

CRHMr Cheatsheet

This document is intended to be a quick guide to using some of the most common CRHMr functions. The functions are grouped by their use in a typical workflow. You should consult the manual **CRHMr.pdf** to get the instructions for each of the functions mentioned here.

Note that all plotting functions return **ggplot2** plot objects, which you can modify.

CRHMr dataframe

All CRHMr functions are based on a consistent dataframe for holding the model observations (obs) and outputs. The first column must be called 'datetime' and contains the date and time in POSIXct format. All column references are with respect to the datetime. So, the first variable is the second column in the dataframe.

Workflow

The typical sequence of operations is

1. Create an obs file. You can create a file from scratch, or read in existing data.
2. Correct the obs data. Remove bad data, fill in gaps, change the data format. Write the obs data to a file.
3. Run the CRHM model. Set the model file for automatic execution, set the parameter values. Execute the model, and capture the output.
4. Post-processing. Read in the model output, massage data, plot graphs.

1. Obs file creation

Function	Action
createObsDataframe	creates a new, empty, dataframe
readObsFile	reads an existing obs file
readCampbell	reads data from a Campbell scientific datalogger file
readClassFile	reads data from a CLASS model file
appendObs	joins two obs dataframes
assembleObs	combines variables from separate obs dataframes

There are also functions for reading obs values from WISKI databases, MSC data, and reanalysis datasets in the WISKIr, MSCr, and Reanalysis packages, respectively.

To plot the values in an obs dataframe, which is useful for seeing bad values, use the command **plotObs**. You can summarise all of the obs files in a directory, optionally plotting their values, with the command **summarizeObsFiles**.

2. Obs correction

There are three types of CRHMr functions for fixing R data:

1. functions that find bad values,
2. functions that allow you to remove or correct bad values, and
3. functions that infill missing values.

finding bad data

Function	Action
findGaps	finds missing values
findDuples	finds duplicate datetimes
findSpikes	finds positive or negative spikes

removing bad data

Function	Action
minObs	checks obs values for minimum threshold
maxObs	checks obs values for maximum threshold
deDupe	removes records with duplicate datetimes
deleteSpikes	removes data spikes

infilling missing values

Function	Action
makeRegular	makes datetime values fit their time step exactly
insertMissing	inserts rows where values are missing
interpolate	fills missing values using linear or spline interpolation
regress	calculates linear regressions between set of values
impute	infills missing values with values from another dataset, using regression constants
tMinMaxToHourly	interpolates daily min, mean and max air temperatures to hourly values

You should never use the **impute** or **interpolate** functions with RH data as the values are strongly affected by air temperature. Instead, use the function **changeRHtoEa** to convert the values to vapour pressures, which can *then* be imputed or interpolated. CRHM can use the vapor pressure values directly, or you can convert them to RH values with the command **changeEatoRH**.

There are also many functions for fixing data from weighing precipitation data. They are listed in the CRHMr manual under **weighing-Gauge methods**.

writing the obs file

Before saving your data to an obs file, you should first use the function **trimObs** to remove values before 01:00 on the first day of complete data and after 00:00 on the last day.

To write the obs file, use the function **writeObsFile**.

3. Running CRHM

Running CRHM through CRHMr allows you to run your model repeatedly, making changes to the inputs and/or parameters.

To run a CRHM model through CRHMr, you should first use the function **automatePrj** to set the .prj file to execute automatically.

You can then use these functions to set values in the .prj file:

Function	Action
setPrjBasinName	sets the name of the basin
setPrjDates	sets the run start and end dates
setPrjHRUnames	sets the names of the HRUs
setPrjOutputVariables	sets variables to be output
setPrjParameters	sets parameter values
setPrjRunID	sets RunID value

Once the .prj file values have been set, you can run CRHM directly from R using the function **runCRHM**.

The output will be stored in a file specified in the command

4. Post-processing

If you have run the CRHM model using runCRHM, read in the output file with the function **readOutputFile**.

If you have run the CRHM model from the command line, instead of from CRHMr, use the function **readExportFile** to read a file that you have exported manually. The reason for the two commands is that manually exporting the data produces several different file formats.

Once you have read in the CRHM output, you can use these functions:

Function	Action
aggDataframe	produces daily, monthly or yearly aggregated values
yearlyPeaks	finds annual max and/or min values
hydrograph	plots hydrograph of CRHM output and/or WSC gauged flows
cumulativeDischargePlot	plots annual cumulative flows
monthlyQQplot	produces monthly quantile-quantile plots of variables
monthlyPrecipTotals	calculates mean monthly precipitation
simpleDailyWater	calculates daily values of all water storages and fluxes
cumulDailyWater	accumulates daily values of all water storages and fluxes
simpleRibbonPlot	produces ribbon plots of daily min, max and mean values
hruGroupWaterSummary	calculates daily water storages and fluxes for grouped HRUs

Of course, you can use all standard **R** plotting analysis and plotting functions with the CRHMr obs and output dataframes.

