htsr Cheat Sheet

R library dependencies:

beepr	DBI	editData	openair	readxl	RSQLite	tidyverse	WriteXLS
data.table	directlabels	lubridate	raster	RODBC	shiny	tools	zoo

Conventions:

Prefixes of function names:	d_, database f_ file h_, hydromet- p_ ,plotting	s_, shiny z_, miscellaneous; u_, internal (with no help infos)
Suffixes of function names:	_app, application functions with questions / responses required in-line	
Prefixes of hts files:	cd_, generated by h_condition co_, generated by h_common cu_, generated by h_cumul eg_, generated by h_gaperr gf_, generated by h_gapfill gr_, generated by h_gaprem_itv na_, generated by h_nodata nw_, generated by f_change_id, pr_, generated by h_rainsnow,	re_, generated by h_replace, ro_, generated by h_rollav rs_, generated by h_restrict, sn_, generated by h_rainsnow su_, generated by h_substitute, sx_, generated by h_season (x=digit 2to4) ws_, generated by h_weightedsum xEtpyy_, generated by h_etp
Suffixes of hts files:	_xxxx, generated by h_timestep (x=digit) _C generated by h_month	_G, generated by h_month _M, generated by h_month
Prefixes of Excel files:	ad_, generated by h_month	cm_, generated by h_month

File formats;

.sqlite	SQLite data base with 38 tables.	

.hts	Rdata file containing a “tibble” object, tstab,
	with 4 columns : Date, Value, Station, Sensor
		`> tstab`
		`Date Value Station Sensor`
		`<dttm> <dbl> <fct> <fct>`
		`1 2015-10-03 12:00:00 2.17 CKS2500 IQ`

.calib	Rdata file containing two “tibble” objects, calibtab with 6 columns & dismtab with 6 columns	`> calibtab`
		`Id_Station Sensor Sen_Out Date H Q`
		`<fct> <fct> <fct> <fct> <dbl> <dbl>`
		`1 CKS2500 IH IQ 2015-10-03 16:45:00 -17 0.0387`
		`> dismtab`
		`Id_Station Sensor Date H Q Archive`
		`<chr> <chr> <chr> <dbl> <dbl> <int>`
		`1 CKS2500 IH 2015-10-03 16:45:00 27.5 1.91 1`

.gap	Rdata file containing a “tibble” object, with 5 columns	Used only with h_gapfill and f_properties

Data base functions

`d_compact(bd.sqlite)`	Compact htsr sqlite data base
`d_create(db.sqlite, cr_table = TRUE, bku = TRUE)`	Create a data base {d_imp_hydraccess}{d_imp_weewx}
`d_exp_hts(db.sqlite, sta, sen, rtime = FALSE, dstart, dend, rplot = FALSE)`	Extraction of a time-series from htsr data base. {s_exphts}
`d_exp_discalib(db.sqlite, sta, calib=TRUE, dism=TRUE)`	Export discharge measurements and calibrations from data base
`d_imp_hts(db.sqlite, file.hts, table, bku = TRUE)`	Import a hts file into a data base
`d_imp_hydraccess(db.sqlite, db.hydraccess)`	Import a full Hydraccess database into a new htsr sqlite database NB. Only works in Windows environment with a 32b R session.
`d_imp_weewx(db.sqlite, db.weewx, sta, name_st, tzo = "CET", bku = TRUE)`	Import a weewx data base into a htsr sqlite base
`d_inventory(db.sqlite, stalist = NA, form.out = NA)`	Inventory an htsr data base
`d_list(db.sqlite, sta = NA, form.out = NA)`	List stations and sensors of an htsr data base
`d_rem_hts(db.sqlite, table, sta, sen, start = NA, end = NA)`	Remove an htsr record from a data base

`d_sensor(db.sqlite, op = "C", sta, sen, table = NA, name_fld = NA, value_fld = NA, bku = TRUE)`	Create, Modify or Remove a sensor
`d_station(db.sqlite, op = "C", sta = NA, name_st = NA, name_fld = NA, value_fld = NA, bku = TRUE)`	Create, Modify or Remove a station
`d_table(db.sqlite, table, op = "C", bku = TRUE)`	Create or remove a table. {d_create}

File functions

`f_(file)`	Shortcut for file.choose(file).
`f_change_id(file, sta, sen, overwrite)`	Change Station id or Sensor id in a hts file.
`f_convert(f, form_start = "hts", form_end = "xlsx", ta = NA, sen = NA, output = NA, variable = NA)`	Convert an hts file in another format (xls, xlsx, csv or hdsm).
`f_properties(file, gaps = FALSE)`	Properties of a hts series. {h_gapfill}

Hydromet- functions

`h_common(files)`	Extract 2 (or more) time-series on their common period {h_weightedsum}{h_condition}{h_substitute}
`h_condition(files, condition)`	Conditional extraction of a time-series regarding another one
`h_cumul(file, start = NA, end = NA)`	Cumul of a time-series
`h_etp(method = c("Turc", "Penman-Monteith", "Priestley-Taylor", "Makkink", "Heargraves-Samani"), freq = c("day", "month"), f_temp, f_relh = NA, f_radg = NA, f_radn = NA, f_atmp = NA, f_wvel = NA, f_tmin = NA, f_tmax = NA, lat = NA, alt = NA, albedo = NA, z = NA)`	Compute the potential evapotranspiration with several methods {h_month}
`h_gaperr(file, nv = 1, itv0 = 43201, df)`	Replace errors with gaps in a time-series based on neighboring values {h_gaprem_itv}
`h_gapfill(file, npdt)`	Simple gapfilling in a time-series
`h_gaprem_itv(file, itv0 = 43201)`	Remove gaps in a time-series with a time interval threshold
`h_month(file, op="M", ba=NA, rmna=FALSE, climedit=FALSE, caledit_j=FALSE, caledit_m=FALSE, gapfill=FALSE)`	Monthly operations, based on a daily time-series
`h_nodata(file, threshold=NA, test="=", start=NA, end=NA)`	Replace values with NA conditionally or in a time interval
`h_rainsnow(fpr, fta, ta0,ta1,sta=NA)`	Share the solid and liquid precipitations with a temperature criteria
`h_rbind(files, sensor, gap = TRUE)`	Bind 2 time-series on consecutive periods {h_nodata}
`h_restrict(file, start=NA, end=NA)`	Restrict a series between 2 dates
`h_replace(file, old.val, new.val)`	Replace a value by another
`h_rollav (file, ti = 7, position = c("central", "right"))`	Rolling average of a daily time-series
`h_season(file, monthstart)`	Seasonal selection
`h_substitute(files)`	Substitute the missing values in a series by existing values of another series {f_properties}
`h_stat_basic(file)`	Basic statistics of a time-series
`h_timestep(file, tst, op = "M")`	Infra-daily fixed time step
`h_weightedsum(files, weights = NA, constant = 0)`	Weighted sum of time-series
`h_wind(db.sqlite, sta = NA, swd = NA, swv = NA)`	Create a wind table

Plotting functions

p_bar(nbst = nbst, filei, serlab, title, type, rnorm, rtime, start, end, rfixy, y.down = NA, y.up = NA, pal) **p_bar_app**(filename = NA, pset = TRUE, pfil = TRUE, rpal = 0,savefig = FALSE, width = 8, height = 6, fileo = "plot.png"	Bar plot
p_box_month(file, title = "Title", axeY = "Y-axis", savefig = FALSE, fileo = "plot.png", width = 8, height = 6)	Boxplot of the 12 months of a time-series.
p_clim(p_clim <- function (files, type="line", hydro.month=1, title="Title", yaxis="Value", y.down=NA, y.up=NA, rpal=FALSE, pal=mapalette, legend.l=NA))	Plot climatologies in hydrological year.
p_discalib(fcalib, sen, plotcalib= TRUE, plotdism=TRUE, title="Title", savefig=FALSE, width= 8, height= 6, fout="plot.png", limx =FALSE, limy = FALSE, xinf=NA, xsup=NA, yinf=NA, ysup=NA)	Plot calibration curves water levels vs discharges.
p_gaps(nbf, title = "Inventory", BW = FALSE, margin = 0.1)	Plot of data inventory
p_hypso(file_mnt, abbrev, prop = FALSE, range=50, fact=5, title="Title", savefig=FALSE, width= 8,height= 6, fileo="plot.png")	Plot the hypsometry curve of one or more basins
p_line(nbst, filei, serlab, title, type, rnorm, rtime, start, end, rfixy, y.down, y.up, pal = pal, linet, rppt, pointt, linew, smooth) **p_line_app** (filename = NA, pset = TRUE, pfil = TRUE, rpal = 0, smooth = FALSE, savefig = FALSE, width = 8, height = 6, fileo = "plot.png")	Line and/or point plot
p_scatter(files, intercept.zero = FALSE, remove.zero = FALSE, lg.axis = c(NA, NA),title = "Title")	Scatter plot of 2 or more time-series
p_wind(data_wind, ws.int = 0.5, angle = 45, grid.line = 10, type = "default", breaks = 5, offset = 5, paddle = FALSE)	Plot wind rose

Shiny functions

s_exphts()	Export hts files from a sqlite data base {d_exp_hts}
s_plot()	Plot hts files {p_line}{p_bar}

Miscellaneous functions

z_coord(ncoord = NA, ccoord = NA, type)	Coordinate utility {u_capteur}
z_set (tz = FALSE, mapal = FALSE, confp = FALSE, filep = FALSE, filelist = NA)	Edit settings {p_line_app}{p_bar_app}

Internal functions

NB. the main function is written in bold.

u_dbackup(db.sqlite)	Backup a data base **{d_create}**{_d_compact}{d_imp_hts}{d_imp_hydraccess}{d_rem_hts}{d_sensor}{d_station}{d_table}
u_fracjour(date) **u_ra**(fracjour, lat) **u_radn**(ra, alt, albedo, tmin, tmax, ea)	Internal functions used for ETP calculations **{h_etp}**
u_gaprem_mul(tstab)	In a tstab table, reduce a sequence of consecutive gap records in a single gap record **{h_gaprem_itv}**
u_merge(files)	Merge hts files **{h_common}**
u_newnomtable(nomtable)	Replace the table names imported from hydraccess data base **{d_imp_hydraccess}**

u_stacapt(db.dsqlite, table, sta, sen) **u_statnom**(db.sqlite, sta)	Used in d_exp_hts **{d_exp_hts}**
u_station(db.dsqlite, form.out) **u_capteur**(db.sqlite, sta, form.out)	Used in d_list **{d_list}**
u_test_robdc(db.hydraccess)	Test if the ODBC configuration is correct for importing hydraccess data base **{d_imp_hydraccess}**