Uncertainty persists after 50 years of global irrigation modelling

R code

Arnald Puy

Contents

1 Preliminary functions

```
#   PRELIMINARY FUNCTIONS #####################################################

sensobol::load_packages(c("openxlsx", "data.table", "tidyverse","cowplot",
                          "benchmarkme", "parallel", "wesanderson", "scales", "ncdf4",
                          "countrycode", "rworldmap", "sp", "doParallel", "here", "lme4",
                          "microbenchmark", "mgcv", "brms", "randomForest", "here"))

# 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),
          strip.background = element_rect(fill = "white"),
          legend.text = element_text(size = 7.3),
          axis.title = element_text(size = 10),
          legend.key.width = unit(0.4, "cm"),
          legend.key.height = unit(0.4, "cm"),
          legend.key.spacing.y = unit(0, "lines"),
          legend.box.spacing = unit(0, "pt"),
          legend.title = element_text(size = 7.8),
          axis.text.x = element_text(size = 7),
          axis.text.y = element_text(size = 7),
          axis.title.x = element_text(size = 7.3),
          axis.title.y = element_text(size = 7.3),
          plot.title = element_text(size = 8),
          strip.text.x = element_text(size = 7.4),
          strip.text.y = element_text(size = 7.4))
}

# Select color palette --------------------------------------------------------

selected.palette <- "Darjeeling1"
```

```
# SOURCE ALL R FUNCTIONS NEEDED FOR THE STUDY ##################################

# Source all .R files in the "functions" folder --------------------------------

r_functions <- list.files(path = here("functions"), pattern = "\\.R$", full.names = TRUE)
lapply(r_functions, source)
```

2 ISIMIP Data

2.1 Historical data

```r
# RETRIEVE DATA FROM ISIMIP #####################################################

# Create vector with list of files ----------------------------------------------

list.of.files <- list.files("./files/isimip")
model.names <- sub("^(.*?)_.*", "\\1", list.of.files)
climate.scenarios <- sapply(strsplit(list.of.files, "_"), function(x) x[2])
social.scenarios <- sapply(strsplit(list.of.files, "_"), function(x) x[which(x == "co2") - 1])
files.directory <- paste("./files/isimip", list.of.files, sep = "/")
start_year <- 1971

# Create parallel cluster -------------------------------------------------------

numCores <- detectCores() * 0.75
cl <- makeCluster(numCores)
registerDoParallel(cl)

# Run for loop ------------------------------------------------------------------

isimip.hist <- foreach(i = 1:length(files.directory),
                       .packages = c("data.table", "countrycode", "tidyverse",
                                     "sp", "rworldmap", "ncdf4")) %dopar% {

                                       get_isimip_fun(nc_file = files.directory[i],
                                                      variable = "airrww",
                                                      start_year = start_year)
                                     }

# Stop the cluster after the computation ----------------------------------------

stopCluster(cl)
```

```r
# ARRANGE DATA ##################################################################

# Number of files ---------------------------------------------------------------

list.of.files
```

```
## [1] "dbh_gswp3_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [2] "dbh_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [3] "dbh_watch_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2001.nc4"
## [4] "dbh_watch-wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [5] "dbh_wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc"
## [6] "h08_gswp3_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [7] "h08_gswp3_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
```

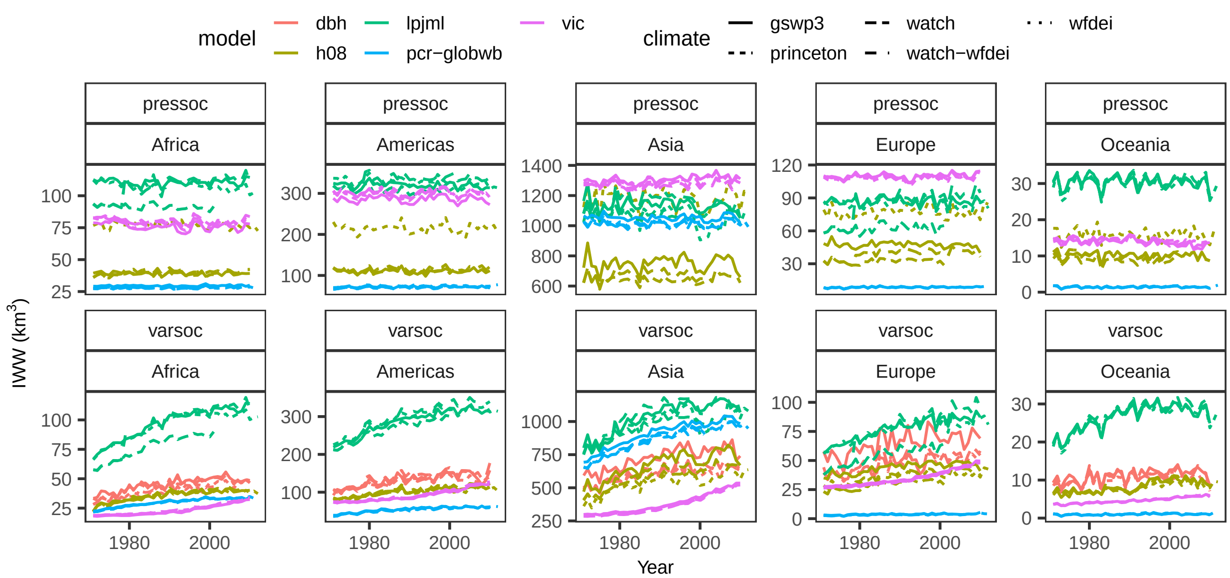
```
##  [8] "h08_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012.nc4"
##  [9] "h08_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012(1).nc4"
## [10] "h08_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2012.nc4"
## [11] "h08_watch_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2001.nc4"
## [12] "h08_watch_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2001.nc4"
## [13] "h08_watch-wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [14] "h08_watch-wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [15] "h08_wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc"
## [16] "lpjml_gswp3_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [17] "lpjml_gswp3_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [18] "lpjml_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012.nc4"
## [19] "lpjml_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2012.nc4"
## [20] "lpjml_watch_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2001.nc4"
## [21] "lpjml_watch_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2001.nc4"
## [22] "lpjml_watch-wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [23] "lpjml_watch-wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [24] "lpjml_wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc"
## [25] "pcr-globwb_gswp3_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [26] "pcr-globwb_gswp3_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [27] "pcr-globwb_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012.nc4"
## [28] "pcr-globwb_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2012.nc4"
## [29] "pcr-globwb_watch_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2001.nc4"
## [30] "pcr-globwb_watch_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2001.nc4"
## [31] "pcr-globwb_watch-wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [32] "pcr-globwb_watch-wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [33] "pcr-globwb_wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc"
## [34] "vic_gswp3_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [35] "vic_gswp3_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [36] "vic_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [37] "vic_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [38] "vic_watch_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2001.nc4"
## [39] "vic_watch_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2001.nc4"
## [40] "vic_watch-wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc4"
## [41] "vic_watch-wfdei_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2010.nc4"
## [42] "vic_wfdei_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2010.nc"
```

```r
# Name the slots ----------------------------------------------------------------

names(isimip.hist) <- paste(model.names, climate.scenarios, social.scenarios, sep = "/")

# Clean and bind dataset -------------------------------------------------------

isimip.dt <- rbindlist(isimip.hist, idcol = "model") %>%
  na.omit() %>%
  .[, model:= factor(model)] %>%
  .[, c("model", "climate", "social"):= tstrsplit(model, "/")]

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

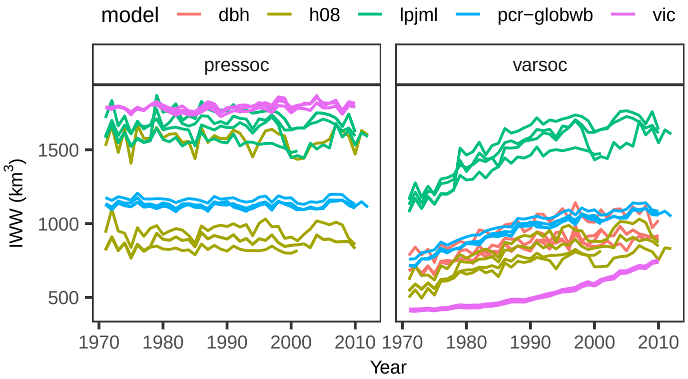
```
# Pressoc: constant human impacts in the form of dams and reservoirs
# varsoc: variable human impacts.
```

2.1.1 Plot data

```
# PLOT ISIMIP #################################################################

# Continental level ---------------------------------------------------------

isimip.dt[, sum(V1, na.rm = TRUE), .(Continent, model, year, climate, social)] %>%
  ggplot(., aes(year, V1, group = interaction(climate, model), color = model,
                linetype = climate)) +
  facet_wrap(social~Continent, scales = "free_y", ncol = 5) +
  geom_line() +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  theme_AP() +
  guides(color = guide_legend(nrow = 2)) +
  guides(linetype = guide_legend(nrow = 2)) +
  theme(legend.position = "top")
```




```
# Global level -------------------------------------------------------------

isimip.dt[, sum(V1, na.rm = TRUE), .(year, model, climate, social)] %>%
  ggplot(., aes(year, V1, group = interaction(climate, model), color = model)) +
  geom_line() +
  facet_wrap(~social) +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  theme_AP() +
  theme(legend.position = "top")
```



2.2 Predictions

```r
# RETRIEVE PROJECTIONS FROM ISIMIP #############################################

# Create vector with list of files ---------------------------------------------

path.projections <- "./files/isimip_future"
list.of.files.projections <- list.files(path.projections)
files.directory.projections <- paste(path.projections, list.of.files.projections, sep = "/")
variable <- "airrww"
start_year <- 2006

# Create parallel cluster ------------------------------------------------------

numCores <- detectCores() * 0.75
cl <- makeCluster(numCores)
registerDoParallel(cl)

# Run for loop -----------------------------------------------------------------

isimip.future <- foreach(i = 1:length(files.directory.projections),
                        .packages = c("data.table", "countrycode", "tidyverse",
                                      "sp", "rworldmap", "ncdf4")) %dopar% {

                                        get_isimip_fun(nc_file = files.directory.projections[i]
                                                       variable = variable,
                                                       start_year = start_year)
                                      }

# Stop the cluster after the computation ---------------------------------------

stopCluster(cl)
```

```r
# ARRANGE DATA #################################################################

# Number of files --------------------------------------------------------------
```

```
list.of.files.projections
```

```
##  [1] "h08_miroc5_ewembi_rcp26_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [2] "h08_miroc5_ewembi_rcp26_rcp26soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [3] "h08_miroc5_ewembi_rcp60_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [4] "h08_miroc5_ewembi_rcp60_rcp60soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [5] "h08_miroc5_ewembi_rcp85_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [6] "lpjml_miroc5_ewembi_rcp26_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [7] "lpjml_miroc5_ewembi_rcp26_rcp26soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [8] "lpjml_miroc5_ewembi_rcp60_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
##  [9] "lpjml_miroc5_ewembi_rcp85_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
## [10] "mpi-hm_miroc5_ewembi_picontrol_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
## [11] "mpi-hm_miroc5_ewembi_rcp26_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
## [12] "mpi-hm_miroc5_ewembi_rcp60_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
## [13] "pcr-globwb_miroc5_ewembi_rcp26_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
## [14] "pcr-globwb_miroc5_ewembi_rcp60_2005soc_co2_airrww_global_monthly_2006_2099.nc4"
```

```
# Arrange names ---------------------------------------------------------------

model.names <- sub("^(.*?)_.*", "\\1", list.of.files.projections)
pattern <- "ewembi_(.*?)soc"
climate <- sub(".*ewembi_(.*?)soc.*", "\\1", list.of.files.projections)
names(isimip.future) <- paste(model.names, climate, sep = "/")

# Clean and bind dataset -------------------------------------------------------

isimip.future.dt <- rbindlist(isimip.future, idcol = "model") %>%
  na.omit() %>%
  .[, model:= factor(model)] %>%
  .[, year:= as.numeric(year)]

isimip.future.dt[, c("model", "climate") := tstrsplit(model, "/")]

# Export ----------------------------------------------------------------------

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

```
# PLOT ISIMIP ##################################################################

# Continental level ------------------------------------------------------------

isimip.future.dt[, sum(V1, na.rm = TRUE), .(year, Continent, model, climate)] %>%
  .[, climate:= gsub("_", "\\\\_", climate)] %>%
  ggplot(., aes(year, V1, group = climate, color = climate)) +
  facet_wrap(model~Continent, scales = "free_y", ncol = 5) +
  geom_line() +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  theme_AP() +
```

```
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme(legend.position = "top")
```


```
# PLOT ISIMIP MERGED ###############################################################

a <- isimip.future.dt[, sum(V1, na.rm = TRUE), .(year, Continent, model, climate)] %>%
  ggplot(., aes(year, V1, group = interaction(climate, model), color = model)) +
  facet_wrap(~Continent, scales = "free_y", ncol = 5) +
  geom_line() +
  scale_color_manual(name = "", values = wes_palette(name = selected.palette)) +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme_AP() +
  theme(legend.position = "top")

b <- isimip.future.dt[, sum(V1, na.rm = TRUE), .(year, Continent, model, climate)] %>%
  ggplot(., aes(year, V1, group = interaction(climate, model), color = climate)) +
  facet_wrap(~Continent, scales = "free_y", ncol = 5) +
  geom_line() +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme_AP() +
  theme(legend.position = "top")
```

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

2.3 ANOVA

```r
# ANOVA ##########################################################################

# Arrange ISIMIP datasets ---------------------------------------------------------

isimip.full <- isimip.dt[social == "varsoc"][, context:= "historic"] %>%
  rbind(., isimip.future.dt[, context:= "prediction"], fill = TRUE) %>%
  .[, social:= NULL]

isimip.anova <- isimip.full[, .(estimation = sum(V1)),
                            .(Continent, climate, context, model, year)]

# ARRANGE DATA ###################################################################

columns_to_factor <- c("Continent", "climate", "model")
isimip.full[, (columns_to_factor):= lapply(.SD, as.factor), .SDcols = (columns_to_factor)]
isimip.anova[, (columns_to_factor):= lapply(.SD, as.factor), .SDcols = (columns_to_factor)]

# RUN MODEL AND ANALYSIS OF VARIANCE #############################################

# List of models ------------------------------------------------------------------

functions <- list(lmm = lmm_fun,
```

```r
                  gamm = gamm_fun,
                  rf = rf_fun,
                  bayes = bayes_fun)

# Apply each function to the data and combine results ---------------------------

results <- mclapply(names(functions), function(fun_name) {

  isimip.anova[, functions[[fun_name]](.SD), .(Continent, context)]

},
mc.cores = detectCores() * 0.75)
```

```r
# PLOT RESULTS ################################################################

results
```

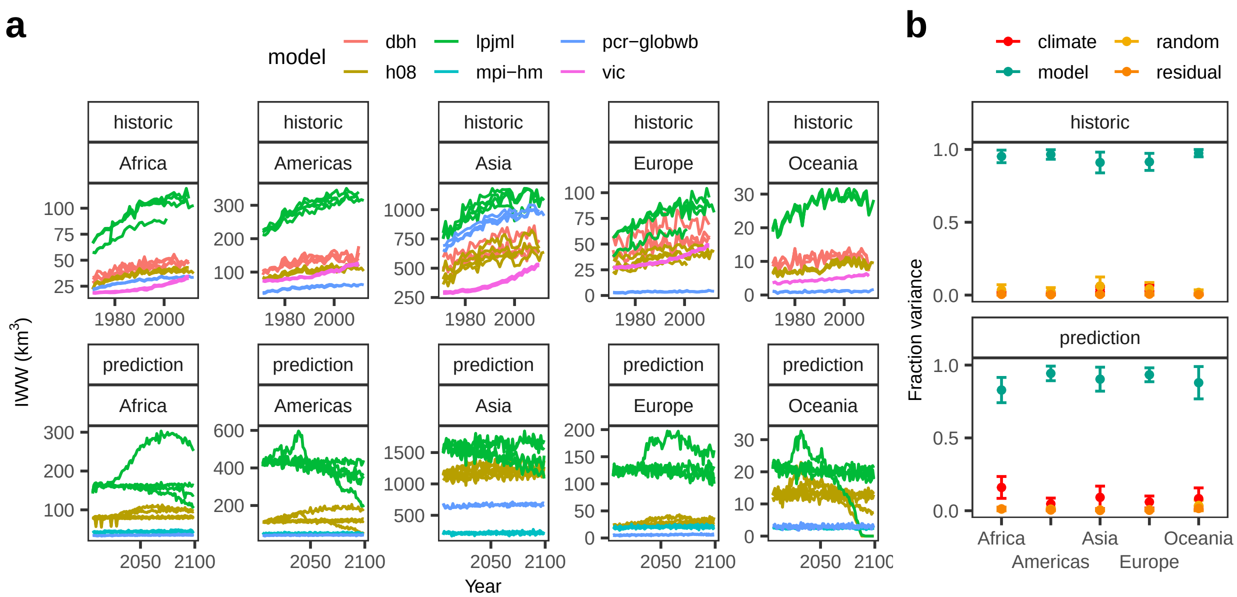
```
## [[1]]
##     Continent    context climate_variance model_variance random_variance
##        <fctr>     <char>            <num>          <num>           <num>
##   1:       Asia   historic     0.0182441856      0.9815439    1.604121e-04
##   2:     Europe   historic     0.0265735831      0.9732386    7.928135e-05
##   3:     Africa   historic     0.0046293623      0.9952289    7.974990e-05
##   4:   Americas   historic     0.0015875370      0.9983346    4.897266e-05
##   5:    Oceania   historic     0.0003011393      0.9996366    2.836314e-05
##   6:       Asia prediction     0.0144443043      0.9855396    1.802974e-21
##   7:     Europe prediction     0.0188199322      0.9811568    9.455137e-07
##   8:     Africa prediction     0.0847272814      0.9151935    1.015636e-22
##   9:   Americas prediction     0.0070916322      0.9928739    2.351915e-06
##  10:    Oceania prediction     0.0099009112      0.9899272    2.436002e-05
##     residual_variance
##                 <num>
##   1:       5.146166e-05
##   2:       1.085044e-04
##   3:       6.196443e-05
##   4:       2.885478e-05
##   5:       3.387542e-05
##   6:       1.606501e-05
##   7:       2.232385e-05
##   8:       7.922237e-05
##   9:       3.208885e-05
##  10:       1.475610e-04
##
## [[2]]
##     Continent    context climate_variance model_variance random_variance
##        <fctr>     <char>            <num>          <num>           <num>
##   1:       Asia   historic     0.0582396865      0.9326743    3.492825e-06
##   2:     Europe   historic     0.0665137789      0.9204392    2.795270e-04
```

```
##  3:    Africa   historic     0.0058233555      0.9841575    3.265345e-04
##  4:  Americas   historic     0.0027474682      0.9923858    3.009048e-05
##  5:   Oceania   historic     0.0004492087      0.9905800    2.549049e-03
##  6:      Asia prediction     0.0233855736      0.9728348    1.274156e-10
##  7:    Europe prediction     0.0472909695      0.9462095    8.071134e-05
##  8:    Africa prediction     0.1977722547      0.7786924    6.947724e-05
##  9:  Americas prediction     0.0228104251      0.9679342    9.751115e-06
## 10:   Oceania prediction     0.0213692004      0.9437060    3.414224e-03
##     residual_variance
##                 <num>
##  1:       0.009082472
##  2:       0.012767490
##  3:       0.009692578
##  4:       0.004836623
##  5:       0.006421754
##  6:       0.003779619
##  7:       0.006418860
##  8:       0.023465907
##  9:       0.009245593
## 10:       0.031510534
##
## [[3]]
##     Continent    context climate_variance model_variance random_variance
##        <fctr>     <char>            <num>          <num>           <num>
##  1:      Asia   historic       0.03651589      0.8392607      0.12421897
##  2:    Europe   historic       0.05732503      0.8561269      0.08615885
##  3:    Africa   historic       0.01925291      0.9091631      0.07124754
##  4:  Americas   historic       0.01621633      0.9325713      0.05118237
##  5:   Oceania   historic       0.01243839      0.9494631      0.03595673
##  6:      Asia prediction       0.16884042      0.8208680      0.01029125
##  7:    Europe prediction       0.10083726      0.8857703      0.01334813
##  8:    Africa prediction       0.23492099      0.7417898      0.02324906
##  9:  Americas prediction       0.08681625      0.8927305      0.02044727
## 10:   Oceania prediction       0.15674244      0.7680757      0.07252973
##     residual_variance
##                <lgcl>
##  1:                NA
##  2:                NA
##  3:                NA
##  4:                NA
##  5:                NA
##  6:                NA
##  7:                NA
##  8:                NA
##  9:                NA
## 10:                NA
##
## [[4]]
```

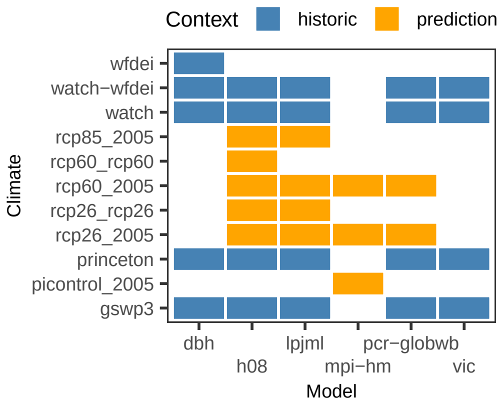
```
##     Continent    context climate_variance model_variance random_variance
##        <fctr>     <char>            <num>          <num>           <num>
##  1:      Asia   historic     0.0585877490      0.9114970    2.307509e-02
##  2:    Europe   historic     0.0679479627      0.9104088    9.472626e-03
##  3:    Africa   historic     0.0069267842      0.9713291    1.254960e-02
##  4:  Americas   historic     0.0030365269      0.9846379    7.963846e-03
##  5:   Oceania   historic     0.0003067787      0.9899282    4.641493e-03
##  6:      Asia prediction     0.0233300054      0.9728583    1.131701e-05
##  7:    Europe prediction     0.0469288511      0.9463859    2.596976e-04
##  8:    Africa prediction     0.1977676855      0.7778277    5.262552e-05
##  9:  Americas prediction     0.0228624389      0.9669629    7.071621e-04
## 10:   Oceania prediction     0.0208228448      0.9423662    5.325046e-03
##     residual_variance
##                 <num>
##  1:       0.006840208
##  2:       0.012170596
##  3:       0.009194540
##  4:       0.004361695
##  5:       0.005123568
##  6:       0.003800369
##  7:       0.006425531
##  8:       0.024351981
##  9:       0.009467473
## 10:       0.031485861
```

```r
results.dt <- rbindlist(results)

a <- isimip.full[, .(estimation = sum(V1)), .(model, Continent, climate, year, context)] %>%
  ggplot(., aes(year, estimation, color = model, group = interaction(climate, model))) +
  geom_line() +
  facet_wrap(context~Continent, scale = "free", ncol = 5) +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme_AP() +
  guides(colour = guide_legend(nrow = 2)) +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  theme(legend.position = "top",
        legend.box.spacing = unit(0, "pt"))

b <- results.dt %>%
  melt(., measure.vars = c("climate_variance", "model_variance", "random_variance",
                           "residual_variance")) %>%
  .[, .(min = min(value, na.rm = TRUE),
        max = max(value, na.rm = TRUE)), .(Continent, context, variable)] %>%
  .[, variance:= tstrsplit(variable, "_", fixed = TRUE)[[1]]] %>%
  ggplot(., aes(x = Continent, ymin = min, ymax = max, y = (min + max) / 2, color = variance))
  geom_errorbar(width = 0.2) +
  geom_point(size = 1) +
  scale_color_manual(name = "", values=wes_palette(selected.palette, n = 4)) +
```

```
  labs(x = "", y = "Fraction variance") +
  facet_wrap(~context, ncol = 1) +
  theme(legend.position = "top") +
  scale_y_continuous(breaks = breaks_pretty(n = 3)) +
  theme_AP() +
  theme(legend.position = "top") +
  guides(color = guide_legend(nrow = 2)) +
  theme(legend.position = "top") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))

plot_grid(a, b, ncol = 2, labels = "auto", rel_widths = c(0.72, 0.28))
```




```
# COUNT COMBINATIONS OF MODEL AND CLIMATE ########################################

unique(isimip.full[, .(model, climate, context)]) %>%
  ggplot(., aes(x = model, y = climate, fill = context)) +
  geom_tile(color = "white", size = 0.5) +
  scale_fill_manual(values = c("historic" = "steelblue", "prediction" = "orange")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Model", y = "Climate", fill = "Context") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  theme_AP() +
  theme(legend.position = "top")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



3 Khan et al dataset

```
# KHAN ET AL 2023 DATASET ######################################################

path.projections <- "./files/khan_et_al_2023"
list.of.files <- list.files(path.projections, pattern = "\\.csv$")
combinations <- lapply(list.of.files, function(x) strsplit(x, "_")[[1]][1:4]) %>%
  do.call(rbind, .) %>%
  data.frame()
colnames(combinations) <- c("SSP", "RCP", "Climate", "Use")

# READ FILES IN PARALLEL #######################################################

# Create parallel cluste -------------------------------------------------------

numCores <- detectCores() * 0.75
cl <- makeCluster(numCores)
registerDoParallel(cl)

# Run for loop -----------------------------------------------------------------

result <- foreach(i = 1:length(list.of.files),
                  .combine = "rbind",
                  .packages = c("data.table", "countrycode",
                                "sp", "rworldmap")) %dopar% {

                                  out <- fread(paste("./files/khan_et_al_2023/", list.of.files
                                  out[, `:=`(SSP = combinations[i, 1],
                                             RCP = combinations[i, 2],
                                             Climate = combinations[i, 3],
                                             Use = combinations[i, 4])]

                                  Country <- coords2country(out[1:nrow(out), 2:3])

                                  df <- cbind(Country, out)
```

```r
                                   df[, Continent := countrycode(Country, origin = "country.name
                                   df[, Dataset := list.of.files[i]]

                                   df
                                 }

# Stop the cluster after the computation ----------------------------------------

stopCluster(cl)
```

```r
# ARRANGE DATA ##################################################################

numeric_cols <- grep("^[0-9]+$", names(result), value = TRUE)
khan.dt <- melt(result, measure.vars = numeric_cols, variable.name = "Year") %>%
  .[, Year:= as.numeric(as.character(Year))] %>%
  .[, model:= "GCAM"] %>%
  na.omit()

# EXPORT DATA ###################################################################

khan.dt.continent <- khan.dt[, .(estimation = sum(value)),
                             .(Year, Continent, Use, RCP, SSP, Climate, Dataset, model)] %>%
  .[, climate:= paste(Climate, RCP, SSP, sep = "_")]

fwrite(khan.dt.continent, "khan.dt.continent.csv")
```

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

# Continental -------------------------------------------------------------------

plot.khan.continental <- khan.dt.continent %>%
  ggplot(., aes(Year, estimation, color = Continent, group = interaction(Dataset, Continent)))
  geom_line(alpha = 0.3) +
  facet_wrap(~Use) +
  theme_AP() +
  theme(legend.position = "top") +
  labs(x = "", y = bquote("km"^3))

plot.khan.continental
```

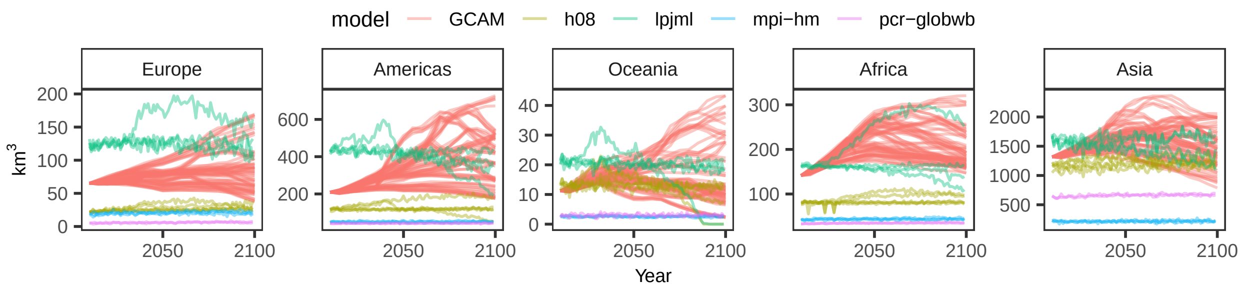
```r
# PLOT ##########################################################################

# Global ------------------------------------------------------------------------

plot.khan.global <- khan.dt[, sum(value), .(Year, Use, Dataset)] %>%
  ggplot(., aes(Year, V1, group = Dataset)) +
  geom_line(alpha = 0.3) +
```

```
  facet_wrap(~Use) +
  theme_AP() +
  theme(legend.position = "top") +
  labs(x = "Year", y = bquote("km"^3))

plot.khan.global
```

```
# MERGE KHAN ET AL DATASETS ####################################################

plot_grid(plot.khan.continental, plot.khan.global, ncol = 1, labels = "auto",
          rel_heights = c(0.53, 0.47))
```

```
# PLOT SSPS VS RCPS ############################################################

khan.dt[, sum(value), .(Year, Use, Dataset, RCP, SSP)] %>%
  ggplot(., aes(Year, V1, group = Dataset, color = Use)) +
  geom_line() +
  facet_grid(RCP~SSP) +
  theme_AP() +
  theme(legend.position = "top") +
  labs(x = "Year", y = bquote("km"^3))
```

```
# MERGE KHAN ET AL DATA WITH ISIMIP ############################################

# Arrange data -----------------------------------------------------------------

khan.dt.continent <- fread("khan.dt.continent.csv")

khan.dt2 <- khan.dt.continent[Use == "withdrawals", .(model, Continent, climate, Year, estimat
  setnames(., "Year", "year")

# Extract prediction data from ISIMIP ------------------------------------------

isimip.full2 <- isimip.full[context == "prediction" & year >= 2010,
            .(estimation = sum(V1)), .(model, Continent, climate, year, context)] %>%
  .[, context:= NULL]

# Merge and plot ---------------------------------------------------------------

merged.dt <- rbind(khan.dt2, isimip.full2)

ggplot(merged.dt, aes(year, estimation, group = interaction(climate, model), color = model)) +
  geom_line(alpha = 0.4) +
  facet_wrap(~Continent, scale = "free_y", ncol = 5) +
  theme_AP() +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme(legend.position = "top") +
  labs(x = "Year", y = bquote("km"^3))
```




```
# Calculate the min and max in 2030-2050 given uncertainty and the global level -----

merged.dt[year %in% c(2030, 2040, 2050),
          .(min = min(estimation), max = max(estimation)), .(Continent, year)] %>%
  .[, .(sum_min = sum(min), sum_max = sum(max)), year]
```

```
##     year  sum_min  sum_max
##    <num>    <num>    <num>
## 1:  2030 272.8320 2529.235
## 2:  2040 281.8063 2958.560
## 3:  2050 278.4169 3188.283
```

4 Bibliographical study

```
# NAOMI DATASET ##################################################################

references.projected <- data.table(read.xlsx("./data/references_projection.xlsx")) %>%
  .[, focus:= "projected"]

references.current <- data.table(read.xlsx("./data/references_current.xlsx")) %>%
  .[, focus:= "current"]

references.full.dt <- rbind(references.projected, references.current) %>%
  .[, study:= paste(author, model, climate.scenario, sep = ".")]

# CLEAN THE DATASET ############################################################

colnames_vector <- c("title", "author", "region")

# Remove leading and trailing spaces -------------------------------------------

references.full.dt[, (colnames_vector):= lapply(.SD, trimws), .SDcols = (colnames_vector)]
references.full.dt[, (colnames_vector):= lapply(.SD, str_squish), .SDcols = (colnames_vector)]

# Lowercaps --------------------------------------------------------------------

references.full.dt[, (colnames_vector):= lapply(.SD, tolower), .SDcols = (colnames_vector)]

# Remove multiple spaces -------------------------------------------------------
```

```r
references.full.dt[, (colnames_vector):= lapply(.SD, function(x)
  gsub("\\s+", " ", x)), .SDcols = (colnames_vector)]

# Correct America ----------------------------------------------------------

references.full.dt[, region:= ifelse(region == "america", "americas", region)]

# Extract the publication year ---------------------------------------------------

references.full.dt[, publication.date:= str_extract(author, "\\d{4}")] %>%
  .[, publication.date:= as.numeric(publication.date)]
```

```r
# FEATURES OF THE DATASET #######################################################

# Name of different studies ---------------------------------------------------

sort(unique(references.full.dt[variable == "iww" & region == "global", title]))
```

```
##  [1] "a global water scarcity assessment under shared socio-economic pathways - part 2: wate
##  [2] "a pathway of global food supply adaptation in a world with increasingly constrained gr
##  [3] "agricultural green and blue water consumption and its influence on the global water sy
##  [4] "an estimation of global virtual water flow and sources of water withdrawal for major c
##  [5] "an integrated assessment of global and regional water demands for electricity generati
##  [6] "an interpreted model for the assessment of global water resources - part 2: applicatio
##  [7] "appraisal and assessment of world water resources"
##  [8] "aquastat: fao's global information system on water and agriculture"
##  [9] "climate change impacts on irrigation water requirements: effects of mitigation, 1990-2
## [10] "climate impacts on global irrigation requirements under 19 gcms, simulated with a vege
## [11] "climate policy implications for agricultural water demand"
## [12] "future long-term changes in global water resources driven by socio-economic and clima
## [13] "global and regional evaluation of energy for water"
## [14] "global impacts of conversions from natural to agricultural ecosystems on water resour
## [15] "global irrigation characteristics and effects simulated by fully coupled land surface
## [16] "global irrigation water demand: variability and uncertainties arising from agricultura
## [17] "global modeling of irrigation water requirements"
## [18] "global monthly sectoral water use for 2010-2100 at 0.5° resolution across alternative
## [19] "global water demand and supply projections"
## [20] "globwat - a global water balance model to assess water use in irrigated agriculture"
## [21] "how can we cope with the water resources situation by the year 2050?"
## [22] "human appropriation of renewable fresh water"
## [23] "implementation and evaluation of irrigation techniques in the community land model"
## [24] "incorporating anthropogenic water regulation modules into a land surface model"
## [25] "incorporation of groundwater pumping in a global land surface model with the represen
## [26] "isimip database"
## [27] "long-term global water projections using six socioeconomic scenarios in an intgrated a
## [28] "modelling global water stress of the recent past: on the relative importance of trends
## [29] "multimodel projections and uncertainties of irrigation water demand under climate chan
```

```
## [30] "pcr-globwb 2: a 5 arcmin global hydrological and water resources model"
## [31] "recent global cropland water consumption constrained by observations"
## [32] "reconciling irrigated food production with environmental flows for sustainable develo
## [33] "reconstructing 20th century global hydrography: a contribution to the global terrestri
## [34] "the state of the world's land and water resources for food and agriculture"
## [35] "the world’s water, 2000-2001: the biennial report on freshwater resources"
## [36] "water 2050. moving toward a sustainable vision fot the earth's fresh water"
## [37] "water and sustainability. global pattern and long-range problems"
## [38] "world agriculture towards 2030/2055"
## [39] "world water demand and supply, 1990 to 2025: scenarios and issues"
## [40] "world water in 2025 - global modeling and scenario analysis for the world commission
## [41] "world water resources and their future"
```

```
# Number of data points ------------------------------------------------------

nrow(references.full.dt[variable == "iww" & region == "global"])
```

```
## [1] 1194
```

```
# Number of different studies per variable ------------------------------------------

references.full.dt[region == "global", unique(title), variable] %>%
  .[, .N, variable]
```

```
##    variable     N
##      <char> <int>
## 1:      iww    42
## 2:      tww    27
## 3:      iwc    17
## 4:      twc     6
## 5:      iwr     2
```

```
# Number of data points for 2000, 2050, 2070, 2100 -----------------------------

references.full.dt[variable == "iww" & region == "global" &
                     estimation.year %in% c(2000, 2050, 2070, 2100), .N, estimation.year]
```

```
##    estimation.year     N
##              <num> <int>
## 1:            2000    55
## 2:            2070   119
## 3:            2100   106
## 4:            2050    98
```

```
# Number of unique studies estimating for 2000, 2050, 2070, 2100 ---------------

references.full.dt[variable == "iww" & region == "global" &
                     estimation.year %in% c(2000, 2050, 2070, 2100), unique(title), estimation
  .[, .N, estimation.year]
```

```
##    estimation.year     N
```

```
##                <num> <int>
## 1:              2000    17
## 2:              2070     4
## 3:              2100     3
## 4:              2050     7
```

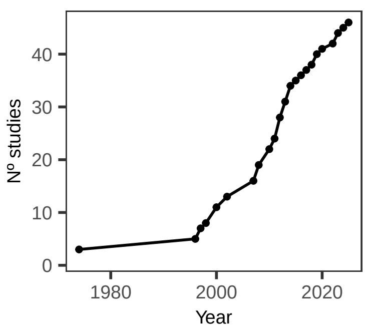
```r
# Number of data points for every targeted year -----------------------------

references.full.dt[variable == "iww" & region == "global", .N, estimation.year] %>%
  .[order(estimation.year)]
```

```
##     estimation.year     N
##                <num> <int>
##  1:             1900     2
##  2:             1940     2
##  3:             1950     2
##  4:             1960     3
##  5:             1970     3
##  6:             1975    22
##  7:             1980    26
##  8:             1983     1
##  9:             1985    26
## 10:             1990    25
## 11:             1994     6
## 12:             1995    38
## 13:             1996     2
## 14:             2000    55
## 15:             2002     2
## 16:             2003     1
## 17:             2004     1
## 18:             2005    19
## 19:             2006     2
## 20:             2008     1
## 21:             2010   100
## 22:             2015     3
## 23:             2020    84
## 24:             2021     1
## 25:             2025     6
## 26:             2030    82
## 27:             2040    93
## 28:             2050    98
## 29:             2055     2
## 30:             2060    82
## 31:             2070   119
## 32:             2075     2
## 33:             2080    91
## 34:             2090    79
## 35:             2095     7
```

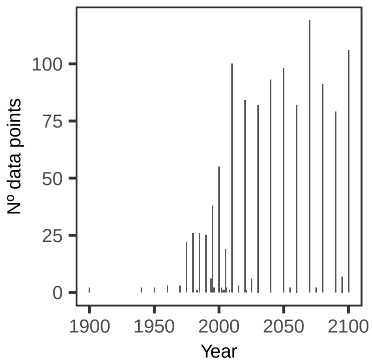
```
## 36:              2100   106
##     estimation.year     N
```

```
# Cumulative sum of published studies ------------------------------------------

cumulative.iww <- references.full.dt[, .(title, publication.date, variable)] %>%
  .[variable == "iww"] %>%
  .[!duplicated(.)] %>%
  setorder(., publication.date) %>%
  .[, .N, publication.date] %>%
  .[, cumulative_sum := cumsum(N)] %>%
  ggplot(., aes(publication.date, cumulative_sum)) +
  geom_line() +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  geom_point(size = 0.7) +
  theme_AP() +
  labs(x = "Year", y = "Nº studies")

cumulative.iww
```

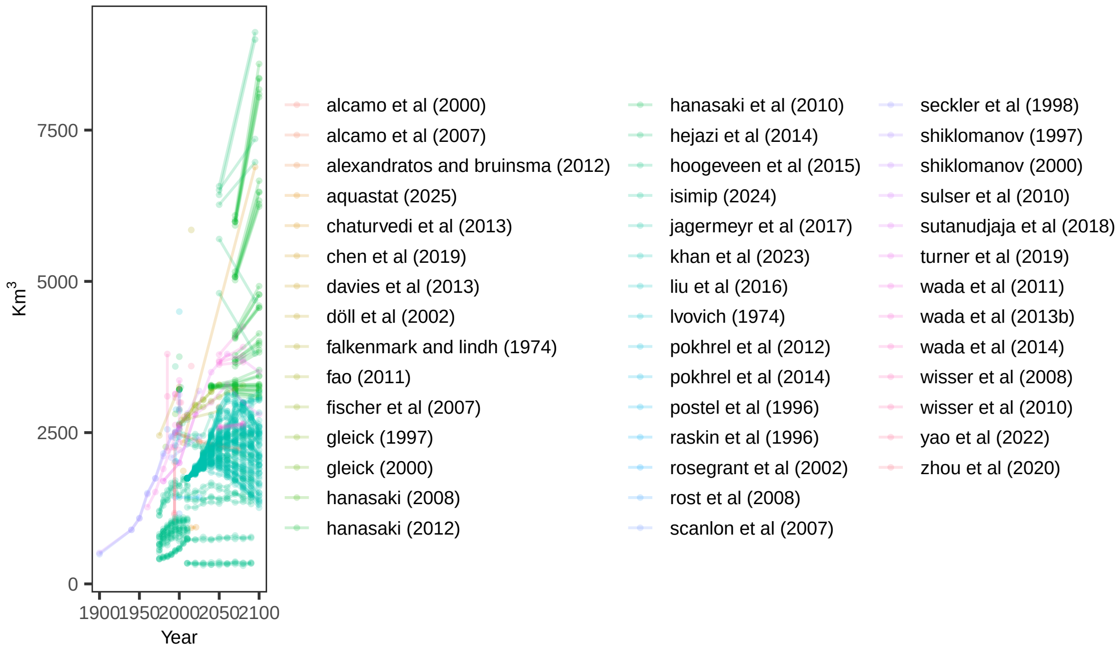
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# DISTRIBUTION OF DATA POINTS THROUGH YEARS @##################################

references.full.dt[variable == "iww" & region == "global", .N, estimation.year] %>%
  ggplot(., aes(estimation.year, N)) +
  geom_bar(stat = "identity") +
  labs(x = "Year", y = "Nº data points") +
  theme_AP()
```

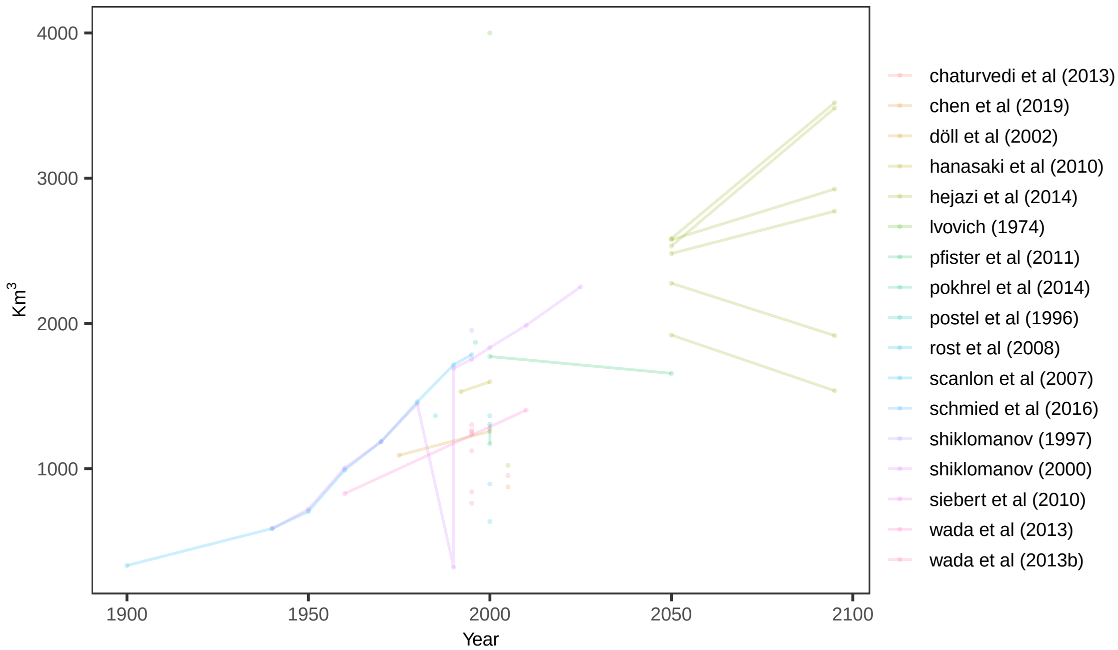



```
# PLOT ALL ESTIMATIONS ##########################################################

def.alpha <- 0.2

plot.iww <- references.full.dt[variable == "iww" & region == "global"] %>%
  .[, .(author, study, estimation.year, value)] %>%
  na.omit() %>%
  ggplot(., aes(estimation.year, value, color = author, group = study)) +
  geom_point(alpha = def.alpha, size = 0.5) +
  labs(x = "Year", y = bquote("Km"^3)) +
  scale_color_discrete(name = "") +
  geom_line(alpha = def.alpha) +
  theme_AP()

plot.iww
```




```
references.full.dt[variable == "iwc" & region == "global"] %>%
  .[, .(author, study, estimation.year, value)] %>%
  na.omit() %>%
```

```
  ggplot(., aes(estimation.year, value, color = author, group = study)) +
  geom_point(alpha = def.alpha, size = 0.2) +
  labs(x = "Year", y = bquote("Km"^3)) +
  scale_color_discrete(name = "") +
  geom_line(alpha = def.alpha) +
  theme_AP()
```



4.1 The garden of forking paths

```
# DEFINE THE UNCERTAINTY SPACE #################################################

# Target year -----------------------------------------------------------------

target_year <- c(2000, 2050, 2070, 2100)

# Target year interval --------------------------------------------------------

target_year_interval <- c("yes", "no")

# Interval publication --------------------------------------------------------

interval <- c(10, 15, 20)

# Metrics of study ------------------------------------------------------------

metrics <- c("cv", "range", "sd", "var", "entropy", "iqr")

# Inclusion criteria ----------------------------------------------------------
```

```
inclusion_criteria <- c("all", "exclude_before_1990")

# Rolling windows ---------------------------------------------------------------

rolling_window_factor <- c(1, 0.5)

# Define the forking paths ------------------------------------------------------

forking_paths <- expand.grid(target_year = target_year,
                             target_year_interval = target_year_interval,
                             interval = interval,
                             inclusion_criteria = inclusion_criteria,
                             rolling_window_factor = rolling_window_factor,
                             metric = c(metrics, paste(metrics, "_normalized", sep = ""))) %>%
  data.table()

# Number of simulations ---------------------------------------------------------

nrow(forking_paths)
```

```
## [1] 1152
```

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

trend <- list()

for (i in 1:nrow(forking_paths)) {

  trend[[i]] <- forking_paths_fun(dt = references.full.dt,
                                  target_year = forking_paths[[i, "target_year"]],
                                  target_year_interval = forking_paths[[i, "target_year_interva
                                  interval = forking_paths[[i, "interval"]],
                                  rolling_window_factor = forking_paths[[i, "rolling_window_fac
                                  inclusion_criteria = forking_paths[[i, "inclusion_criteria"]]
                                  metric = forking_paths[[i, "metric"]])
}
```

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

output.dt <- lapply(trend, function(x) x[["results"]]) %>%
  do.call(rbind, .) %>%
  data.table() %>%
  setnames(., "V1", "trend")

final.dt <- cbind(forking_paths, output.dt)

# Export simulations ------------------------------------------------------------
```

```r
fwrite(final.dt, "forking.paths.dataset.csv")

# Print the fraction of simulations in each classification ----------------------

final.dt %>%
  .[, .(total = .N), trend] %>%
  .[, fraction:= total / nrow(output.dt)] %>%
  print()
```

```
##           trend total   fraction
##          <char> <int>      <num>
## 1:       Random   515 0.44704861
## 2:    Ascending   300 0.26041667
## 3:   Descending   289 0.25086806
## 4: single point    48 0.04166667
```

```r
# Now remove all simulations that produced just one single point ---------------

final.dt <- final.dt[!trend == "single point"]

# Simulations that did not lead to a reduction in uncertainty ------------------

final.dt %>%
  .[, .(total = .N), trend] %>%
  .[, fraction:= total / nrow(output.dt)] %>%
  .[!trend == "Descending"] %>%
  .[, sum(fraction)]
```

```
## [1] 0.7074653
```

```r
# PLOTS FORKING PATHS EXAMPLES #################################################

plots.dt <- lapply(trend, function(x) x[["plot"]])

random.plots <- c(1, 986, 345)
decreasing.plots <- c(1093, 556, 4)
increasing.plots <- c(10, 602, 770)

out.random <- out.decreasing <- out.increasing <- list()

for (i in 1:length(random.plots)) {

  out.random[[i]] <- plot_plots_forking_paths_fun(random.plots[i])
  out.decreasing[[i]] <- plot_plots_forking_paths_fun(decreasing.plots[i])
  out.increasing[[i]] <- plot_plots_forking_paths_fun(increasing.plots[i])
}

pt.random <- plot_grid(out.random[[1]] + geom_smooth() + labs(x = "", y = "+ Uncertainty"),
                       out.random[[2]] + geom_smooth() + labs(x = "", y = ""),
```

```
                        out.random[[3]] + geom_smooth() + labs(x = "", y = ""),
                        ncol = 3)
```

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

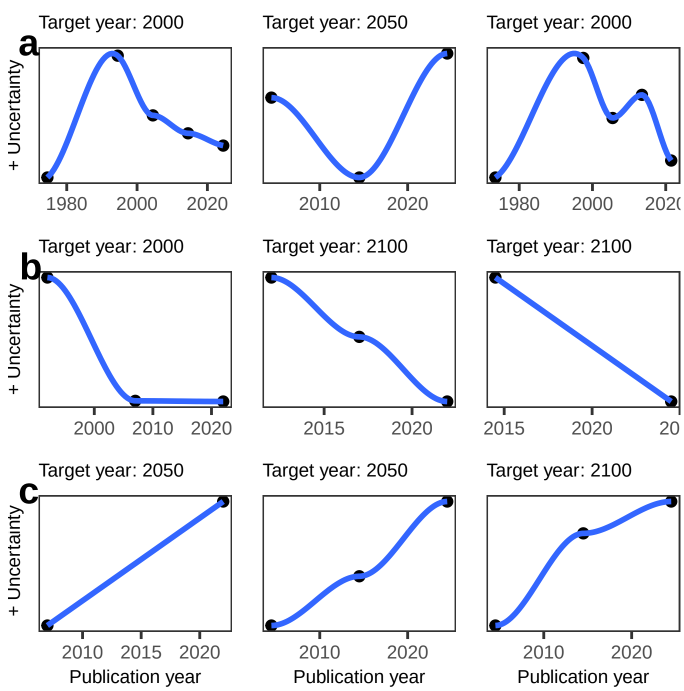
```
pt.decreasing <- plot_grid(out.decreasing[[1]] + geom_smooth() + labs(x = "", y = "+ Uncertain
                           out.decreasing[[2]] + geom_smooth() + labs(x = "", y = ""),
                           out.decreasing[[3]] + geom_smooth(method = "lm", se = F) + labs(x =
                           ncol = 3)
```

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

```
pt.increasing <- plot_grid(out.increasing[[1]] + geom_smooth(method = "lm", se = F),
                           out.increasing[[2]] + geom_smooth() + labs(x = "Publication year",
                           out.increasing[[3]] + geom_smooth() + labs(x = "Publication year",
                           ncol = 3)
```

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

```
plot_grid(pt.random, pt.decreasing, pt.increasing, ncol = 1, labels = "auto")
```

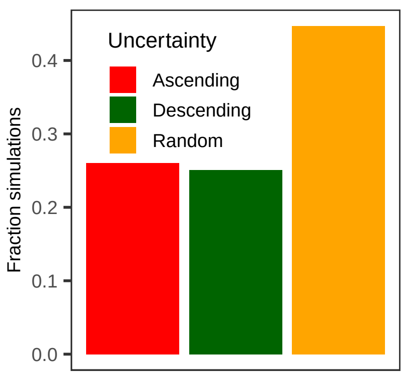



```
# PLOT RESULTS ##################################################################

selected_colors <- c("Ascending" = "red", "Descending" = "darkgreen", "Random" = "orange")

```

```
plot.fraction <- final.dt[, .(total = .N), trend] %>%
  .[, fraction:= total / nrow(output.dt)] %>%
  ggplot(., aes(trend, fraction, fill = trend)) +
  geom_bar(stat = "identity") +
  labs(x = "", y = "Fraction simulations") +
  scale_fill_manual(values = selected_colors, name = "Uncertainty") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  theme_AP() +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        legend.position = c(0.33, 0.77))
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plot.fraction
```

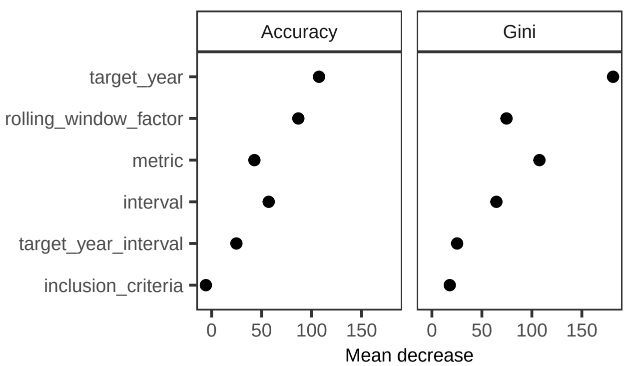



```
# RANDOM FOREST ################################################################

# Convert categorical variables to factors -------------------------------------

df <- data.frame(final.dt)
df$inclusion_criteria <- as.factor(final.dt$inclusion_criteria)
df$metric <- as.factor(final.dt$metric)
df$trend <- as.factor(df$trend)
df$target_year_interval <- as.factor(df$target_year_interval)

# Train the model --------------------------------------------------------------

rf_model <- randomForest(trend ~ target_year + target_year_interval + interval +
                           inclusion_criteria + rolling_window_factor + metric,
                         data = df, importance = TRUE)

# View variable importance -----------------------------------------------------
```

```r
dt_rf_model <- data.frame(importance(rf_model))
dt_rf_model
```

```
##                       Ascending Descending    Random MeanDecreaseAccuracy
## target_year            58.82274  84.764119 94.622667           107.382975
## target_year_interval   25.70447   9.853205  8.164398            24.733313
## interval               39.06084  30.958593 46.892258            57.058990
## inclusion_criteria    -24.19508   4.749013  4.471091            -5.770201
## rolling_window_factor  49.95345  34.466797 80.799476            86.706311
## metric                 37.90890  27.888115 21.941872            42.765101
##                       MeanDecreaseGini
## target_year                  181.20505
## target_year_interval          25.24297
## interval                      64.54198
## inclusion_criteria            17.88005
## rolling_window_factor         74.63270
## metric                       107.56085
```

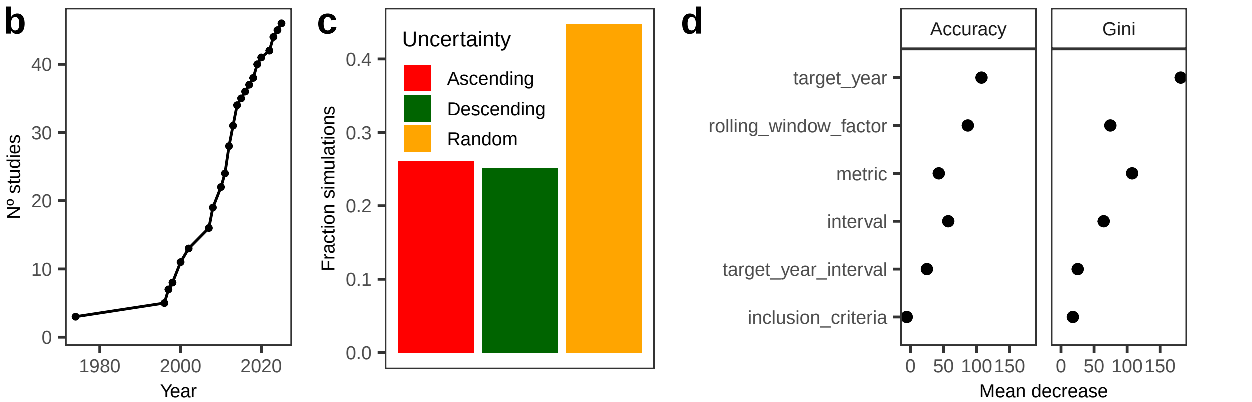
```r
# Plot -------------------------------------------------------------------------

plot.rf <- dt_rf_model %>%
  rownames_to_column(., var = "factors") %>%
  data.table() %>%
  setnames(., c("MeanDecreaseAccuracy", "MeanDecreaseGini"),
           c("Accuracy", "Gini")) %>%
  melt(., measure.vars = c("Accuracy", "Gini")) %>%
  ggplot(., aes(reorder(factors, value), value)) +
  geom_point() +
  coord_flip() +
  facet_wrap(~variable) +
  scale_y_continuous(breaks = breaks_pretty(n = 3)) +
  labs(x = "", y = "Mean decrease") +
  theme_AP()

plot.rf
```



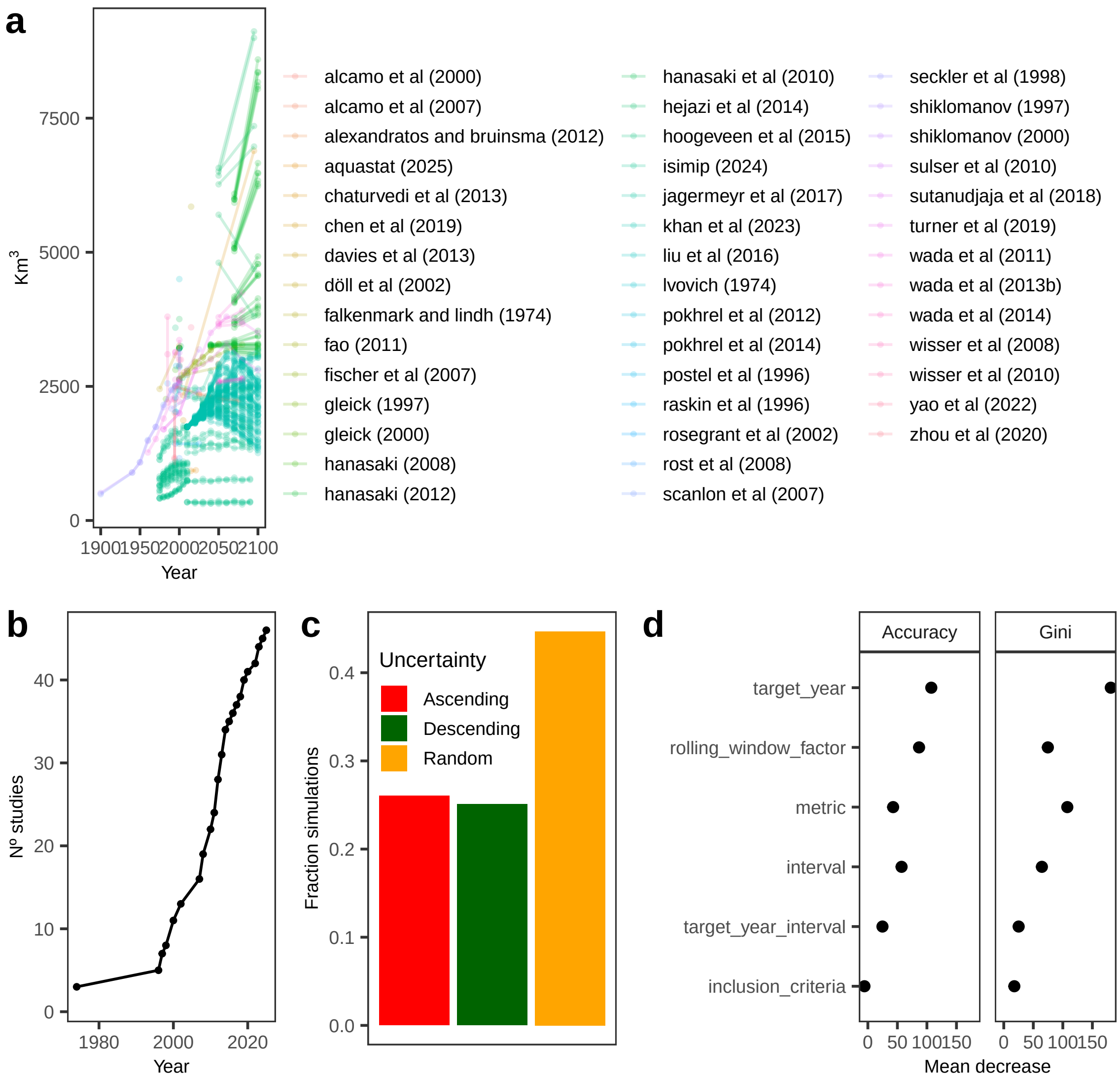
```
bottom <- plot_grid(cumulative.iww, plot.fraction, plot.rf, ncol = 3, labels = c("b", "c", "d"
          rel_widths = c(0.26, 0.3, 0.44))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

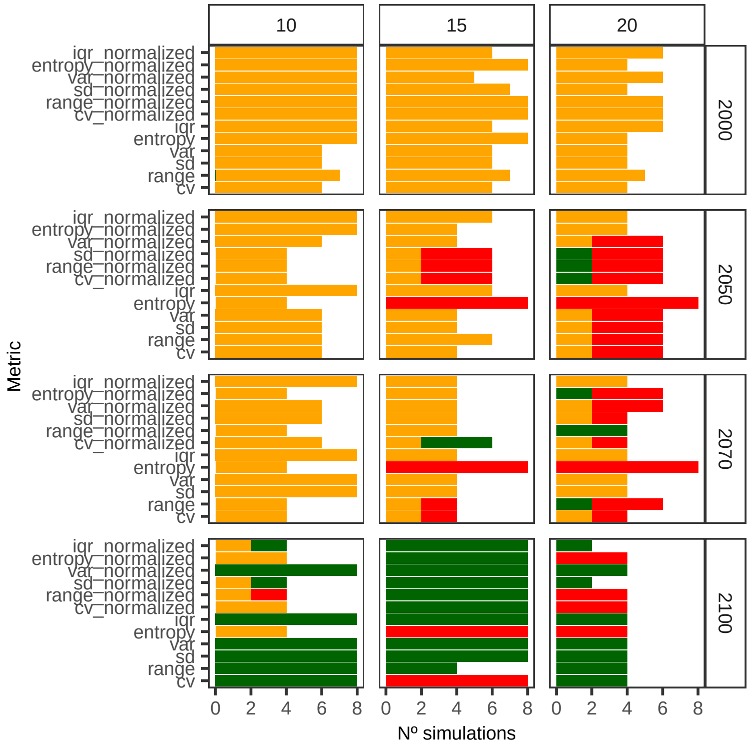
```
bottom
```




```
#
final.faceted.plot <- plot_grid(plot.iww, bottom, ncol = 1, labels = c("a", ""),
                                rel_heights = c(0.55, 0.45))

final.faceted.plot
```



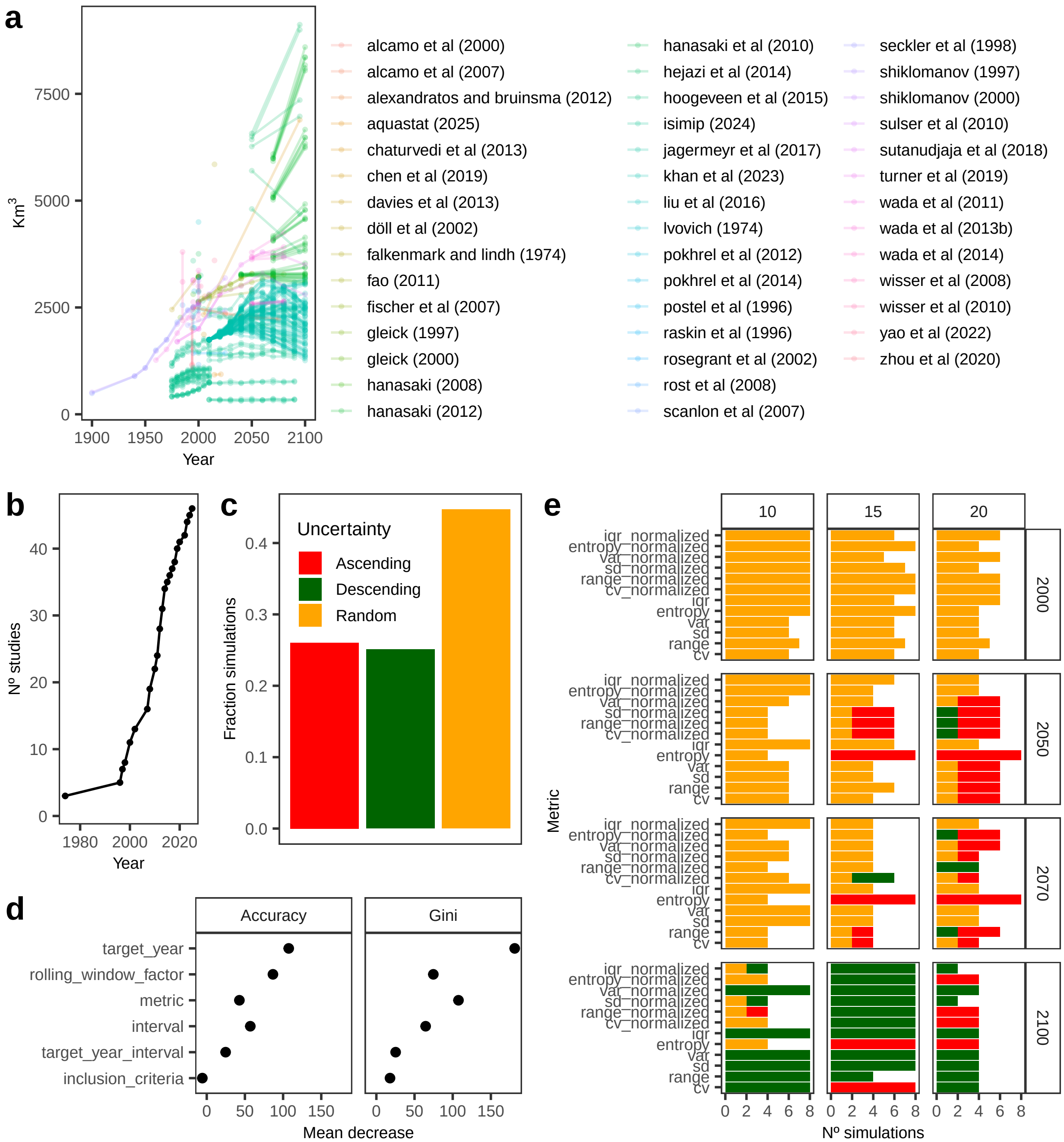

```
# RESULTS FACETED BY INTERVAL AND TARGET YEAR, X AXIS METRICS ###################

plot.faceted.metrics <- final.dt %>%
  ggplot(., aes(x = factor(metric), fill = trend)) +
  geom_bar(position ="identity") +
  facet_grid(target_year ~ interval, scales = "free_y") +
  scale_fill_manual(values = selected_colors, name = "Uncertainty") +
  theme_AP() +
  labs(x = "Metric", y = "Nº simulations") +
  theme(legend.position = "none") +
  coord_flip()

plot.faceted.metrics
```




```
bottom <- plot_grid(cumulative.iww, plot.fraction, ncol = 2, rel_widths = c(0.4, 0.6),
                    labels = c("b", "c"))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

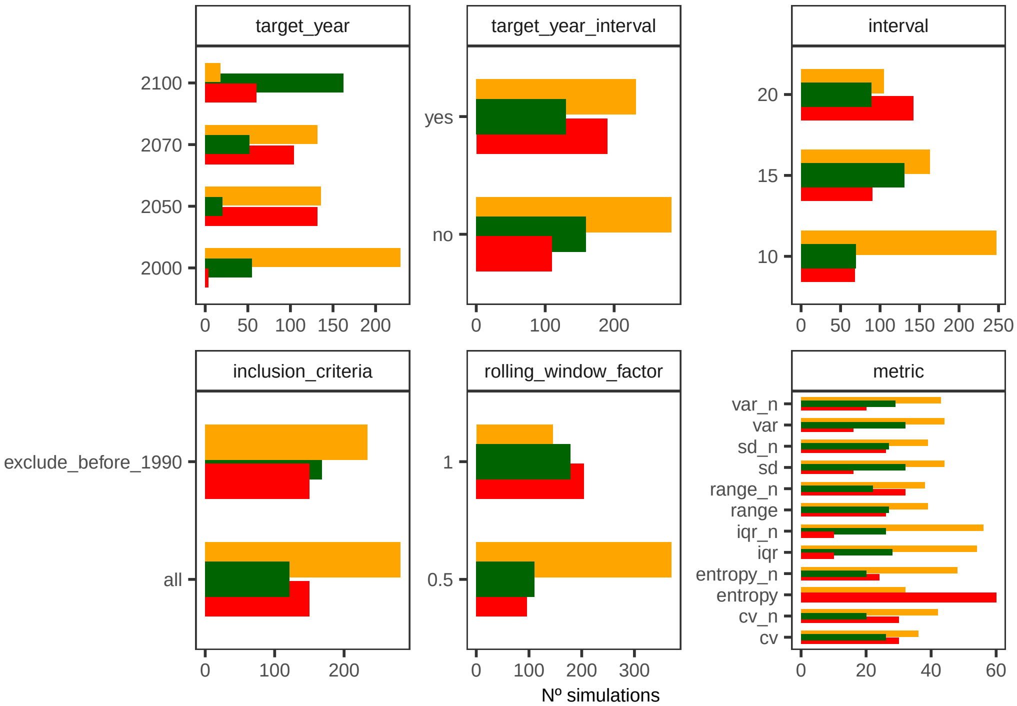
```
left <- plot_grid(bottom, plot.rf, ncol = 1, labels = c("", "d"), rel_heights = c(0.6, 0.4))
bottom2 <- plot_grid(left, plot.faceted.metrics, ncol = 2, labels = c("", "e"))
plot_grid(plot.iww, bottom2, rel_heights = c(0.42, 0.58), ncol = 1, labels = c("a", ""))
```




```
final.dt %>%
  melt(., measure.vars = c("target_year", "target_year_interval", "interval",
                           "inclusion_criteria", "rolling_window_factor", "metric")) %>%
  .[, .N, .(variable, value, trend)] %>%
  .[ , value := gsub("_normalized", "_n", value)] %>%
  ggplot(., aes(value, N, fill = trend)) +
  scale_fill_manual(values = selected_colors, name = "Uncertainty") +
  geom_bar(stat = "identity", position = position_dodge(0.5)) +
  facet_wrap(~variable, scale = "free") +
  labs(x = "", y = "Nº simulations") +
```

```r
  theme_AP() +
  coord_flip() +
  theme(legend.position = "none")
```

```
## Warning in melt.data.table(., measure.vars = c("target_year",
## "target_year_interval", : 'measure.vars' [target_year, target_year_interval,
## interval, inclusion_criteria, ...] are not all of the same type. By order of
## hierarchy, the molten data value column will be of type 'character'. All
## measure variables not of type 'character' will be coerced too. Check DETAILS in
## ?melt.data.table for more on coercion.
```



5 Session information

```r
# SESSION INFORMATION ##########################################################

sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.2.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/London
## tzcode source: internal
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] microbenchmark_1.5.0 lme4_1.1-35.5        Matrix_1.6-5
##  [4] here_1.0.1           doParallel_1.0.17    iterators_1.0.14
##  [7] foreach_1.5.2        rworldmap_1.3-8      sp_2.1-4
## [10] countrycode_1.6.0    ncdf4_1.23           scales_1.3.0
## [13] wesanderson_0.3.7    benchmarkme_1.0.8    cowplot_1.1.3
## [16] lubridate_1.9.3      forcats_1.0.0        stringr_1.5.1
## [19] dplyr_1.1.4          purrr_1.0.2          readr_2.1.5
## [22] tidyr_1.3.1          tibble_3.2.1         ggplot2_3.5.1
## [25] tidyverse_2.0.0      data.table_1.16.2    openxlsx_4.2.7.1
##
## loaded via a namespace (and not attached):
##  [1] dotCall64_1.2         benchmarkmeData_1.0.4 gtable_0.3.6
##  [4] spam_2.11-0           xfun_0.49             raster_3.6-30
##  [7] lattice_0.22-6        tzdb_0.4.0            Rdpack_2.6.2
## [10] vctrs_0.6.5           tools_4.3.3           generics_0.1.3
## [13] fansi_1.0.6           pkgconfig_2.0.3       lifecycle_1.0.4
## [16] compiler_4.3.3        fields_16.3           munsell_0.5.1
## [19] terra_1.7-78          codetools_0.2-20      htmltools_0.5.8.1
## [22] maps_3.4.2.1          yaml_2.3.10           nloptr_2.1.1
## [25] pillar_1.9.0          MASS_7.3-60.0.1       boot_1.3-31
## [28] nlme_3.1-166          tidyselect_1.2.1      zip_2.3.1
## [31] digest_0.6.37         stringi_1.8.4         splines_4.3.3
## [34] sensobol_1.1.5        rprojroot_2.0.4       fastmap_1.2.0
```

```
## [37] grid_4.3.3           colorspace_2.1-1      cli_3.6.3
## [40] magrittr_2.0.3       utf8_1.2.4            withr_3.0.2
## [43] timechange_0.3.0     rmarkdown_2.29        httr_1.4.7
## [46] hms_1.1.3            evaluate_1.0.1        knitr_1.49
## [49] rbibutils_2.3        viridisLite_0.4.2     rlang_1.1.4
## [52] Rcpp_1.0.13-1        glue_1.8.0            minqa_1.2.8
## [55] rstudioapi_0.17.1    R6_2.5.1
```

```r
## Return the machine CPU -----------------------------------------------------

cat("Machine:     "); print(get_cpu()$model_name)
```

```
## Machine:
```

```
## [1] "Apple M1 Max"
```

```r
## Return number of true cores -------------------------------------------------

cat("Num cores:   "); print(detectCores(logical = FALSE))
```

```
## Num cores:
```

```
## [1] 10
```

```r
## Return number of threads ---------------------------------------------------

cat("Num threads: "); print(detectCores(logical = FALSE))
```

```
## Num threads:
```

```
## [1] 10
```