
```

```
## 31:     53                 1
##     factor number.countries
```

3.1 Retrieve data from ISIMIP

```
# FUNCTIONS TO EXTRACT DATA FROM .NC FILES ------------------------------------------

coords2country = function(points) {
  countriesSP <- rworldmap::getMap(resolution = 'low')
  pointsSP = sp::SpatialPoints(points, proj4string=CRS(proj4string(countriesSP)))
  indices = sp::over(pointsSP, countriesSP)
  indices$ADMIN
}

# Function to load and extract data from .nc files from ISIMIP
open_nc_files <- function(file, dname, selected.years, vec) {
  ncin <- nc_open(file)
  # get longitude, latitude, time
  lon <- ncvar_get(ncin, "lon")
  lat <- ncvar_get(ncin, "lat")
  # Get variable
  tmp_array <- ncvar_get(ncin, dname)
  m <- lapply(selected.years, function(x) vec[[x]])

  out <- lapply(m, function(x) {
    tmp_slice <- lapply(x, function(y) tmp_array[, , y])
    # create dataframe -- reshape data
    # matrix (nlon*nlat rows by 2 cols) of lons and lats
    lonlat <- as.matrix(expand.grid(lon,lat))
    # vector of `tmp` values
    tmp_vec <- lapply(tmp_slice, function(x) as.vector(x))
    # create dataframe and add names
    tmp_df01 <- lapply(tmp_vec, function(x) data.frame(cbind(lonlat, x)))
    names(tmp_df01) <- x
    da <- lapply(tmp_df01, data.table) %>%
      rbindlist(., idcol = "month") %>%
      na.omit()
    # Convert coordinates to country
    Country <- coords2country(da[1:nrow(da), 2:3])
    df <- cbind(Country, da)
    setDT(df)
    out <- na.omit(df)[, .(Water.Withdrawn = sum(x)), Country]
    out[, Water.Withdrawn:= Water.Withdrawn * 10000]
    out[, Continent:= countrycode(out[, Country],
                                  origin = "country.name",
                                  destination = "continent")] %>%
      .[, Code:= countrycode(out[, Country],
```

```r
                              origin = "country.name",
                              destination = "un")] %>%
      .[, Country:= countrycode(out[, Code],
                                 origin = "un",
                                 destination = "country.name")] %>%
      .[!Continent == "Oceania"]
    setcolorder(out, c("Country", "Continent", "Code", "Water.Withdrawn"))
  })
  return(out)
}
```

```r
# READ IN NC FILES -----------------------------------------------------------------

# Define settings
vecs <- 1:((2010 - 1970) * 12)
vec <- split(vecs, ceiling(seq_along(vecs) / 12))
names(vec) <- 1971:2010
selected.years <- "2010"
dname <- "pirrww"

files <- list("h08_wfdei_nobc_hist_varsoc_co2_pirrww_global_monthly_1971_2010.nc",
              "pcr-globwb_wfdei_nobc_hist_varsoc_co2_pirrww_global_monthly_1971_2010.nc",
              "lpjml_wfdei_nobc_hist_varsoc_co2_pirrww_global_monthly_1971_2010.nc",
              "watergap2_wfdei_nobc_hist_varsoc_co2_pirrww_global_monthly_1971_2010.nc")

names.isimip <- c("H08", "PCR-GLOBWB", "LPJmL", "WaterGap")

isimip.dt <- mclapply(files, function(x)
  open_nc_files(file = x, dname = dname, selected.years = selected.years, vec = vec),
  mc.cores = detectCores() * 0.75)
```

```r
# EXTRACT CORRECTIVE COEFFICIENTS FOR IRRIGATION EFFICIENCY FOR LPJML -------------

ncin <- nc_open("irrigation_project_efficiencies.nc")
lon <- ncvar_get(ncin, "lon")
lat <- ncvar_get(ncin, "lat")
tmp_array <- ncvar_get(ncin)
lonlat <- as.matrix(expand.grid(lon,lat))
da <- na.omit(cbind(lonlat, as.vector(tmp_array))) %>%
  data.frame() %>%
  na.omit()
Country <- coords2country(da[1:nrow(da), 1:2])
lpjml_efficiencies <- cbind(Country, da) %>%
  na.omit() %>%
  data.table() %>%
  .[, .(Ep = mean(V3)), Country]
```

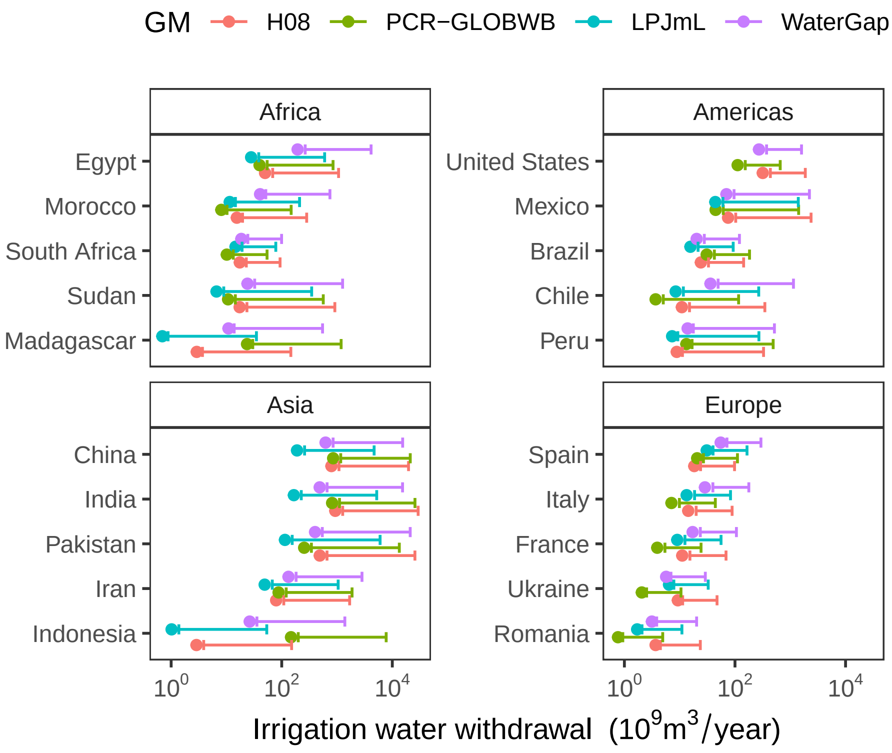
```r
# ARRANGE NC FILES ------------------------------------------------------------------

names(isimip.dt) <- names.isimip

isimip.dt <- lapply(isimip.dt, function(x) rbindlist(x)) %>%
  rbindlist(., idcol = "Model") %>%
  na.omit() %>%
  # To correct for duplicate country in Cyprus
  .[, .(Water.Withdrawn = mean(Water.Withdrawn)), .(Model, Country, Continent, Code)]

lpjml_harmonized <- merge(isimip.dt[Model == "LPJmL"], lpjml_efficiencies, all.x = TRUE) %>%
  .[, Water.Withdrawn:= Water.Withdrawn * Ep] %>%
  .[, Ep:= NULL]

isimip.dt <- rbind(isimip.dt[!Model == "LPJmL"], lpjml_harmonized)

fwrite(isimip.dt, "isimip.dt")
```

```r
# MERGE UNCERTAINTY IN EP WITH ISIMIP DATA ------------------------------------------

efficiency.dt <- copy(uncertainty.dt) %>%
  setnames(., "V1", "Ep")

ghm.dt <- dcast(isimip.dt, Country + Continent + Code ~ Model, value.var = "Water.Withdrawn")
full.dt <- merge(efficiency.dt, ghm.dt, by = c("Country", "Continent"), all.x = TRUE) %>%
  .[, (names.isimip):= lapply(.SD, function(x) x / Ep), .SDcols = names.isimip]
tmp.dt <- melt(full.dt, measure.vars = names.isimip, variable.name = "Model",
               value.name = "IWW_corrected")
ghm.large <- melt(ghm.dt, measure.vars = names.isimip, variable.name = "Model",
     value.name = "IWW")
gm.uncertainty <- tmp.dt[, .(min = min(IWW_corrected), max = max(IWW_corrected)),
                         .(Country, Continent, Model)]
gm.dt <- merge(ghm.large, gm.uncertainty)
```

```r
# PLOT UNCERTAINTY IN EP WITH ISIMIP DATA -------------------------------------------

countries_list <- c("Egypt", "Sudan", "South Africa", "Morocco", "Madagascar",
                    "United States", "Mexico", "Brazil", "Chile", "Peru",
                    "India", "China", "Pakistan", "Iran", "Indonesia",
                    "Italy", "Spain", "France", "Ukraine", "Romania")

gm.dt[Country %in% countries_list] %>%
  ggplot(., aes(reorder(Country, IWW), IWW, color = Model)) +
  geom_point(position = position_dodge(0.7)) +
  geom_errorbar(aes(ymin = min,
                    ymax = max),
                position = position_dodge(0.7)) +
  scale_y_log10(breaks = trans_breaks("log10", function(x) 10 ^ (2 * x)),
```

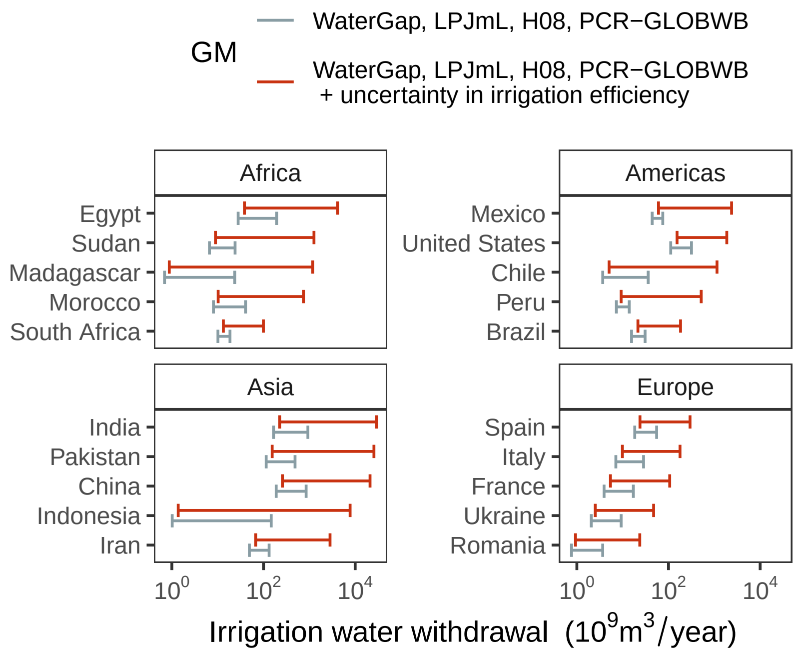
```
                 labels = trans_format("log10", math_format(10 ^ .x))) +
  scale_color_discrete(name = "GM") +
  labs(y = expression(paste("Irrigation water withdrawal ", " ", "(", 10^9, m^3/year, "", ")")
       x = "") +
  facet_wrap(~Continent, scales = "free_y") +
  coord_flip() +
  theme_AP()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
# PLOT RANGES OF STRUCTURAL UNCERTAINTY AND RANGES OF
# STRUCTURAL UNCERTAINTY + UNCERTAINTY IN IRRIGATION EFFICIENCY --------------------

range.gm <- gm.dt %>%
  .[, .(min = min(IWW, na.rm = TRUE), max = max(IWW, na.rm = TRUE)), .(Country, Continent)] %>%
  .[, Approach:= "WaterGap, LPJmL, H08, PCR-GLOBWB"]

range.study <- gm.dt %>%
  .[, .(min = min(min, na.rm = TRUE), max = max(max, na.rm = TRUE)), .(Country, Continent)] %>%
  .[, Approach:= "WaterGap, LPJmL, H08, PCR-GLOBWB \n + uncertainty in irrigation efficiency"]

rbind(range.gm, range.study) %>%
  .[Country %in% countries_list] %>%
  .[, mean:= (min + max) / 2] %>%
  ggplot(., aes(reorder(Country, mean), mean, color = Approach)) +
  geom_errorbar(aes(ymin = min,
                    ymax = max),
```

```
                position = position_dodge(0.7)) +
  scale_y_log10(breaks = trans_breaks("log10", function(x) 10 ^ (2 * x)),
                labels = trans_format("log10", math_format(10 ^ .x))) +
  scale_color_manual(name = "GM", values = wes_palette("Royal1")) +
  labs(y = expression(paste("Irrigation water withdrawal ", " ", "(", 10^9, m^3/year, "", ")")
       x = "") +
  facet_wrap(~Continent, scales = "free_y") +
  coord_flip() +
  theme_AP() +
  guides(color = guide_legend(nrow = 2, byrow = TRUE))
```




```
# PLOT RANGES OF STRUCTURAL UNCERTAINTY AND RANGES OF
# STRUCTURAL UNCERTAINTY + UNCERTAINTY IN IRRIGATION EFFICIENCY (COMPLETE) ---------

dd <- list()
for (i in 1:length(list_continents)) {
  dd[[i]] <- rbind(range.gm, range.study) %>%
    .[, mean:= (min + max) / 2] %>%
    .[Continent %in% list_continents[[i]]] %>%
    ggplot(., aes(reorder(Country, mean), mean, color = Approach)) +
    geom_errorbar(aes(ymin = min,
                      ymax = max),
                  position = position_dodge(0.7)) +
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10 ^ (2 * x)),
                  labels = trans_format("log10", math_format(10 ^ .x))) +
    scale_color_manual(name = "GM", values = wes_palette("Royal1")) +
    labs(y = expression(paste("Irrigation water withdrawal ", " ", "(", 10^9, m^3/year, "", ")")
         x = "") +
    facet_wrap(~Continent, scales = "free_y") +
```

```
    coord_flip() +
    theme_AP() +
    guides(color = guide_legend(nrow = 2, byrow = TRUE))
}

dd
```

```
## [[1]]

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
```

```
##
## [[2]]

## Warning: Transformation introduced infinite values in continuous y-axis
```


```
# PRINT RANGES -----------------------------------------------------------------

# print(range.study, n = Inf)
```

4 Sensitivity analysis

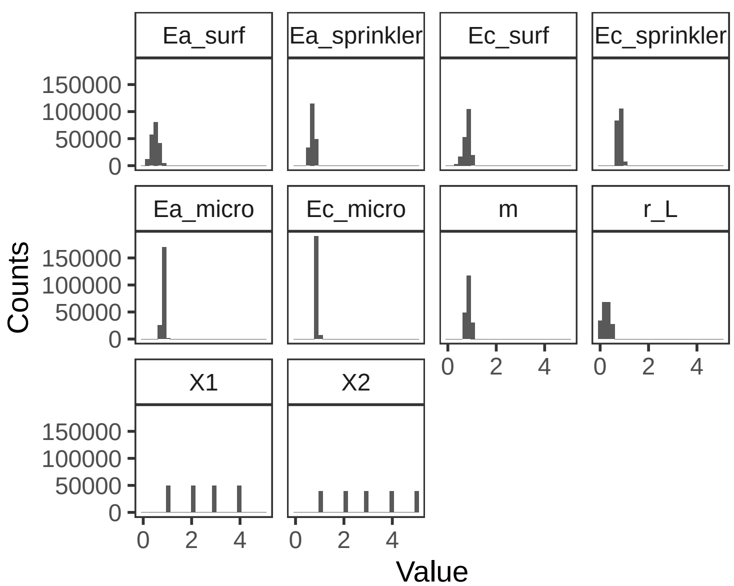
```
# SAMPLE MATRIX DISTRIBUTIONS ------------------------------------------------------

# Define labels
```

```r
label_facets <- c("Ea_surf" = "$E_{a_{su}}$",
                  "Ec_surf" = "$E_{c_{su}}$",
                  "Ea_sprinkler" = "$E_{a_{sp}}$",
                  "Ec_sprinkler" = "$E_{c_{sp}}$",
                  "Ea_micro" = "$E_{a_{mi}}$",
                  "Ec_micro" = "$E_{c_{mi}}$",
                  "Proportion_large" = "$f_L$",
                  "m" = "$m$",
                  "r_L" = "$r_L$")

mat <- data.table(full_sample_matrix(IFT = "Jager", Country = "Spain")$matrix)
mat <- mat[, Proportion_large:= NULL]
```

```
## Warning in `[.data.table`(mat, , `:=`(Proportion_large, NULL)): Column
## 'Proportion_large' does not exist to remove
```

```r
melt(mat, measure.vars = colnames(mat)) %>%
  ggplot(., aes(value)) +
  geom_histogram() +
  labs(x = "Value", y = "Counts") +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  facet_wrap(~variable) +
  theme_AP()
```

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



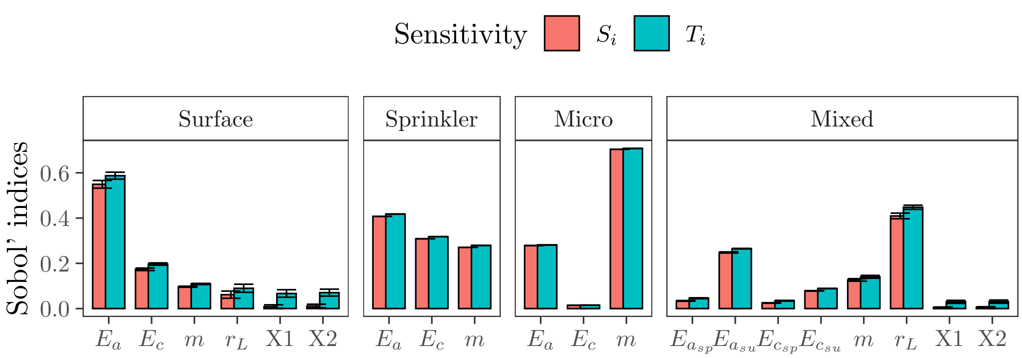

```r
# EXTRACT SOBOL' INDICES ----------------------------------------------------------

ind <- lapply(y$`Rohwer et al. 2007`, function(x) x[["indices"]]$results)
names(ind) <- rohwer$Country
ind <- rbindlist(ind, idcol = "Country")
```

```r
ind[, Continent:= countrycode(ind[, Country], origin = "country.name",
                              destination = "continent")]
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

```r
tmp.ift <- split(rohwer, rohwer$IFT)

out <- list()
for(i in names(tmp.ift)) {
  out[[i]] <- ind[Country %in% tmp.ift[[i]][, Country]]
}
```

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

ind.dt <- rbindlist(out, idcol = "IFT") %>%
  .[, IFT:= factor(IFT, levels = c("Surface", "Sprinkler", "Micro", "Mixed"))]

tmp <- ind.dt[, .(mean = mean(original), sd = sd(original)),
              .(sensitivity, parameters, IFT)]

tmp2 <- tmp[!IFT == "Mixed"][, parameters:= ifelse(parameters == "Ea_surf", "$E_a$",
                                                   ifelse(parameters == "Ec_surf", "$E_c$",
                                                          ifelse(parameters == "Ea_sprinkler",
                                                                 ifelse(parameters == "Ec_sprin
                                                                        ifelse(parameters == "
                                                                               ifelse(paramete


rbind(tmp[IFT == "Mixed"], tmp2) %>%
  ggplot(., aes(parameters, mean, fill = sensitivity), color = black) +
  geom_bar(stat = "identity", position = position_dodge(0.6), color = "black") +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), position = position_dodge(0.6)) +
  scale_x_discrete(labels = label_facets) +
  scale_fill_discrete(name = "Sensitivity", labels = c("$S_i$", "$T_i$")) +
  labs(x = "", y = "Sobol' indices") +
  facet_grid(~IFT, space = "free_x", scale = "free_x") +
  theme_AP()
```




```r
# EXTRACT SOBOL' INDICES FOR JAGER -------------------------------------------------
```

```
jager.tmp <- lapply(y[["Jägermeyr et al. 2015"]], function(x) x$indices$results)
names(jager.tmp) <- new.rohwer$Country

jager.ind <- rbindlist(jager.tmp, idcol = "Country") %>%
  .[, Continent:= countrycode(.[, Country],
                             origin = "country.name",
                             destination = "continent")] %>%
  .[, parameters:= ifelse(parameters == "Ea_surf", "E[a[su]]",
                         ifelse(parameters == "Ec_surf", "E[c[su]]",
                                ifelse(parameters == "Ea_sprinkler", "E[a[sp]]",
                                       ifelse(parameters == "Ec_sprinkler", "E[c[sp]]",
                                              ifelse(parameters == "Ea_micro", "E[a[mi]]",
                                                     ifelse(parameters == "Ec_micro", "E[c[mi]
                                                            ifelse(parameters == "Proportion_la
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

```
Continent_vector <- c("Africa", "Americas", "Asia", "Europe")

lapply(Continent_vector, function(x)
  ggplot(jager.ind[Continent == x], aes(parameters, original, fill = sensitivity), color = bla
    geom_bar(stat = "identity", position = position_dodge(0.6), color = "black") +
    scale_fill_discrete(name = "Sensitivity", labels = c("Si", "Ti")) +
    labs(x = "", y = "Sobol' indices") +
    scale_x_discrete(labels = ggplot2::parse_safe) +
    coord_flip() +
    scale_y_continuous(breaks = pretty_breaks(n = 3)) +
    facet_wrap(~Country) +
    theme_AP() +
    theme(strip.text.x = element_text(size = 6),
          axis.text.x = element_text(size = 6)) +
    ggtitle(x)
)
```

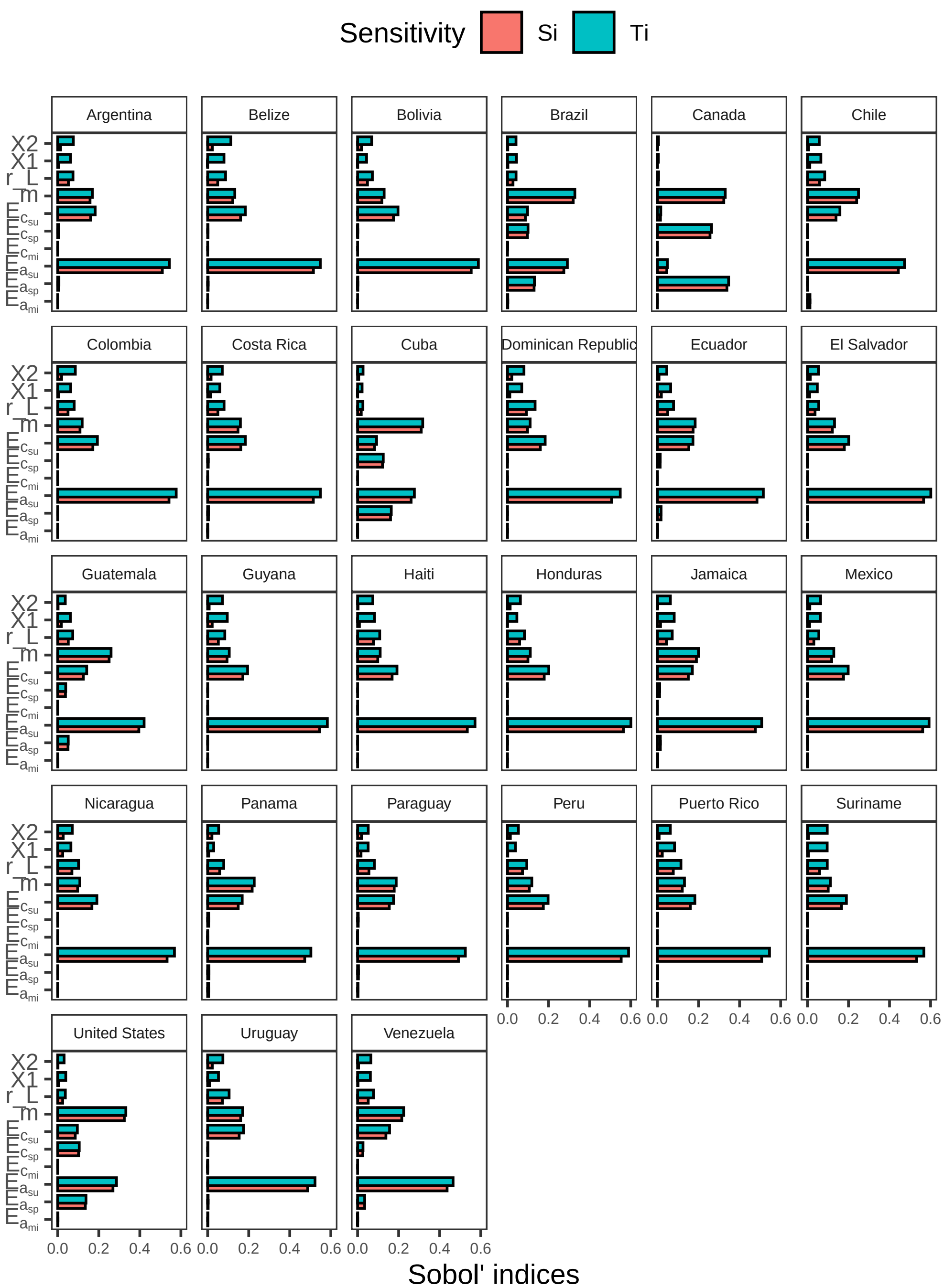
```
## [[1]]
```

Africa

Sensitivity
Si
Ti
Algeria
Angola
Benin
Botswana
Burkina Faso
Burundi
Cameroon
ntral African Repu
Chad
Djibouti
Egypt
Equatorial Guinea
Eritrea
Ethiopia
Gabon
Gambia
Ghana
Guinea
Guinea−Bissau
Ivory Coast
Kenya
Lesotho
Liberia
Libya
Madagascar
Malawi
Mali
Mauritania
Morocco
Mozambique
Namibia
Niger
Nigeria
Rwanda
Senegal
Sierra Leone
Somalia
South Africa
Sudan
Swaziland
Tanzania
Togo
Tunisia
Uganda
Zambia
Zimbabwe
0.0 0.2 0.4 0.6
Sobol' indices

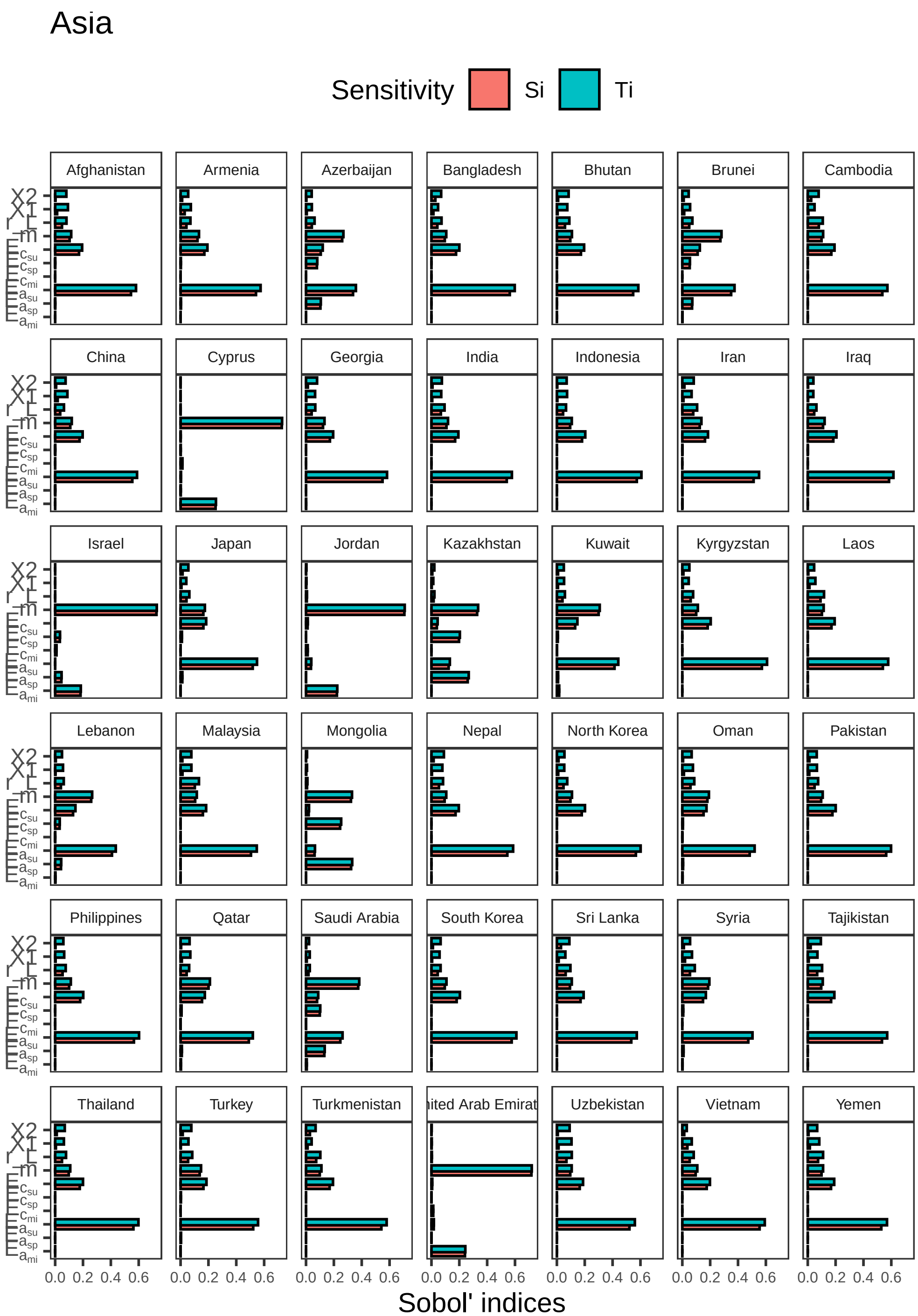

```
##
## [[2]]
```

Americas



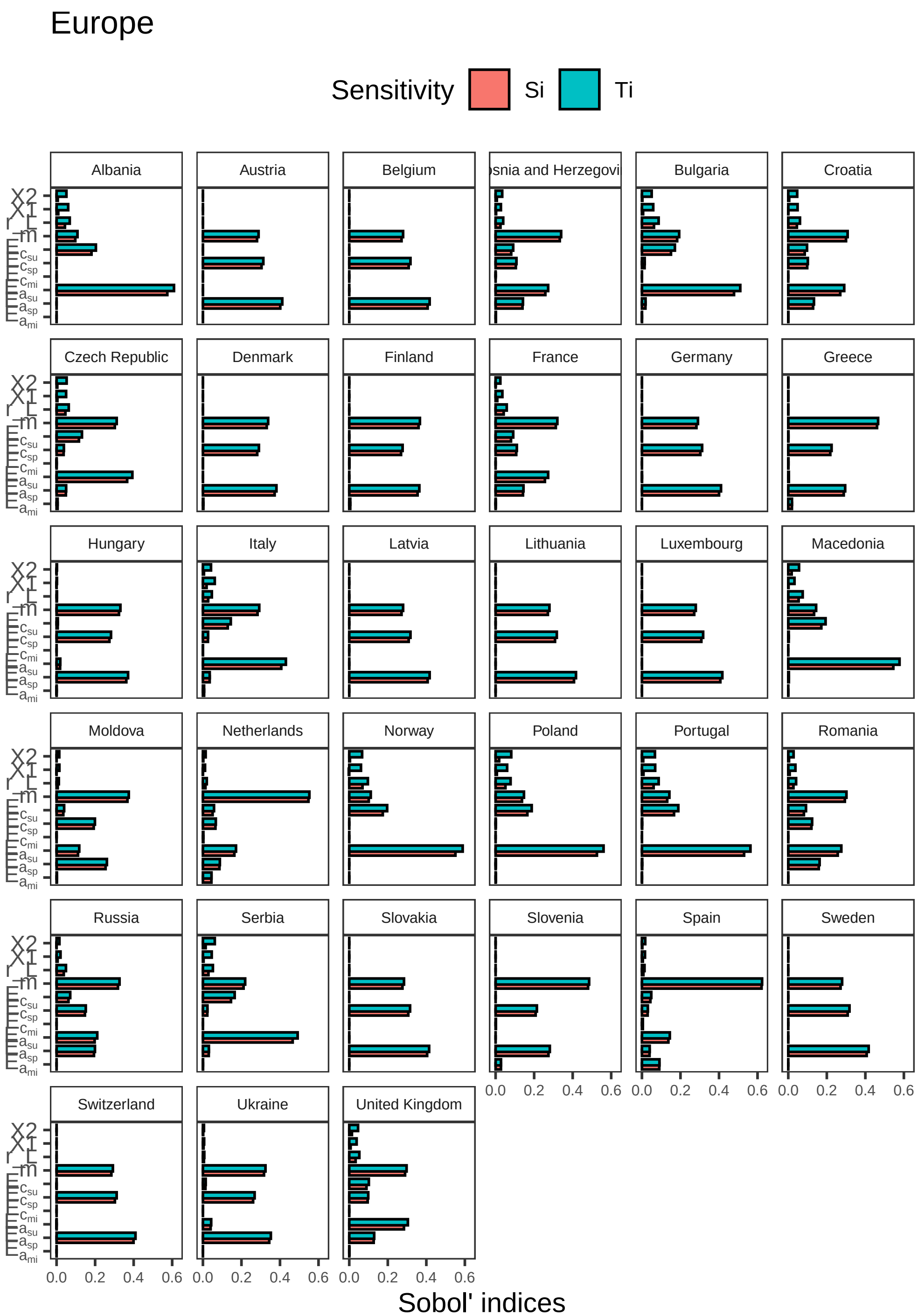
Sensitivity
Si
Ti
Argentina
Belize
Bolivia
Brazil
Canada
Chile
Colombia
Costa Rica
Cuba
Dominican Republic
Ecuador
El Salvador
Guatemala
Guyana
Haiti
Honduras
Jamaica
Mexico
Nicaragua
Panama
Paraguay
Peru
Puerto Rico
Suriname
United States
Uruguay
Venezuela
X2
X1
r_L
m
0.0
0.2
0.4
0.6
Sobol' indices


```
##
## [[3]]
```



Asia
Sensitivity
Si
Ti
Afghanistan
Armenia
Azerbaijan
Bangladesh
Bhutan
Brunei
Cambodia
China
Cyprus
Georgia
India
Indonesia
Iran
Iraq
Israel
Japan
Jordan
Kazakhstan
Kuwait
Kyrgyzstan
Laos
Lebanon
Malaysia
Mongolia
Nepal
North Korea
Oman
Pakistan
Philippines
Qatar
Saudi Arabia
South Korea
Sri Lanka
Syria
Tajikistan
Thailand
Turkey
Turkmenistan
ited Arab Emirat
Uzbekistan
Vietnam
Yemen
0.0 0.2 0.4 0.6
Sobol' indices


```
##
## [[4]]
```


Europe
Sensitivity
Si
Ti
Albania
Austria
Belgium
snia and Herzegovi
Bulgaria
Croatia
Czech Republic
Denmark
Finland
France
Germany
Greece
Hungary
Italy
Latvia
Lithuania
Luxembourg
Macedonia
Moldova
Netherlands
Norway
Poland
Portugal
Romania
Russia
Serbia
Slovakia
Slovenia
Spain
Sweden
Switzerland
Ukraine
United Kingdom
0.0 0.2 0.4 0.6
Sobol' indices