A critique of irrigation efficiency modeling

R code

Arnald Puy

Contents

1 Read in data **3**

2 The model **5**

2.1 Function to create sample matrix 5

2.2 Define distributions 6

2.3 Uncertainty in the proportion of large-scale irrigated areas 8

2.4 Function to create sample matrix and transfrom to appropriate distributions 8

2.5 Run the model 8

2.6 Define settings 9

2.7 Run model 9

2.8 Extract model output 10

2.9 Uncertainty analysis 10

```
# 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"))

# 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"))
}

# 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]

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

# 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)
```

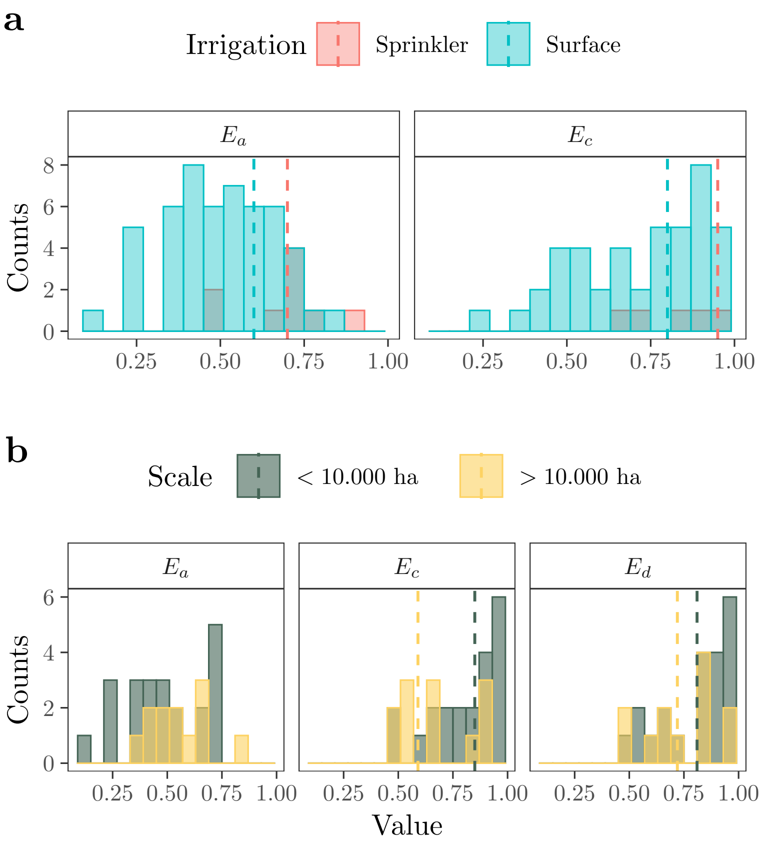
```
# PLOT ----------------------------------------------------------------------------

# 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)) +
```

```r
  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 = "Value", y = "Counts") +
  scale_fill_manual(values = wes_palette(2, name = "Chevalier1"),
                    name = "Scale") +
  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_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()
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

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

plot_grid(a, b, ncol = 1, labels = "auto")
```

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



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")
)

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

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

```
# 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) -----

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))
```

```
# 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, 0.9),
  "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)
  }
)
```

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]]
  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) {

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

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

2.7 Run model

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

y <- mclapply(1:nrow(rohwer), function(x)
  full_model(IFT = rohwer[[x, "IFT"]],
             Country = rohwer[[x, "Country"]],
             sample.size = N,
             R = R),
  mc.cores = detectCores() * 0.75)
```

2.8 Extract model output

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

output <- lapply(y, function(x) x[[1]])
names(output) <- rohwer$Country
tmp <- lapply(output, data.table) %>%
  rbindlist(., idcol = "Country") %>%
  merge(., rohwer[, .(Country, IFT)], all.x = TRUE)

tmp <- tmp[, Continent:= countrycode(tmp[, Country], origin = "country.name",
                                     destination = "continent")] %>%
  .[, IFT:= factor(IFT, levels = c("Surface", "Sprinkler", "Micro", "Mixed"))]
```

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

2.9 Uncertainty analysis

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

plot_ggridges <- function(dt, Cont) {
  pp <- ggplot(dt[Continent == Cont], aes(x = V1, y = fct_reorder(Country, V1),
                                           fill = IFT), alpha = 0.5) +
    geom_density_ridges(scale = 2) +
    labs(x = "Irrigation efficiency", y = "") +
    facet_wrap(~Continent) +
    theme_AP() +
    theme(legend.position = "none")
  return(pp)
}

a <- plot_ggridges(dt = tmp, Cont = "Africa") +
  scale_fill_manual(labels = c("Surface", "Sprinkler", "Mixed"),
                    values = c("#D8B70A", "#02401B", "#81A88D"),
                    name = "Irrigation")

b <- plot_ggridges(dt = tmp, Cont = "Americas") +
  scale_fill_manual(labels = c("Surface", "Mixed"),
                    values = c("#D8B70A", "#81A88D"),
                    name = "Irrigation")

c <- plot_ggridges(dt = tmp, Cont = "Asia") +
  scale_fill_manual(labels = c("Surface", "Micro", "Mixed"),
                    values = c("#D8B70A", "#A2A475", "#81A88D"),
                    name = "Irrigation")

d <- plot_ggridges(dt = tmp, Cont = "Europe") +
  scale_fill_manual(labels = c("Surface", "Sprinkler", "Mixed"),
                    values = c("#D8B70A", "#02401B", "#81A88D"),
                    name = "Irrigation")


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

```
## Picking joint bandwidth of 0.0109
```

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

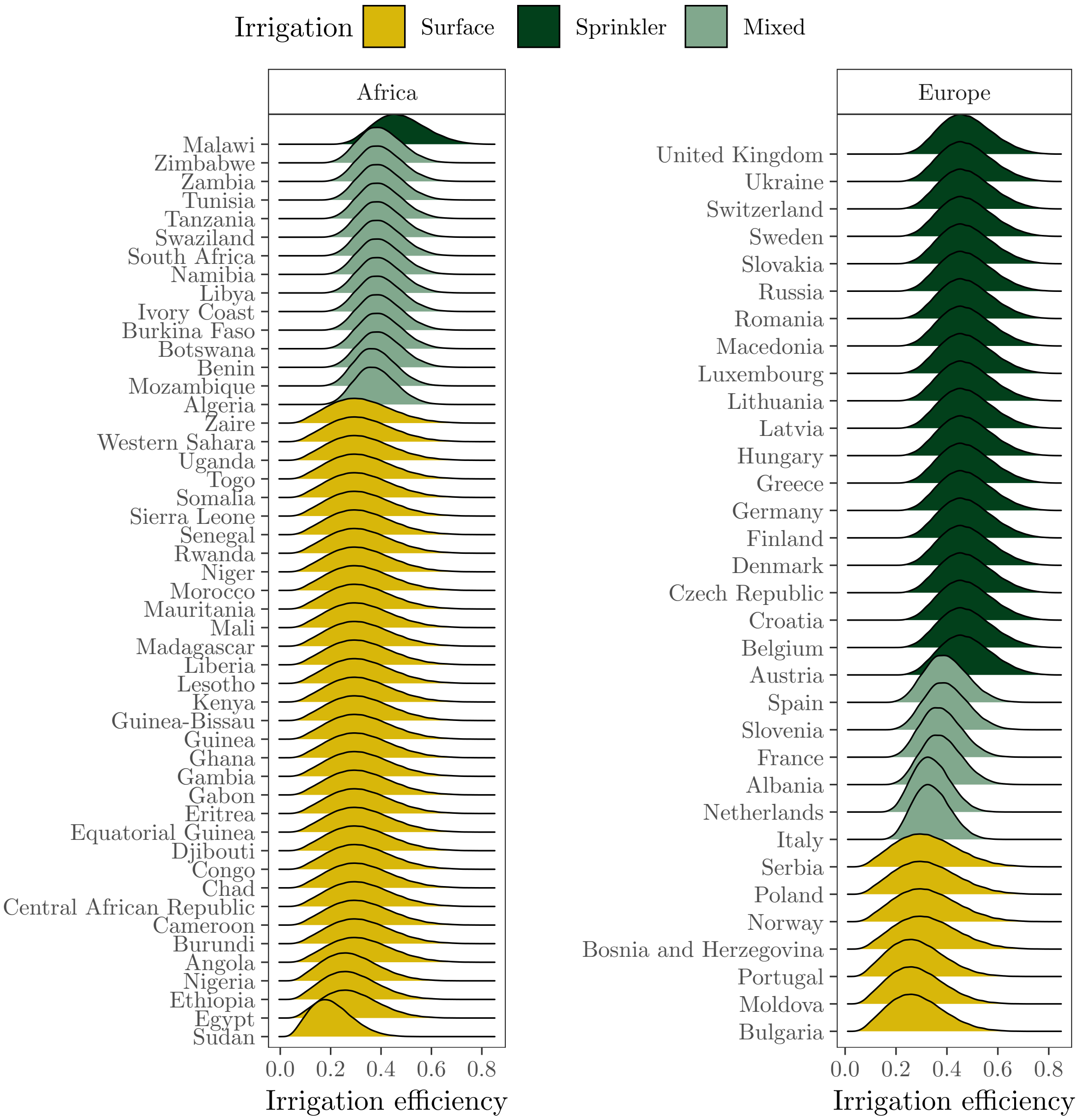
```
## Picking joint bandwidth of 0.00995
```

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

bottom <- plot_grid(a, d, ncol = 2)
```

```
## Picking joint bandwidth of 0.0109
```

```
## Picking joint bandwidth of 0.0101
```

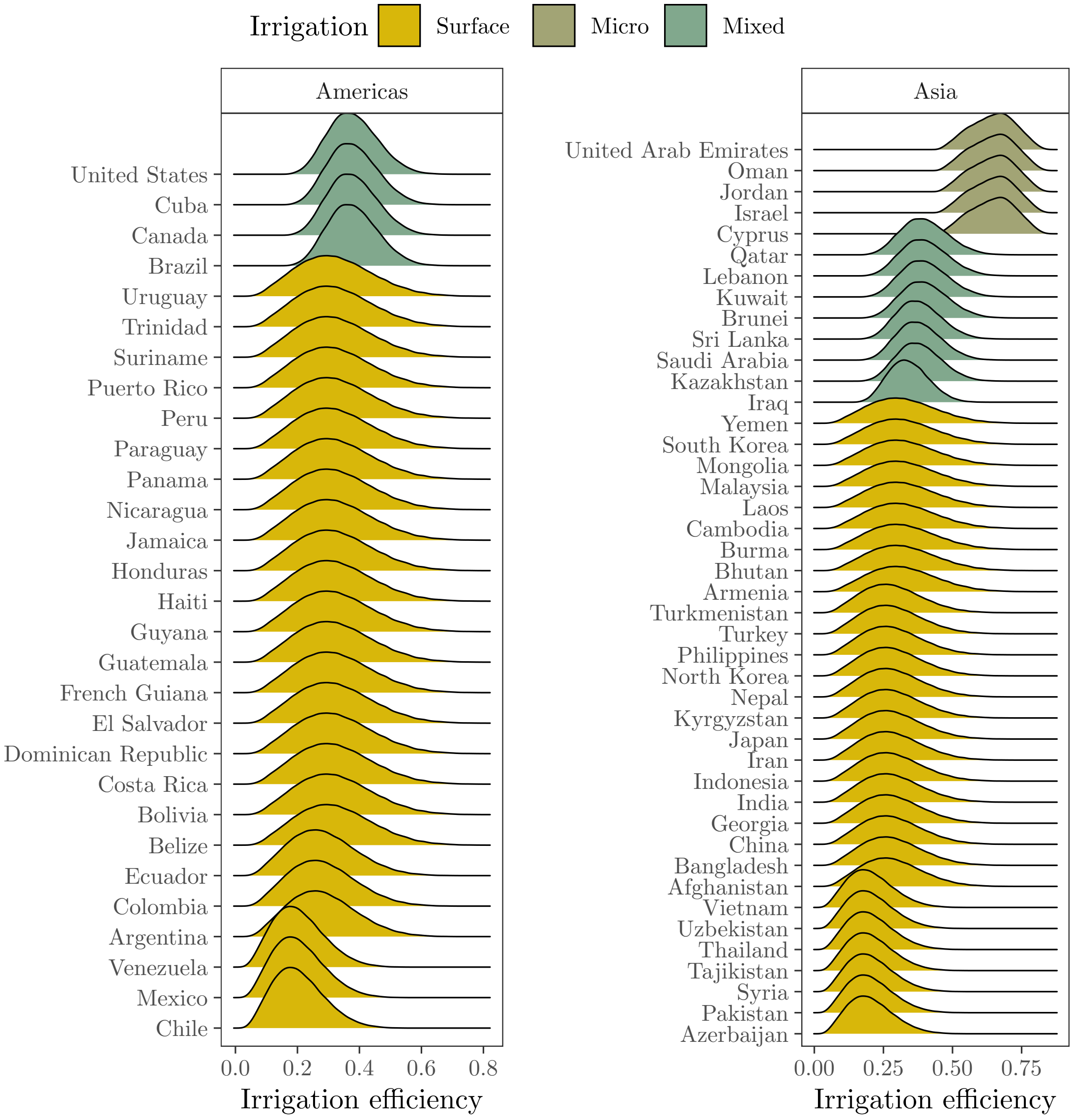
```
plot_grid(legend.africa, bottom, ncol = 1, rel_heights = c(0.05, 0.95))
```



```
bottom <- plot_grid(b, c, ncol = 2)
```

```
## Picking joint bandwidth of 0.0112

## Picking joint bandwidth of 0.00995
```

```
plot_grid(legend.asia, bottom, ncol = 1, rel_heights = c(0.05, 0.95))
```


Irrigation
Surface
Micro
Mixed
Americas
United States
Cuba
Canada
Brazil
Uruguay
Trinidad
Suriname
Puerto Rico
Peru
Paraguay
Panama
Nicaragua
Jamaica
Honduras
Haiti
Guyana
Guatemala
French Guiana
El Salvador
Dominican Republic
Costa Rica
Bolivia
Belize
Ecuador
Colombia
Argentina
Venezuela
Mexico
Chile
0.0
0.2
0.4
0.6
0.8
Irrigation efficiency
Asia
United Arab Emirates
Oman
Jordan
Israel
Cyprus
Qatar
Lebanon
Kuwait
Brunei
Sri Lanka
Saudi Arabia
Kazakhstan
Iraq
Yemen
South Korea
Mongolia
Malaysia
Laos
Cambodia
Burma
Bhutan
Armenia
Turkmenistan
Turkey
Philippines
North Korea
Nepal
Kyrgyzstan
Japan
Iran
Indonesia
India
Georgia
China
Bangladesh
Afghanistan
Vietnam
Uzbekistan
Thailand
Tajikistan
Syria
Pakistan
Azerbaijan
0.00
0.25
0.50
0.75
Irrigation efficiency