A critique of irrigation efficiency modeling

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"))

# 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

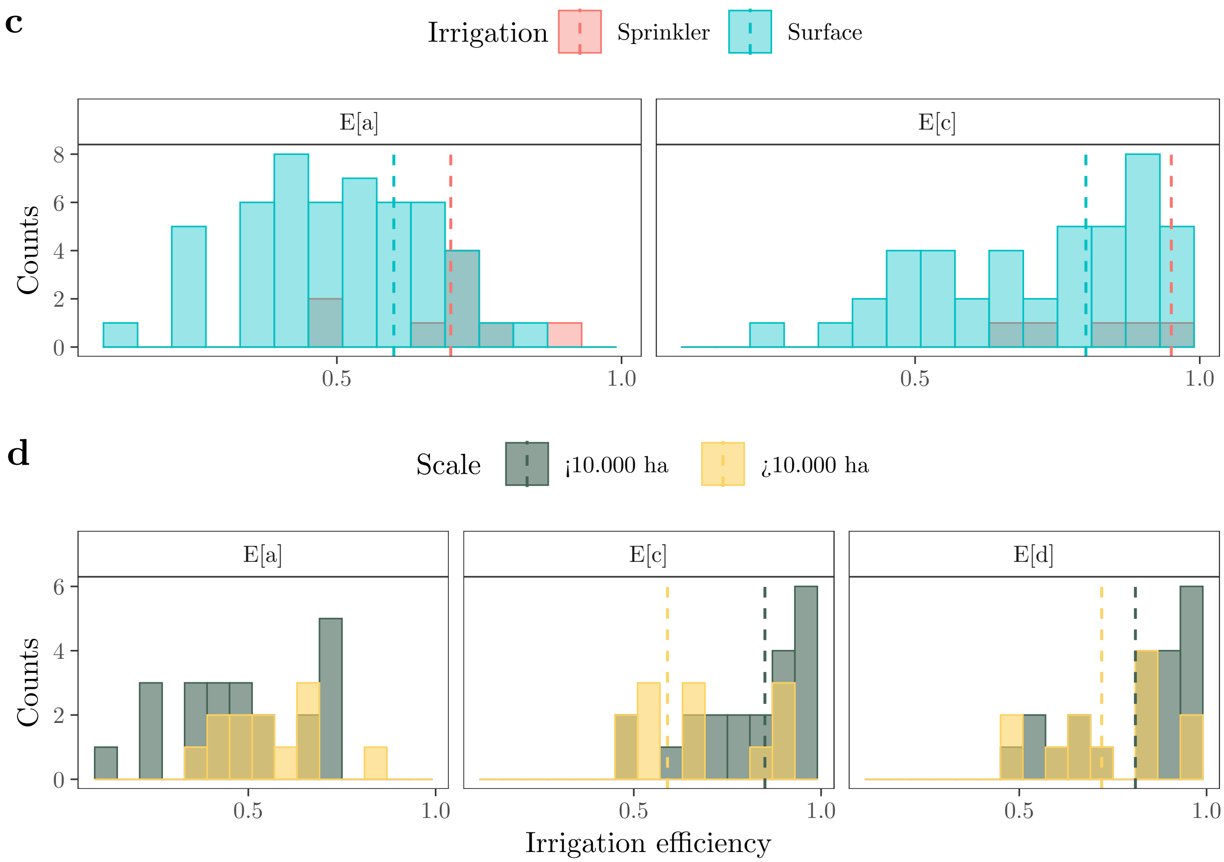
```r
# READ IN DATA ---------------------------------------------------------------------

# 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 -----------------------------------------------------------------------------

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

efficiencies_labeller <- c("E_c" = "E[c]",
                           "E_a" = "E[a]")

a <- bos %>%
  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 = as_labeller(efficiencies_labeller)) +
  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 ----------------------

efficiencies_labeller <- c("E_c" = "E[c]",
                           "E_a" = "E[a]",
                           "E_d" = "E[d]")

b <- melt(bos, 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 = as_labeller(efficiencies_labeller)) +
  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
```

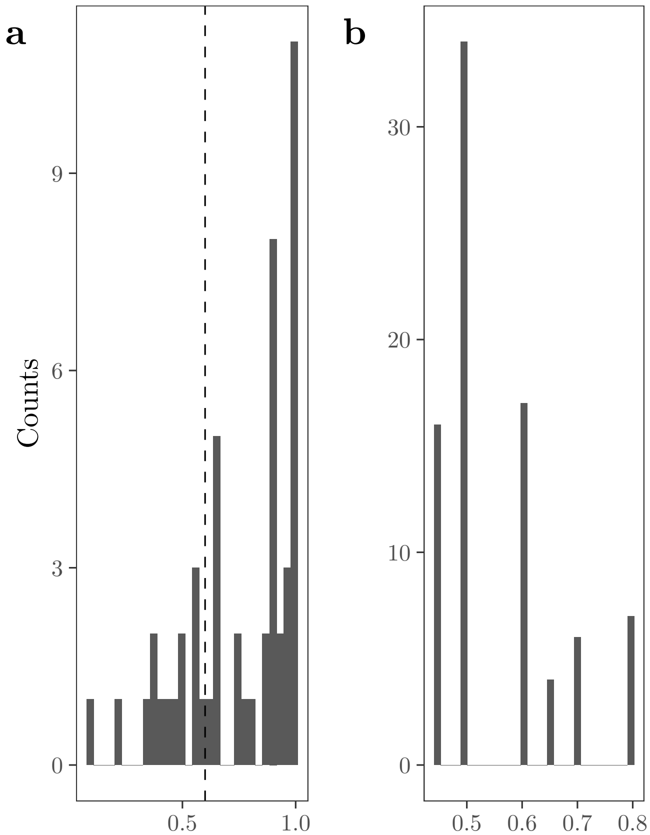



```
# PLOT USA AND AFRICA ---------------------------------------------------------------

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")
```

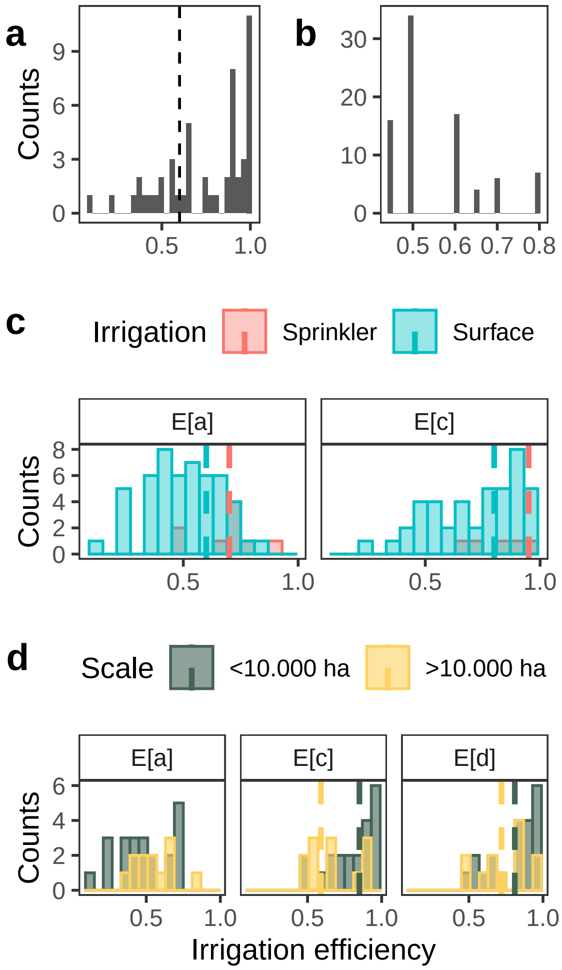
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## 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", "Proportion_large", "m"),
  "Sprinkler" = c("Ea_sprinkler", "Ec_sprinkler", "Proportion_large", "m"),
  "Micro" = c("Ea_micro", "Ec_micro", "Proportion_large", "m"),
  "Mixed" = c("Ea_surf", "Ea_sprinkler", "Ec_surf", "Ec_sprinkler", "Proportion_large", "m"),
  "Jager" = c("Ea_surf", "Ea_sprinkler", "Ec_surf", "Ec_sprinkler",
              "Ea_micro", "Ec_micro", "Proportion_large", "m")
)

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,
```

```
  # 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)
)
```

2.3 Uncertainty in the proportion of large-scale irrigated areas

```
# 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
```

2.4 Function to create sample matrix and transfrom to appropriate distributions

```
# 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]]

  if(IFT == "Surface" | IFT == "Mixed") {

  mat[, "Proportion_large"] <- qunif(mat[, "Proportion_large"],
                                     countries.frac$min, countries.frac$max)
  }
  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") {

    Mf <- mat[, "m"] - 0.5 * mat[, "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"] - 0.5 * mat[, "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 + 0.5 * 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"] - 0.5 * mat[, "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"]
```

```r
    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

  }

  if(IFT == "Jager") {
    boot <- FALSE
    R <- NULL
  } else {
    boot <- TRUE
    R <- R
  }

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

2.6 Define settings

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

N <- 2^13
R <- 10^2
```

2.7 Run model

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

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

```
# 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",
                                                 destination = "continent")]
}
```

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

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

```
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")]
```

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

```r
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 0.4650154
##   2:      Asia     United Arab Emirates 0.4889552
##   3:      Asia                   Israel 0.4901008
##   4:      Asia                   Jordan 0.5107562
##   5:    Africa                    Benin 0.5460129
##   6:      Asia                   Kuwait 0.5519086
##   7:    Africa                  Tunisia 0.5546542
##   8:    Africa               Mozambique 0.5549232
##   9:      Asia             Saudi Arabia 0.5556065
##  10:    Africa                  Namibia 0.5567580
##  11:    Europe                    Italy 0.5576075
##  12:  Americas            United States 0.5586242
##  13:  Americas                   Brazil 0.5586775
##  14:      Asia               Kazakhstan 0.5595325
##  15:    Africa              Ivory Coast 0.5607401
##  16:    Africa                 Zimbabwe 0.5626920
##  17:  Americas                     Cuba 0.5643215
##  18:    Europe                   France 0.5645315
##  19:    Africa                Swaziland 0.5681810
##  20:    Africa                   Zambia 0.5700714
##  21:  Americas                   Canada 0.5722302
##  22:    Africa             South Africa 0.5749587
##  23:    Africa                 Botswana 0.5802624
##  24:      Asia                   Brunei 0.5816775
##  25:    Europe                    Spain 0.5824526
##  26:      Asia                  Lebanon 0.5850318
##  27:    Europe                   Greece 0.5923817
##  28:    Africa                   Malawi 0.5923817
##  29:    Europe                  Austria 0.5941574
##  30:    Europe              Netherlands 0.5954245
##  31:    Europe              Switzerland 0.5960538
##  32:    Europe                  Denmark 0.5963984
##  33:    Europe                 Slovakia 0.5977046
##  34:    Europe                  Finland 0.5987308
##  35:    Europe                  Germany 0.5987520
##  36:    Europe                  Belgium 0.6002391
##  37:      <NA>                 Byelarus 0.6002391
```

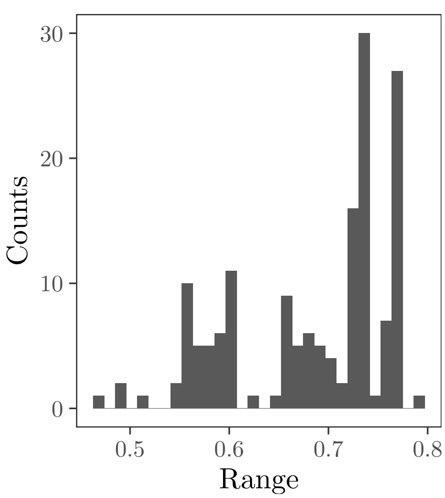
```
## 38:   Europe                 Latvia 0.6002391
## 39:   Europe              Lithuania 0.6002391
## 40:   Europe             Luxembourg 0.6002391
## 41:   Europe                 Sweden 0.6002391
## 42:   Europe                Hungary 0.6051666
## 43:     Asia                  Qatar 0.6074024
## 44:   Europe                Ukraine 0.6224391
## 45:   Europe               Slovenia 0.6463006
## 46:   Europe                 Russia 0.6523247
## 47:   Europe         Czech Republic 0.6526027
## 48:   Europe               Bulgaria 0.6560802
## 49: Americas                Ecuador 0.6560802
## 50:     Asia                  Japan 0.6567438
## 51:  Oceania              Australia 0.6572847
## 52:   Africa           Burkina Faso 0.6599589
## 53:   Europe                Croatia 0.6610773
## 54:   Europe                Romania 0.6627251
## 55:   Europe         United Kingdom 0.6639204
## 56:   Africa                  Libya 0.6666597
## 57: Americas              Argentina 0.6675349
## 58:     Asia                 Turkey 0.6729158
## 59: Americas              Venezuela 0.6730108
## 60:   Africa                Algeria 0.6789910
## 61:     Asia                  Syria 0.6790189
## 62:   Europe               Portugal 0.6794900
## 63:     Asia                   Iran 0.6814100
## 64:   Africa                  Egypt 0.6817869
## 65:     Asia             Azerbaijan 0.6830715
## 66:     Asia                  China 0.6897170
## 67:     Asia                Georgia 0.6921287
## 68: Americas               Colombia 0.6945741
## 69: Americas                  Chile 0.6955291
## 70: Americas                 Mexico 0.6956149
## 71:   Europe                Moldova 0.6990668
## 72:     Asia            Afghanistan 0.7010588
## 73:     Asia                   Iraq 0.7012100
## 74:     Asia                  India 0.7029295
## 75:   Africa               Ethiopia 0.7084147
## 76:     Asia            Philippines 0.7161822
## 77:     Asia             Uzbekistan 0.7199820
## 78:   Europe                Albania 0.7204869
## 79:     Asia             Bangladesh 0.7204869
## 80:     Asia              Indonesia 0.7204869
## 81:     Asia             Kyrgyzstan 0.7204869
## 82:     Asia                  Nepal 0.7204869
## 83:   Africa                Nigeria 0.7204869
## 84:     Asia            North Korea 0.7204869
## 85:     Asia               Pakistan 0.7204869
```

```
##  86:      Asia                   Sri Lanka 0.7204869
##  87:    Africa                       Sudan 0.7204869
##  88:      Asia                  Tajikistan 0.7204869
##  89:    Africa                    Tanzania 0.7204869
##  90:      Asia                    Thailand 0.7204869
##  91:      Asia                Turkmenistan 0.7204869
##  92:      Asia                     Vietnam 0.7204869
##  93:    Europe  Bosnia and Herzegovina 0.7335640
##  94:      Asia                       Burma 0.7335640
##  95:    Africa                    Cameroon 0.7335640
##  96:    Africa                        Chad 0.7335640
##  97:    Africa                       Congo 0.7335640
##  98:  Americas                  Costa Rica 0.7335640
##  99:  Americas               French Guiana 0.7335640
## 100:    Africa                       Gabon 0.7335640
## 101:    Africa                       Ghana 0.7335640
## 102:  Americas                   Guatemala 0.7335640
## 103:  Americas                     Jamaica 0.7335640
## 104:    Africa                       Kenya 0.7335640
## 105:    Africa                     Lesotho 0.7335640
## 106:      Asia                    Mongolia 0.7335640
## 107:    Africa                     Morocco 0.7335640
## 108:   Oceania                 New Zealand 0.7335640
## 109:  Americas                      Panama 0.7335640
## 110:   Oceania            Papua New Guinea 0.7335640
## 111:  Americas                    Paraguay 0.7335640
## 112:    Europe                      Poland 0.7335640
## 113:    Europe                      Serbia 0.7335640
## 114:    Africa                        Togo 0.7335640
## 115:  Americas                    Trinidad 0.7335640
## 116:  Americas                     Uruguay 0.7335640
## 117:    Africa              Western Sahara 0.7335640
## 118:    Africa                       Zaire 0.7335640
## 119:  Americas                      Belize 0.7339335
## 120:  Americas                 El Salvador 0.7369390
## 121:      Asia                     Armenia 0.7374656
## 122:  Americas                 Puerto Rico 0.7413326
## 123:  Americas                     Bolivia 0.7433014
## 124:    Africa                      Uganda 0.7555268
## 125:    Africa                  Mauritania 0.7593394
## 126:    Africa                      Guinea 0.7594368
## 127:    Europe                   Macedonia 0.7596352
## 128:      Asia                    Malaysia 0.7605567
## 129:  Americas                        Peru 0.7611599
## 130:  Americas                    Suriname 0.7618812
## 131:    Africa                     Senegal 0.7670757
## 132:  Americas                   Nicaragua 0.7680709
## 133:      Asia                       Yemen 0.7681130
```

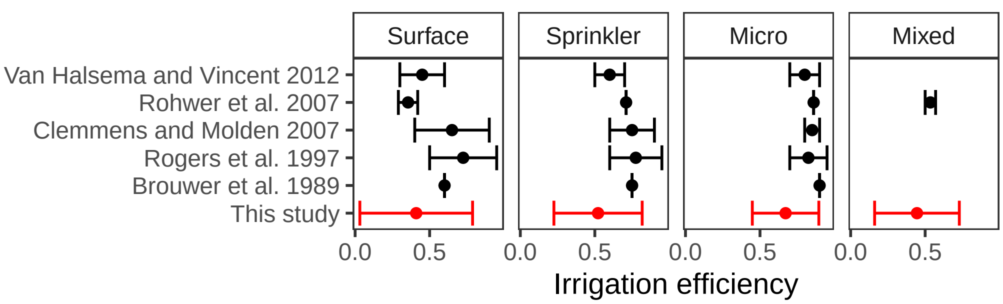
```
## 134:    Africa               Madagascar 0.7682063
## 135:    Africa                     Mali 0.7685685
## 136:    Africa                   Angola 0.7690661
## 137:      Asia                   Bhutan 0.7690661
## 138:    Africa                  Burundi 0.7690661
## 139:      Asia                 Cambodia 0.7690661
## 140:    Africa Central African Republic 0.7690661
## 141:    Africa                 Djibouti 0.7690661
## 142:  Americas       Dominican Republic 0.7690661
## 143:    Africa        Equatorial Guinea 0.7690661
## 144:    Africa                  Eritrea 0.7690661
## 145:    Africa                   Gambia 0.7690661
## 146:    Africa            Guinea-Bissau 0.7690661
## 147:  Americas                   Guyana 0.7690661
## 148:  Americas                    Haiti 0.7690661
## 149:  Americas                 Honduras 0.7690661
## 150:      Asia                     Laos 0.7690661
## 151:    Africa                  Liberia 0.7690661
## 152:    Africa                    Niger 0.7690661
## 153:    Europe                   Norway 0.7690661
## 154:    Africa                   Rwanda 0.7690661
## 155:    Africa             Sierra Leone 0.7690661
## 156:    Africa                  Somalia 0.7690661
## 157:      Asia              South Korea 0.7690661
## 158:      Asia                     Oman 0.7883613
##      Continent                  Country     range
```

```
ggplot(calc, aes(range)) +
  geom_histogram() +
  labs(x = "Range", y = "Counts") +
  theme_AP()
```

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



```r
# 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 == "One IFT per country"] %>%
  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
```

```
## named list()
```

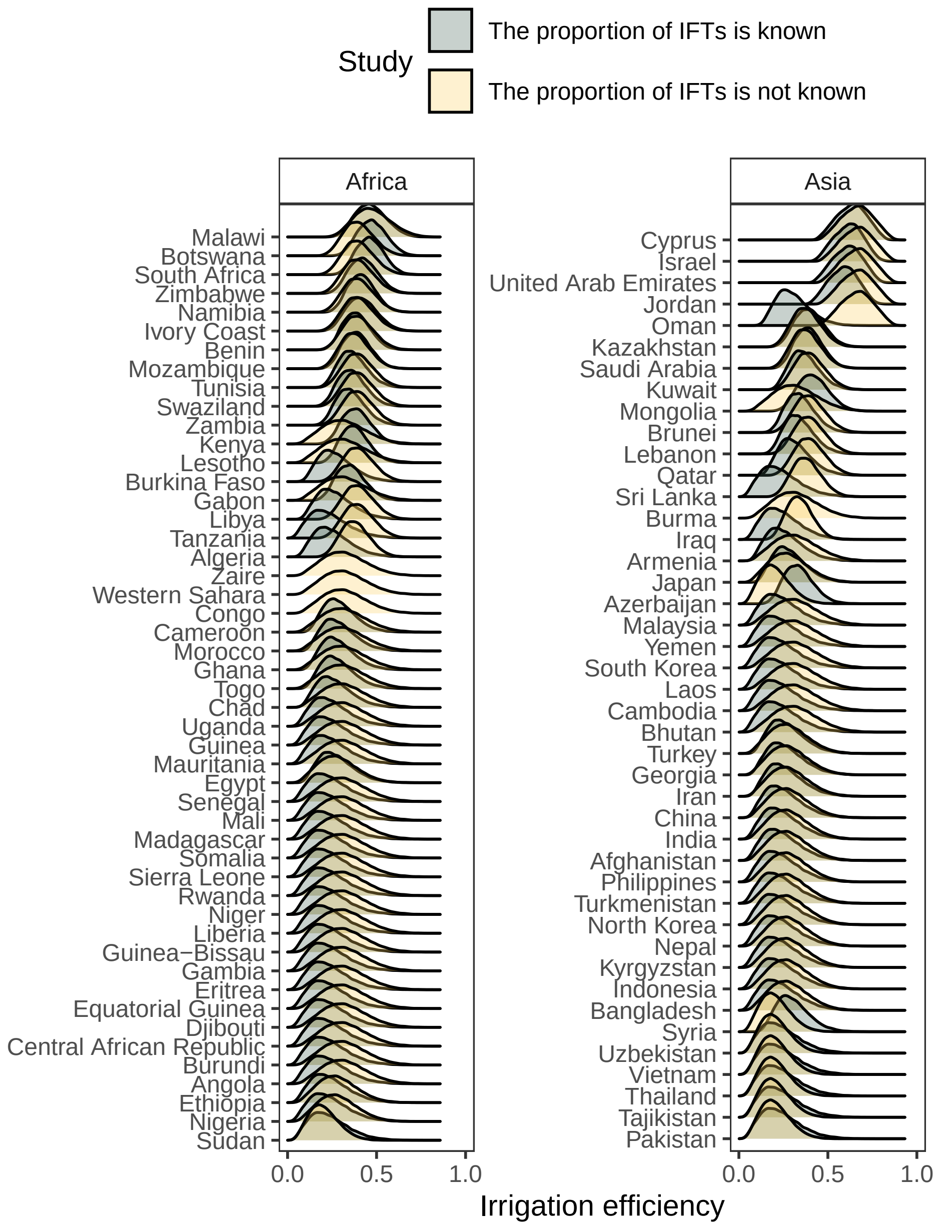
```r
ff <- uncertainty.dt[!Continent == "Oceania"] %>%
  split(., .$Approach, drop = TRUE)
```

3 Uncertainty analysis

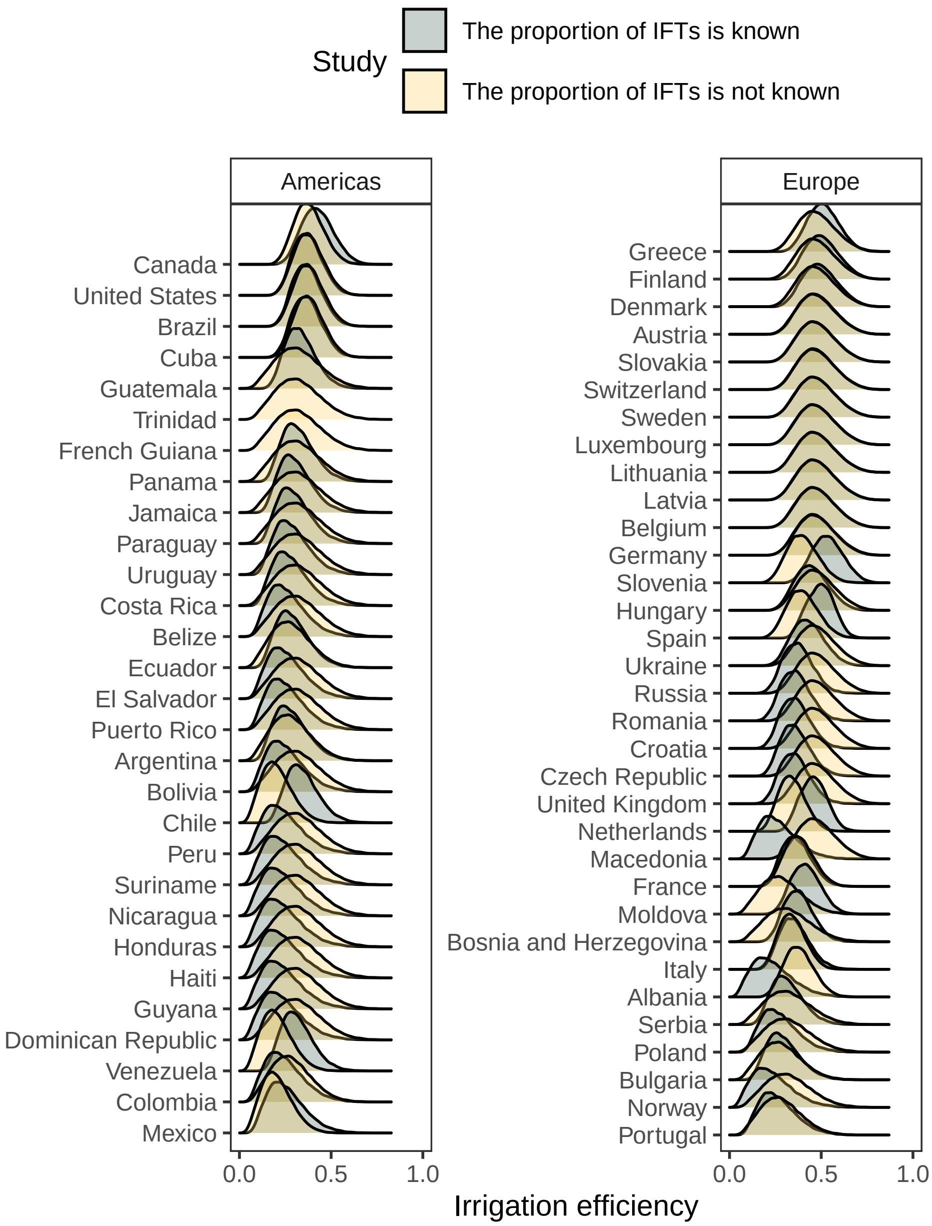
```r
# PLOT UNCERTAINTY -------------------------------------------------------------

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

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),
                       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 0.0132

## Picking joint bandwidth of 0.0126
```




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

## Picking joint bandwidth of 0.0132

## Picking joint bandwidth of 0.0121
```



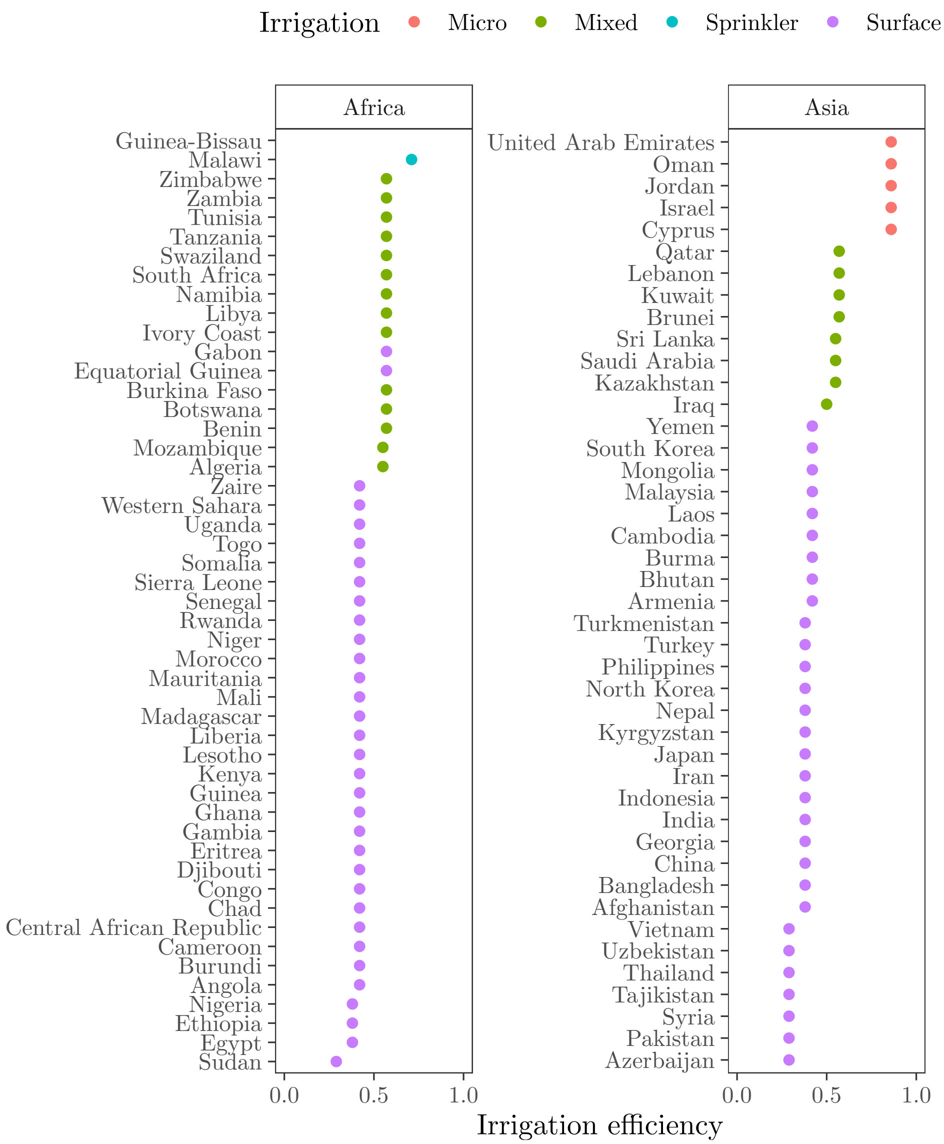

```
# PLOT ROHWER ET AL.'S IRRIGATION EFFICIENCY VALUES --------------------------------

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

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

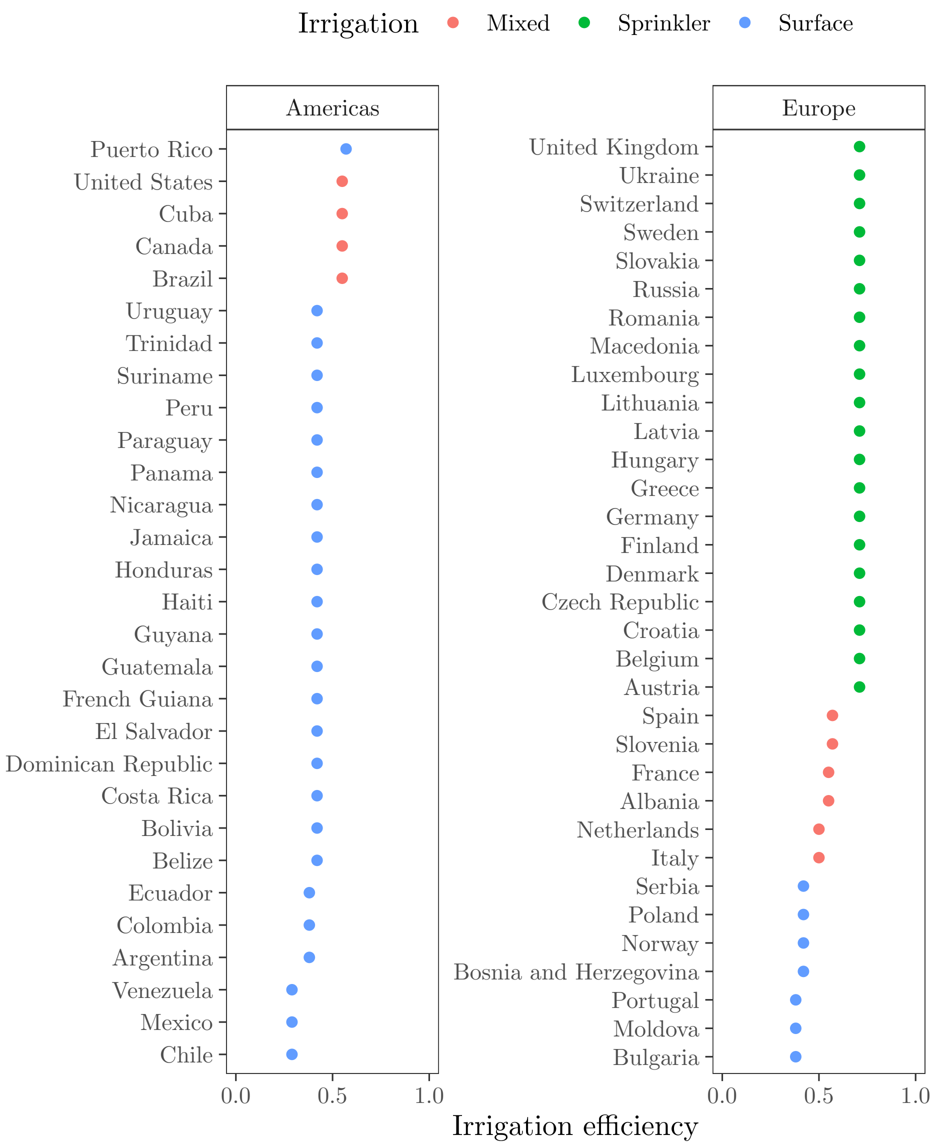
```
dd <- list()
for (i in 1:length(list_continents)) {
  dd[[i]] <- ggplot(rohwer[Continent %in% list_continents[[i]]],
                    aes(x = Ep, y = fct_reorder(Country, Ep), color = IFT)) +
    geom_point() +
    labs(x = "Irrigation efficiency", y = "") +
    scale_x_continuous(breaks = pretty_breaks(n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

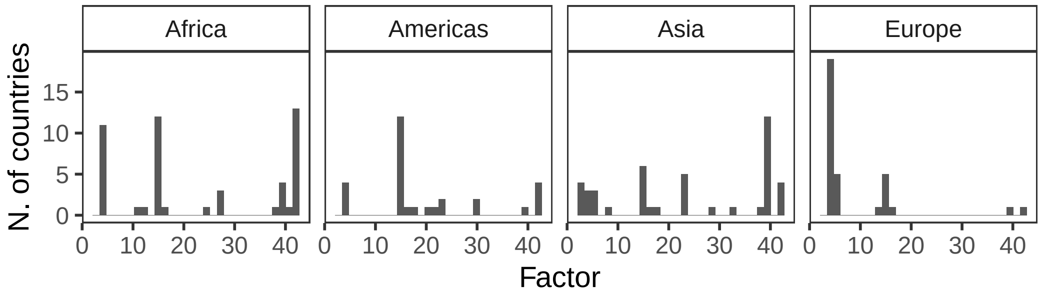



```
##
## [[2]]
```




```r
# 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`.
```



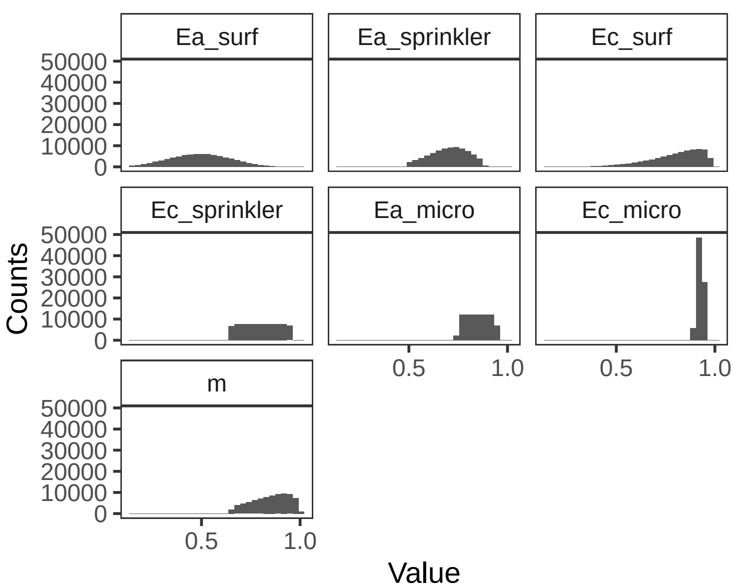

```
# Number of countries whose irrigation water withdrawals fluctuate a factor of x
# due to uncertainty in irrigation efficiency
factor_unc %>%
  .[, factor:= floor(max / min)] %>%
  .[, .(number.countries = .N), factor] %>%
  .[order(factor)] %>%
  print()
```

```
##     factor number.countries
##  1:      2                4
##  2:      3               16
##  3:      4               24
##  4:      5                5
##  5:      8                1
##  6:     10                1
##  7:     11                1
##  8:     14               26
##  9:     15               12
## 10:     16                2
## 11:     17                1
## 12:     18                1
## 13:     20                1
## 14:     22                4
## 15:     23                4
## 16:     24                1
## 17:     27                3
## 18:     28                1
## 19:     29                1
## 20:     30                1
## 21:     32                1
## 22:     37                2
## 23:     38               15
## 24:     39                3
## 25:     40                1
## 26:     41               22
##     factor number.countries
```

4 Sensitivity analysis

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

# Define labels
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]

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`.
```




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

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

```r
ind <- rbindlist(ind, idcol = "Country")

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 == "I
                                                                         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()
```