Uncertainties in the estimation of global irrigation water withdrawals

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"))

# 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.margin = margin(0.5, 0.1, 0.1, 0.1),
          legend.box.margin = margin(0.2,-4,-7,-7),
          plot.margin = margin(3, 4, 0, 4),
          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.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),
          strip.text.x = 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

```
# 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)
```

```
# 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"
```

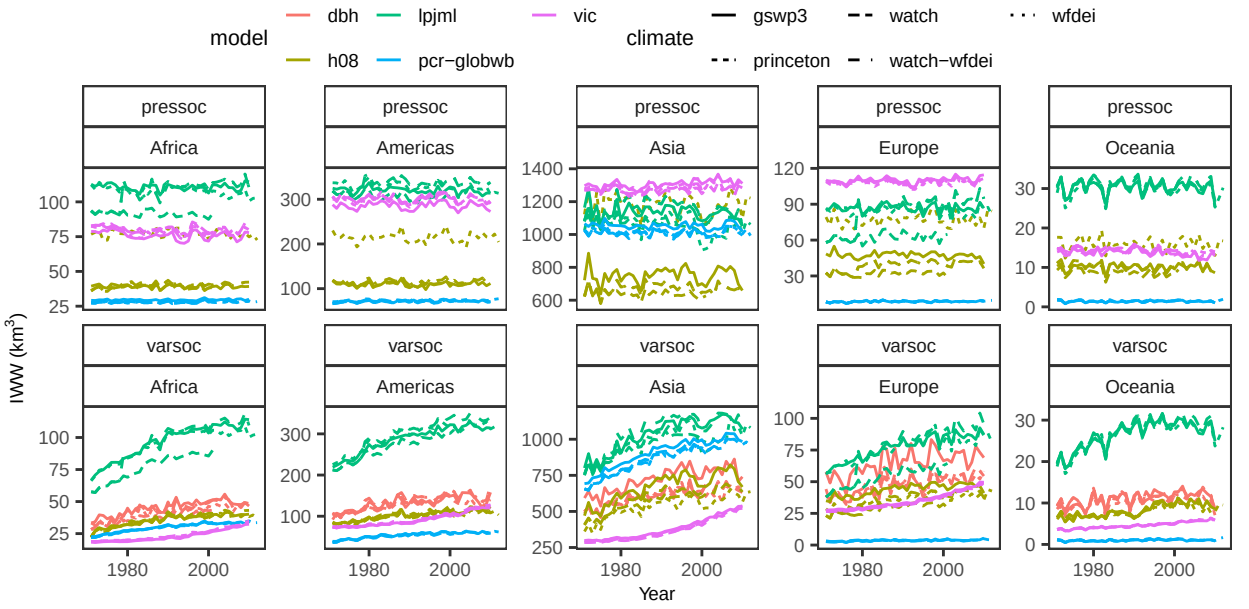
```
##  [8] "h08_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012.nc4"
##  [9] "h08_princeton_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2012(1).nc4"
## [10] "h08_princeton_nobc_hist_varsoc_co2_airrww_global_monthly_1971_2012.nc4"
## [11] "h08_watch_nobc_hist_pressoc_co2_airrww_global_monthly_1971_2001.nc4"
## [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")
```

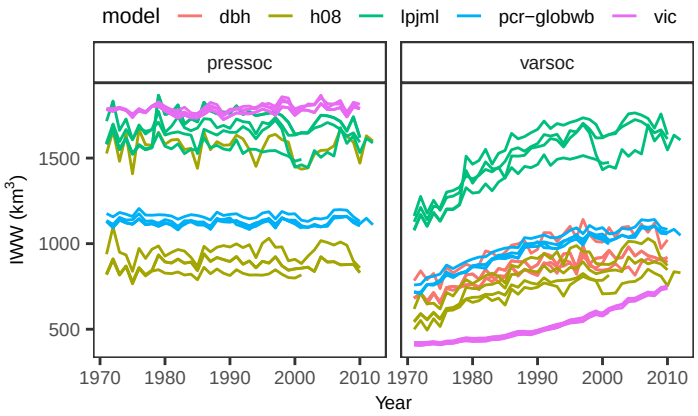
```
# Pressoc: constant human impacts in the form of dams and reservoirs
# varsoc: variable human impacts.
```

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

```
# 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)
```

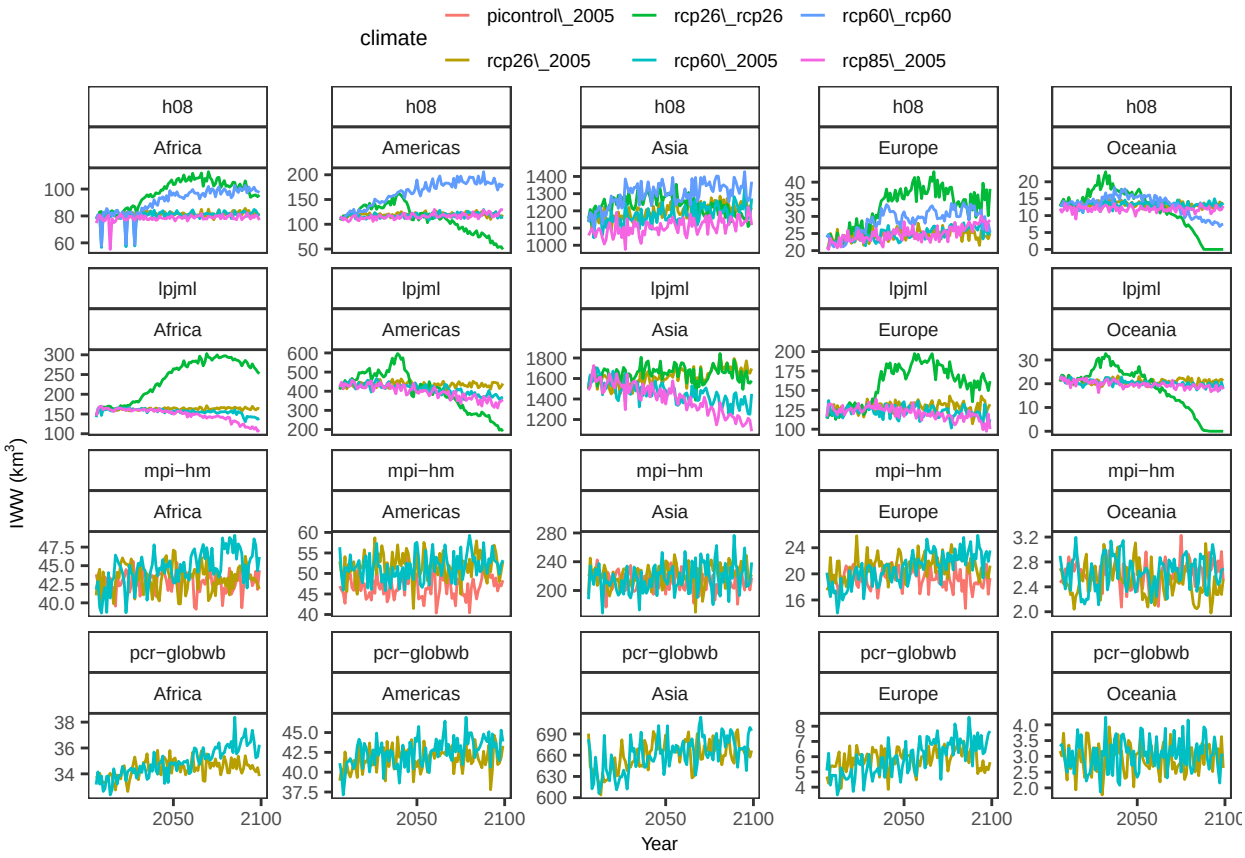
```r
# ARRANGE DATA ##################################################################

# 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"
```

```r
# 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")
```

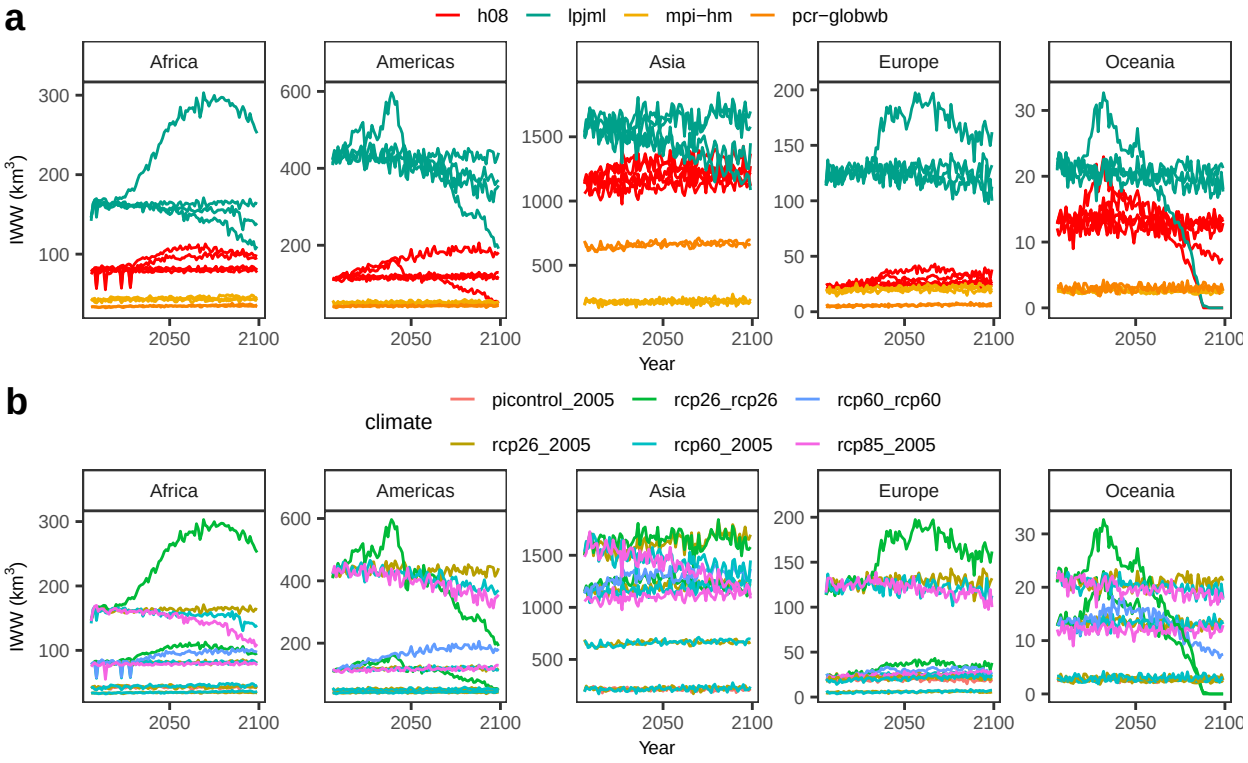
```r
# PLOT ISIMIP ###################################################################

# 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")

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

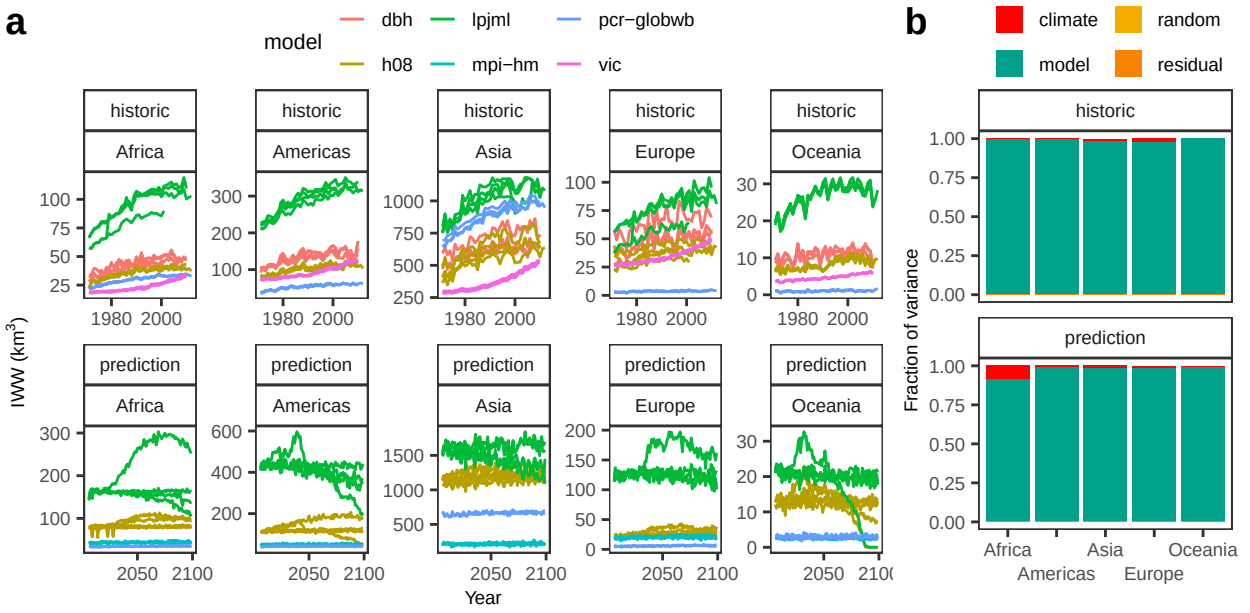


2.3 ANOVA

```
# ANOVA ###################################################################

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

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

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

```r
# Linear mixed-effects model with random intercept for year ------------------

analysis_variance_fun <- function(dt) {

  model_lmm <- lmer(estimation ~ model + climate + (1 | year), data = dt)

  anova_res <- anova(model_lmm)

  # Extract variance components ----------------------------------------------

  var_comp <- as.data.frame(VarCorr(model_lmm))
  residual_var <- attr(VarCorr(model_lmm), "sc")^2  # Residual variance
  random_var <- var_comp$vcov[var_comp$grp == "year"]  # Year random effect variance

  # Calculate total variance -------------------------------------------------

  fixed_effects_var <- sum(anova_res$`Sum Sq`)  # Sum of squares for all fixed effects
  total_var <- fixed_effects_var + random_var + residual_var

  # Calculate proportion of variance explained -------------------------------

  climate_var <- anova_res["climate", "Sum Sq"] / total_var  # Proportion due to climate
  model_var <- anova_res["model", "Sum Sq"] / total_var  # Proportion due to model
  residual_proportion <- residual_var / total_var  # Proportion of residual variance
  random_effect_proportion <- random_var / total_var  # Proportion of variance due to random e

  # Return results ----------------------------------------------------------

  output <- data.table(
    climate_variance = climate_var,
    model_variance = model_var,
    random_variance = random_effect_proportion,
    residual_variance = residual_proportion
  )

  return(output)

}

# RUN MODEL ###################################################################

results.dt <- isimip.full[, analysis_variance_fun(.SD), .(Continent, context)]

# PLOT RESULTS ################################################################

a <- isimip.full[, .(estimation = sum(V1)), .(model, Continent, climate, year, context)] %>%
  ggplot(., aes(year, estimation, color = model, group = interaction(climate, model))) +
```

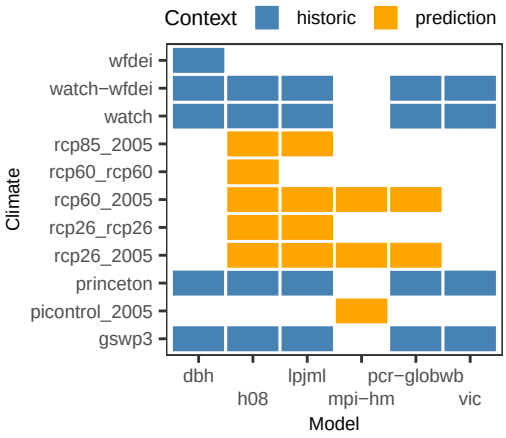
```
  geom_line() +
  facet_wrap(context~Continent, scale = "free", ncol = 5) +
  scale_x_continuous(breaks = breaks_pretty(n = 3)) +
  theme_AP() +
  guides(colour = guide_legend(nrow = 2)) +
  labs(x = "Year", y = bquote("IWW (km"^3 * ")"))  +
  theme(legend.position = "top")

b <- melt(results.dt, measure.vars = paste(c("climate", "model", "random", "residual"),
                                           "variance", sep = "_")) %>%
  .[, variance:= tstrsplit(variable, "_", fixed = TRUE)[[1]]] %>%
  ggplot(., aes(Continent, value, fill = variance)) +
  geom_bar(stat = "identity") +
  facet_wrap(~context, ncol = 1) +
  labs(x = "", y = "Fraction of variance") +
  scale_fill_manual(name = "", values= wes_palette(selected.palette, n = 4)) +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  guides(fill = guide_legend(nrow = 2)) +
  theme_AP() +
  theme(legend.position = "top")

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)

                                    df[, Continent := countrycode(Country, origin = "country.name

                                    df[, Dataset := list.of.files[i]]

                                    df
                                  }

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

stopCluster(cl)
```

```
# 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")
```

```
# 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
```

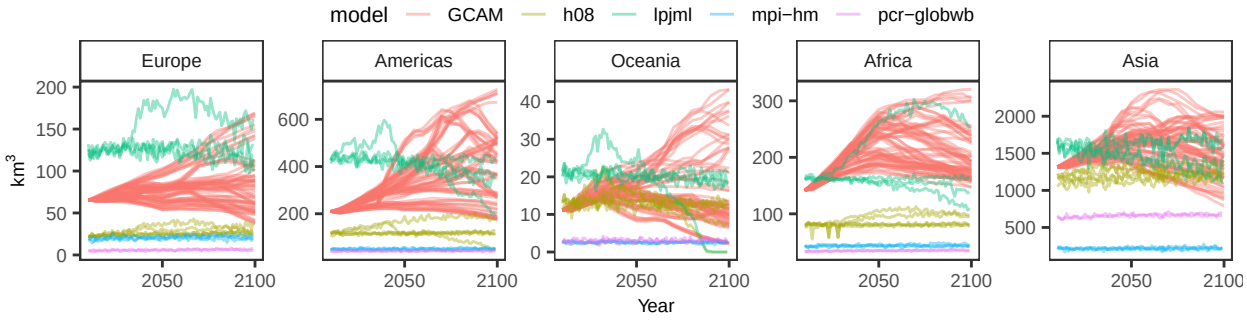
```
# PLOT ###############################################################################

# 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, estimati
  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
```