Package: CropWat (via r-universe)

May 14, 2026

Type Package

Title R Implementation of the FAO CropWat Model

Version 0.2.2

Description This package implements the functions describing water balance on an irrigated crop as described in FAO publications of the Irrigation and Drainage Series, namely, No. 56 ` `Crop Evapotranspiration - Guidelines for computing crop water requirements" and No. 33 titled "Yield response to water".

URL https://inrae.r-universe.dev/CropWat, https://umr-g-eau.pages-forge.inrae.fr/cropwat/

BugReports https://forge.inrae.fr/umr-g-eau/cropwat/-/issues

License AGPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports dplyr, ggplot2, lubridate, readr, rlang, stats, tibble, tidyr

Depends R (>= 2.10)

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev libx11-dev

Repository https://inrae.r-universe.dev

Date/Publication 2026-03-14 18:36:32 UTC

RemoteUrl https://forge.inrae.fr/umr-g-eau/cropwat

RemoteRef main

RemoteSha 60da161ea9795fa524237450fedbb8186d56e58f

Contents

`calc_isCycle` *Compute cycle period extend during the year*

Description

Compute cycle period extend during the year

Usage

```
calc_isCycle(cp, DatesR, sowing_date = cp$sowing_date)
```

Arguments

`cp`	Crop parameters (See get_crop_params)
`DatesR`	A vector of continuous lubridate::Date If `NULL`, the calculation is limited to the crop cycle of the plant
`sowing_date`	Sowing date in format "MM-DD"

Value

A vector of logical of the crop cycle for each day of the year.

Examples

```
DatesR <- seq(as.Date("2010-03-01"), as.Date("2010-10-31"), by = "1 day")
isCycle <- calc_isCycle(get_crop_params("SB2023-soja"), DatesR)
plot(isCycle)
```

calc_Kc *Compute crop parameters time series from crop parameters pivot points*

Description

These functions compute the following parameters:

Usage

```
calc_Kc(cp, DatesR = NULL, sowing_date = cp$sowing_date)

calc_RAW(TAW, p)

calc_TAW(
  cp,
  AWC,
  soil_depth,
  DatesR = NULL,
  sowing_date = cp$sowing_date,
  root_depth = calc_root_depth(cp, soil_depth, DatesR, sowing_date)
)

calc_p(cp, DatesR = NULL, sowing_date = cp$sowing_date)

calc_root_depth(cp, soil_depth, DatesR = NULL, sowing_date = cp$sowing_date)
```

Arguments

cp	Crop parameters (See get_crop_params)
DatesR	A vector of continuous lubridate::Date If NULL, the calculation is limited to the crop cycle of the plant
sowing_date	Sowing date in format "MM-DD"
TAW	Total Available Water (mm). See calc_TAW
p	fraction of Readily Available Water time series. See calc_p
AWC	Available Water Capacity (mm)
soil_depth	Soil depth (m)
root_depth	Root depth time series (m). See calc_root_depth

Details

- Calc_Kc: crop coefficient K_c
- calc_p: critical depletion fraction $p=RAW/TAW$
- calc_root_depth: root depth in m

- calc_RAW: Readily Available Water (RAW) in mm
- calc_TAW: Total Available Water (TAW) in mm

For calc_TAW, parameters cp, year, and sowing_date are useless if root_depth is provided.

Value

A vector of the parameter for each day of the crop cycle or the period defined by DatesR.

Examples

```
# Compute Kc for the crop cycle
Kc <- calc_Kc(get_crop_params("SB2023-soja"))
plot(Kc, type = "l")

# Compute Kc for a given period (less than one year)
# with default sowing date defined in crop parameters
DatesR <- seq(as.Date("2010-03-01"), as.Date("2010-10-31"), by = "1 day")
Kc <- calc_Kc(get_crop_params("SB2023-soja"), DatesR = DatesR)
plot(Kc, type = "l")

# Compute Kc for a given period with user defined sowing date
Kc <- calc_Kc(get_crop_params("FAO-MAIZE"),
              DatesR = DatesR,
              sowing_date = "05-01")
plot(Kc, type = "l")
```

CW_create_input *Create CropWat model input*

Description

Create CropWat model input

Usage

```
CW_create_input(
  crop,
  DatesR,
  ETo,
  P,
  soil_depth,
  AWC,
  cp = get_crop_params(crop),
  sowing_date = cp$sowing_date
)
```

Arguments

crop	The code of the crop
DatesR	Simulation period (vector of lubridate::Date)
ETo	Potential Evaporation (mm/day)
P	Precipitation (mm/day)
soil_depth	Soil depth (m)
AWC	Available Water Capacity (mm)
cp	Crop parameters (See get_crop_params)
sowing_date	Sowing date in format "MM-DD"

Details

The crop cycle begins at the first matching sowing date in the simulation period. So, first time steps of simulation if are not in a crop cycle even if they occur during the crop cycle.

Value

A tibble::tibble containing the input times series.

Examples

```
# Import example climate dataset
data(ZH_3_clim)
str(ZH_3_clim)
# Selecting years 2010 and 2011
meteo <-
 ZH_3_clim[ZH_3_clim$Date >= as.Date("2010-01-01") & ZH_3_clim$Date <= as.Date("2011-12-31"), ]
# Create model input
cw_input <- CW_create_input("SB2023-soja",
                            DatesR = meteo$Date,
                            ETo = meteo$ETP,
                            P = meteo$Ptot,
                            soil_depth = 1.2,
                            AWC = 140)
str(cw_input)

plot(cw_input)
```

CW_create_state *CropWat model state initiation*

Description

CropWat model state initiation

Usage

```
CW_create_state(
  Dr = 0,
  cw_input = NULL,
  iTS = 1,
  isCycle = cw_input$isCycle[iTS],
  TAW = cw_input$TAW[iTS]
)
```

Arguments

Dr	Initial root zone depletion (-)
cw_input	Model input
iTS	Time step to pick up in cw_input for defining isCycle and TAW
isCycle	logical defining if the time step is in the crop cycle
TAW	Total Available Water (mm)

Value

A list with items:

- Dr: Root zone depletion (-)
- isCycle: logical indicating if the state stands during the crop cycle
- TAW: Total Available Water (m)

Examples

```
example(CW_create_input, "CropWat")

X <- CW_create_state(cw_input = cw_input)
str(X)
```

CW_data *Retrieve the path of internal package data*

Description

Retrieve the path of internal package data

Usage

```
CW_data(file)
```

Arguments

file	File name

Value

The path of the file stored in the package dataset

Examples

```
CW_data("crop_params.tsv")
```

`CW_irrig_0` *Irrigation management functions*

Description

- `CW_irrig_0` simulates the absence of irrigation
- `CW_irrig_Dr` simulates an irrigation system that fills the soil's moisture depletion on a daily basis

Usage

```
CW_irrig_0(X, input, Dr)

CW_irrig_Dr(X, input, Dr)
```

Arguments

`X`	Model state at previous time step
`input`	Model input for the current time step
`Dr`	Current soil moisture depletion

Value

The amount of the daily irrigation (m)

Examples

```
CW_irrig_0()

example(CW_create_state, "CropWat")
CW_irrig_Dr(X, cw_input[1, ], 0.1)
CW_irrig_Dr(X, cw_input[100, ], 0.1)
```

`CW_irrig_fun_factory` *Irrigation management function factory*

Description

This function is a helper for generating irrigation management based on thresholds calculated from fixed ratio of RAW and/or TAW.

Usage

```
CW_irrig_fun_factory(
  RAW_ratio,
  TAW_ratio = 0,
  apply_Dr = FALSE,
  dates_irrig = NULL
)
```

Arguments

`RAW_ratio`	Ratio of RAW
`TAW_ratio`	Ratio of TAW
`apply_Dr`	When the threshold is reached, does the irrigation dose should fills the soil's moisture depletion?
`dates_irrig`	vector of lubridate::Date when irrigation occurs (By default, irrigation is applied during the crop cycle)

Value

A function which computes the irrigation dose for the current day.

Examples

```
# An irrigation management function which triggers an irrigation when
# Dr < 0.5 * RAW with a dose filling until 0.5 * RAW
fun_irrig_half_RAW <- CW_irrig_fun_factory(0.5)

fun_irrig_half_RAW

str(as.list(environment(fun_irrig_half_RAW)))
```

CW_model *Run CropWat model for one time step*

Description

Run CropWat model for one time step

Usage

```
CW_model(X, input, FUN_IRRIG = CW_irrig_Dr, is_irrigable_day = TRUE)
```

Arguments

X	Initial state (See CW_create_state)
input	Model input (See CW_create_input)
FUN_IRRIG	Irrigation management function
is_irrigable_day	logical TRUE if an irrigation is applicable for this time step

Value

A named numeric vector of class "CW_model" with items:

- Ks: Water stress coefficient (0-1, where 1 = no stress)
- ETc_adj: Adjusted crop evapotranspiration (mm/day)
- Dr: Root zone depletion at the end of the time step (mm)
- Ir: Irrigation dose for the current day (mm)

The returned value has an attribute "state" which contains the final state of the model at the end of the day (See CW_create_state).

Examples

```
data(ZH_3_clim)
meteo <-
 ZH_3_clim[ZH_3_clim$Date >= as.Date("2010-01-01") & ZH_3_clim$Date <= as.Date("2010-12-31"), ]
cw_input <- CW_create_input("SB2023-soja",
                            DatesR = meteo$Date,
                            ETo = meteo$ETP,
                            P = meteo$Ptot,
                            soil_depth = 1.2,
                            AWC = 140)
X0 <- CW_create_state(cw_input = cw_input)
X1 <- CW_model(X0, cw_input[1, ])
str(X1)
```

CW_run_simulation *Run a simulation with the CropWat model*

Description

Run a simulation with the CropWat model

Usage

```
CW_run_simulation(X, cw_input, FUN_IRRIG = CW_irrig_Dr)
```

Arguments

X	Initial state (See CW_create_state)
cw_input	Model input (See CW_create_input)
FUN_IRRIG	Irrigation management function

Value

A tibble::tibble with

Examples

```
example("CW_create_state", "CropWat")
cw_output <- CW_run_simulation(X, cw_input)
str(cw_output)

plot(cw_output)
```

load_crop_params *Crop parameters*

Description

- load_crop_params: load the parameters for all the crops
- get_crop_params: returns the parameters of a given crop

Usage

```
load_crop_params(file = CW_data("crop_params.tsv"))

get_crop_params(crop, crop_params = load_crop_params())
```

Arguments

file	Path of the table of crop parameters
crop	The code of the crop
crop_params	data.frame of crop parameters (See load_crop_params)

Details

The file to read should have the following columns and format:

- crop: crop code
- Lini: Length of initial stage of crop development
- Ldev: Length of crop development stage
- Lmid: Length of mid season
- Lend: Length of late season
- Kini: Initial Kc at the beginning of the crop cycle during Lini days (-)
- Kmax: Maximum Kc during Lmid day of the cycle (-)
- Kend: Final Kc at the end of the cycle (-)
- Zini: Initial root depth at the beginning of the crop cycle (m)
- Zend: Final root depth at the end of the crop cycle (m)
- p_ini: Initial fraction of readily available water (-)
- p_mid: Fraction of readily available water at Lmid day of the cycle (-)
- p_end: Final Initial fraction of readily available water at Lend day of the cycle (-)
- Yini: Yield response factor (Ky, not used yet)
- Ydev: Yield response factor (Ky, not used yet)
- Ymid: Yield response factor (Ky, not used yet)
- Yend: Yield response factor (Ky, not used yet)
- Ytot: Yield response factor (Ky, not used yet)
- h: crop height (m) (not used)
- sowing_date: Sowing date in format "MM-DD"

The default crop parameters provided in the package come from a variety of sources.

Crop names starting with "KURN-" or "FAO-" are imported from the CRO files provided with the CropWat 8.0 installation. "KURN-" are crop parameters calibrated at "Kurnool, Andhra Pradesh, India".

Crop names starting with "SB2023-" are crop parameters coming from a calibration performed on outputs of the AqYield model for the Aveyron catchment (Soutif-Bellenger, 2023).

Reference:

Soutif-Bellenger, Myriam, Guillaume Thirel, Olivier Therond, et Jean Villerd. 2023. As Simple as Possible but Not Simpler?: The Case of Irrigation Modeling at Catchment Scale in Southwestern France. Irrigation Science 41 (5): 713-36.

Value

A tibble::tibble, see details for the column formats.

Examples

```
# Get available crops in the package
cp <- load_crop_params()
cp$crop

str(cp)

# Get parameters for sugar beet from FAO
get_crop_params("FAO-SUGARBET")
```

---

load_CW8_CRO *Load CropWat v8.0 crop parameter CRO file*

Description

Load CropWat v8.0 crop parameter CRO file

Usage

```
load_CW8_CRO(file)
```

Arguments

file	Path of the CRO file

Value

Crop parameters formatted as

Examples

```
load_CW8_CRO(system.file("extdata/TOMATO.CRO", package = "CropWat"))
```

plot.CW_TS *Plot times series of CropWater inputs and outputs*

Description

See CW_create_input and CW_run_simulation for examples.

Usage

```
## S3 method for class 'CW_TS'
plot(x, ...)
```

Arguments

x	Either CW_input or CW_output
...	Not used

Value

A ggplot object

ZH_3_clim *Climate example dataset*

Description

A data.frame with 3 columns:

- Date: the date
- Prtot: precipitation (mm/day)
- ETP: Potential evaporation (mm/day)

Usage

```
ZH_3_clim
```

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 2922 rows and 3 columns.

Index