Package: htsr (via r-universe)

November 2, 2024

Type Package

Title Hydro-Meteorology Time-Series

Version 2.1.7

Date 2024-12-31

Maintainer Pierre Chevallier <pierre.chevallier@mailo.com>

Description Functions for the management and treatment of hydrology and meteorology time-series stored in a 'Sqlite' data base.

License GPL-2

Depends R (>= 3.5.0)

Imports dplyr, ggplot2, purrr, readr, tibble, lubridate, stringr, fs, Rcpp, readxl, RSQLite, DBI, shiny, shinyFiles, waiter, WriteXLS

Suggests knitr, rmarkdown, RODBC, openair, editData, terra, directlabels

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

LinkingTo Rcpp

URL https://github.com/p-chevallier/htsr

BugReports https://github.com/p-chevallier/htsr/issues

Repository https://p-chevallier.r-universe.dev

RemoteUrl https://github.com/p-chevallier/htsr

RemoteRef HEAD

RemoteSha 0244becbb88bfb3ad96b7c9e96dd7ae49197bc10

Contents

ds_exp_hts 3
ds_inventory_sensor 4
ds_inventory_station 4

ds_sensor . . . 5
ds_station . . . 6
d_backup . . . 6
d_compact . . . 7
d_convert_eaufrance . . . 7
d_convert_hydraccess . . . 8
d_convert_meteofrance_d . . . 9
d_convert_meteofrance_h . . . 10
d_convert_weewx . . . 10
d_create . . . 11
d_exp_discalib . . . 12
d_imp_hts . . . 13
d_rem_hts . . . 13
d_table . . . 14
fc . . . 15
f_change_id . . . 15
f_convert . . . 16
f_csv_multivar . . . 17
f_month2day . . . 17
f_properties . . . 18
hs_tstep . . . 19
h_addna . . . 20
h_adjust . . . 20
h_avday . . . 21
h_changetz . . . 22
h_common . . . 22
h_condition . . . 23
h_cumul . . . 24
h_gaperr . . . 25
h_gapfill . . . 25
h_gaprem_itv . . . 26
h_nodata . . . 27
h_rainsnow . . . 28
h_rbind . . . 29
h_replace . . . 30
h_restrict . . . 30
h_rollav . . . 31
h_season . . . 32
h_stat_basic . . . 33
h_substitute . . . 33
h_weightedsum . . . 34
h_wl_di . . . 35
h_year . . . 36
ps_plothts . . . 37
p_box_month . . . 38
p_clim . . . 39
p_discalib . . . 40
p_gaps . . . 41

p_hypso 42
p_scatter 43
p_wind 44
u_index 45
u_timestep 46
w_atmp_alt 46
w_etp 47
w_spechum2relhum 49
w_temp_alt 50
z_coord 50

Index **52**

`ds_exp_hts` *Extraction of a time-series from htsr data base*

Description

The function display a web page allowing to extract a time-series in the "hts" format.

Usage

```
ds_exp_hts()
```

Details

Complete the requested information in the left panel, then press the submit button in order to extract the file. If you want to display the plot of the extracted file, choose "line" or "bar" and press the plot button.

When the subfunction "d_exp_hts(fsq, sta,sen,rtime=FALSE,dstart=NA,dend=NA, rplot=FALSE)" is used solely it returns a tibble tstab with 4 columns Date, Value, Station, Sensor. In this last subfunction fsq is the sqlite data base; sta, the station id, sen, the sensor id; rtime, dstart and dend define a time interval; rplot, the resulted plot.

Value

The function returns a file (nomfic) with the following name: <sensor.id>_<station.id>.hts

Author(s)

P. Chevallier - Oct 2017 - Sep 2023

ds_inventory_sensor *Inventory of a station sensors of an htsr data base*

Description

The function display a web page in order to produce an inventory of the sensors for a selected station in an htsr data base.

Usage

```
ds_inventory_sensor()
```

Details

Complete the requested information in the left panel, then press the submit button. When finished press "done".

If "Output format" is "none", the results are displayed on the screen, If it is "xlsx", or "csv" (, as separator) or "csv2" (; as separator), the corresponding file with a sensor list is written.

Value

A table with the inventory of sensors of a selected station in the data base.

Author(s)

P. Chevallier - Jan 2024

ds_inventory_station *Inventory of the stations of an htsr data base*

Description

The function display a web page in order to produce an inventory of the stations in an htsr data base.

Usage

```
ds_inventory_station()
```

Details

Complete the requested information in the left panel, then press the submit button. When finished press "done".

If "Output format" is "none", the results are displayed on the screen, If it is "xlsx", or "csv" (, as separator) or "csv2" (; as separator), the corresponding file with a station list is written.

Value

A table with the inventory of stations in the data base.

Author(s)

P. Chevallier - Dec 2023

`ds_sensor` *Create, Modify or Remove a sensor*

Description

Create, Modify or Remove a sensor.

Usage

```
ds_sensor()
```

Details

If operation is Create, the fields Station, Table and Sensor are compulsory and cannot be modified afterwards.

Allowed entries for table are: WL (water levels), DI (discharges), QU (Quality), PR (precipitations), WE (weather).

If op is Create or Modify, the following text fields can be completed optionally: Nature, Description, Comment.

If op is Remove, all data corresponding to the sensor of the selected station are removed.

The data base is automatically backuped before any operation.

Value

Sensor created, modified or removed from the data base

Author(s)

P. Chevallier - Feb 2018-Sep 2023

ds_station *Create, Modify or Remove a station*

Description

Create, Modify or Remove a station.

Usage

```
ds_station()
```

Details

If operation is Create, the fields Id_Station, Type_Station and Name (name_st) are compulsory. The field Name can be modified afterwards.

If op is Create or Modify the following fields can be completed optionnaly: Country, Zone, Subzone, Large basin, Basin, Small basin, River, Longitude, Latitude, Altitude, Basin area, Manager.

If op is Modify, station type and station id cannot be modified. The sensors and data corresponding to the station are conserved.

If op is Remove, all data and sensors of the station are removed.

Value

Station created, modified ou removed from the data base

Author(s)

P. Chevallier - Jan 2018 - Sep 2023

d_backup *Backup a data base*

Description

Back a htsr sqlite data base

Usage

```
d_backup(fsq)
```

Arguments

fsq	Full name of the data base

Value

A saved data base with extension .bak

Author(s)

P. Chevallier - Jan 2019 / Nov 2020

---

d_compact *Compact a data base*

Description

Compact htsr sqlite data base

Usage

```
d_compact(fsq)
```

Arguments

fsq	Full name of the data base

Value

New data base or overwritten data base. Note that the created data base is empty.

Author(s)

P. Chevallier - Jan 2019

---

d_convert_eaufrance *Convert eaufrance station files into a htsr sqlite base*

Description

Convert a eaufrance hydrological file into a htsr sqlite base. It regards the "basic" data file, which includes water level and discharge data. .

Usage

```
d_convert_eaufrance(eaufrance.dir, station.id, fsqname)
```

Arguments

eaufrance.dir	Full path of the folder were the eaufrance data base is extracted (character)
station.id	Id list of the stations to convert (character)
fsqname	Name of the returned sqlite data base without extension (character)

Details

The data base is build from selected stations in the "stations.tar" file available on the data.eaufrance web site : https://data.ofb.fr/catalogue/data-eaufrance/fre/catalog.search. This file must be first downloaded and extracted in the folder eaufrance.dir. For the extraction the R function untar() can be used.

Secondly, within the eaufrance.dir, the file stations/stations.csv give the full list of the available stations. One or more station ids must be chosen and included in the station.id list parameter. Another possibility is to consult the "Hydro Portail" (https://www.hydro.eaufrance.fr/rechercher/entites-hydrometriques) in order to select the station ids.

In the sqlite data base, the units of water level data is cm and of discharge data is m3/s.

Value

A sqlite database compatible with the htsr library.

Author(s)

P. Chevallier - Jul/Aug 2024

---

d_convert_hydraccess *Convert a full Hydraccess database into a new htsr sqlite database (Windows only)*

Description

Because the Hydraccess application only works into a Windows environment, this function cannot be applied on other platforms (Mas OS or Linux). Additionally, the R session must be configured in 32b (see the htsr-package vignette).

Usage

```
d_convert_hydraccess(fsq, db.hydraccess)
```

Arguments

fsq	Full name of the sqlite data base
db.hydraccess	Full name of the hydraccess data base

Details

If the specified sqlite data base already exists, a confirmation is requested to overwrite it.

An 32b ODBC Microsoft driver must be configured in the "administrative tools" and installed for the hydraccess data base. The correct functioning can be verified using the sub-function u_test_rodbc (db.hydraccess), which must be successful.

Value

A new or a replaced sqlite htsr data base.

Author(s)

P. Chevallier - Nov 2018-Nov 2020

See Also

ds_inventory_station and ds_inventory_sensor for displaying the content of the sqlite data base; ds_exp_hts for extracting a time-series.

Examples

```
## Not run:

d_import_hydraccess("foo.sqlite","foo.mdb")

## End(Not run)
```

---

d_convert_meteofrance_d

Convert a Meteo-France csv daily basic data file into a htsr sqlite base

---

Description

Convert a Meteo-France csv daily data file into a htsr sqlite base. It regards the "basic" data file, which includes precipitation, temperature and wind data. For other variables the function d_convert_meteofrance_d1 shall be used with the corresponding csv file. The csv file shall be downloaded from https://meteo.data.gouv.fr/ The name of the created sqlite file is the same as the csv file with an extension .sqlite.

Usage

```
d_convert_meteofrance_d(fmeteo)
```

Arguments

fmeteo	Full name of the Meteo-France csv file

Details

The sensors have an additional prefix d (as daily) in order to distinguish them from sensors with another time reference.

Author(s)

P. Chevallier - dec 2023 - jan 2024

`d_convert_meteofrance_h` *Convert a Meteo-France csv hourly basic data file into a htsr sqlite base*

Description

Convert a Meteo-France csv hourly data file into a htsr sqlite base. It regards the "basic" data file, which includes precipitation, temperature and wind data. The csv file shall be downloaded from https://meteo.data.gouv.fr/ The name of the created sqlite file is the same as the csv file with an extension .sqlite.

Usage

```
d_convert_meteofrance_h(fmeteo)
```

Arguments

`fmeteo`	Full name of the Meteo-France csv file

Details

The sensors have an additional prefix h (as hourly) in order to distinguish them from sensors with another time reference.

Author(s)

P. Chevallier - jan 2024

`d_convert_weewx` *Convert a weewx data base into a htsr sqlite base*

Description

Convert (or update) a weewx data base into a htsr sqlite base

Usage

```
d_convert_weewx(
  db.weewx,
  fsq = NA,
  update = TRUE,
  tzo = "Europe/Paris",
  sta = NA,
  name_st = NA
)
```

Arguments

db.weewx	Full name of the weewx data base
fsq	Full name of the htsr data base
update	(default = TRUE)
tzo	Time zone, Olson syntax (default = "Europe/Paris")
sta	Station id (default = NA)
name_st	Station name (default = NA)

Details

If update is TRUE, sta and name_st are unnecessary. I update is FALSE and fsq is NA, fsq is named "weewx.sqlite".

Author(s)

P. Chevallier - Feb 2018 - Jul 2024

Examples

```
## Not run:

d_convert_weewx("weewx.sql", "foo.sqlite")

## End(Not run)
```

---

d_create *Create a data base*

Description

Create htsr sqlite data base

Usage

```
d_create(fsq, cr_table = TRUE, bku = TRUE)
```

Arguments

fsq	Full name of the data base
cr_table	Create the basis tables : TRUE (default), FALSE
bku	Operate a backup if fsq exists : TRUE (default) / FALSE

Details

If the data base already exists and bku is TRUE, a backup is automatically generated.

If cr_table is TRUE, The following tables are also created: ST (stations), SS (sensors), WL (water levels), DI (discharges), PR (Precipitations), WE (weather) and QU (quality)

Value

a new data base

Author(s)

P. Chevallier - Jan 2019

---

d_exp_discalib *Export discharge measurements and calibrations from data base*

Description

Export discharge measurements and calibrations from data base

Usage

```
d_exp_discalib(fsq, sta, calib = TRUE, dism = TRUE)
```

Arguments

fsq	Full name of the data base
sta	Station Id.
calib	Calibration extraction TRUE (default)/FALSE
dism	Discharge measurement extraction TRUE (default)/FALSE

Value

a list of 2 tibbles, one with the calibration table and one with the discharge measurements

Author(s)

P. Chevallier - Sep 2017 - Nov 2020

See Also

ds_exp_hts for export time-series

d_imp_hts *Import a hts file into a data base*

Description

Import a hts file into a tshm sqlite base

Usage

```
d_imp_hts(fsq, filein, table, bku = TRUE)
```

Arguments

fsq	Full name of the data base
filein	Full name of hts file to import
table	Table
bku	Automatic Backup TRUE (default) / FALSE

Details

The main table where the data have to be imported must be selected with one of the following abbreviation: WL (water level), DI (discharge), WE (weather), PR (precipitation) or QU (quality) If records already exist during the same interval, they are removed and replaced.

Value

Actualized data base

Author(s)

P. Chevallier - jan 2019-jan 2024

d_rem_hts *Remove hts records from a data base*

Description

Remove hts records from a Sqlite base

Usage

```
d_rem_hts(fsq, table, sta, sen, start, end)
```

Arguments

fsq	Full name of the data base
table	Table
sta	Station id
sen	Sensor id
start	Start time of removed records
end	End time of removed records

Details

The main table where the data have to be removed must be selected with one the following abbreviation: WL (water level), DI (discharge), WE (weather), PR (precipitation) or QU (quality)

Value

Actualized data base

Author(s)

P. Chevallier - jan 2019 - dec 2022

---

d_table *Create or remove a table of a htsr sqlite base*

Description

The function allows to create or remove of a tshm sqlite base. If the base doesn't exist, it is created.

Usage

```
d_table(fsq, table, op = "C", bku = TRUE)
```

Arguments

fsq	Full name of the data base
table	Table name
op	Create (default) or Remove C/R
bku	Automatic Backup TRUE (default) / FALSE

Details

Possible table names : ST (Stations), SS (Sensors), WL (Water levels), DI (Discharges), WE, (Weather), PR (Precipitations), QU (Quality)

Value

Table created or removed

Author(s)

P. Chevallier - Jan-Feb 2018

See Also

- ds_inventory_station and ds_inventory_station to list the content of the base ;
- ds_exp_hts to extract a time-series

---

fc *Short-cut for file.choose*

Description

Short-cut for file.choose

Usage

```
fc()
```

Value

A filename

Author(s)

P. Chevallier

---

f_change_id *Change Station id or Sensor id in a hts file*

Description

The function changes the station and/or the sensor id of a hts file. The new file is renamed with the new ids and a prefix n_: nw_<sensor.id>_<station.id>.hts, BUT the eventual prefixes or suffixes of the original name are not conserved. The original file is not removed.

Usage

```
f_change_id(file, sta = NA, sen = NA, overwrite = FALSE)
```

Arguments

file	file to proceed
sta	new station id (default: NA)
sen	new sensor id (default: NA)
overwrite	TRUE / FALSE (default) if the output file exists

Author(s)

P. Chevallier - Nov 2017-Jan 2019

---

f_convert *Convert an hts file in another format (xls, xlsx or csv) and vice-versa*

Description

Converter in formats hts, xls, xlsx and text (csv et csv2)

Usage

```
f_convert(file, form_start = "hts", form_end = "xlsx")
```

Arguments

file	Hts file
form_start	Initial format ("hts" (default) or "xls" or "xlsx")
form_end	Final format ("hts" or "xls" or "xlsx" (default) or "csv" (separator , & decimal .) or "csv2" (separator ; and decimal ,)

Details

'form_start' = csv or csv2 is for instance not accepted. It could be converted previously in xls or xlsx format.

Value

A file in the requested format with 4 columns: Date, Value, Station, Sensor

Author(s)

P. Chevallier - October 2017 - May 2022

Examples

```
## Not run:
f_convert(file,  "xlsx", "hts")

## End(Not run)
```

f_csv_multivar *Build a multivariable table file in csv format*

Description

Build a multivariable table file in csv format

Usage

```
f_csv_multivar(files, daily = TRUE, fileo = "fileo")
```

Arguments

files	list of hts files
daily	default = TRUE
fileo	name of the output file (without extension)

Details

The function build a cvs file with values extracted from several hts files at the same date. So, it's better to run 'h_common' before to apply 'f_csv_multivar'

If daily is TRUE, only the date is taking into account, not the time.

Value

A csv table, where the first field is a date and the next fields values

Author(s)

P. Chevallier - Jan-Feb 2022

f_month2day *Interpolation of daily records from a monthly time series*

Description

Interpolation of daily records from a monthly time-series

Usage

```
f_month2day(file)
```

Arguments

file	monthly time series to process

Details

The function build and interpolated daily time-series from a monthly one. The daily values are linearly computed between two consecutive monthly values.

Value

a daily time series

Author(s)

P. Chevallier - dec 2022

`f_properties` *Properties of a hts series*

Description

The function provides the properties of a time-series, its duration and the inventory of its gaps

Usage

```
f_properties(file, gaps = FALSE)
```

Arguments

`file`	: file to be analyzed
`gaps`	: produce a file with a table of the gaps: TRUE / FALSE (default)

Details

If gaps = TRUE, a file is produced, with the same name of file and the extension .gap. It contents a table with the gaps of the series and allows to build a plot with the function `p_gaps`.

Value

Basic infos on a hts time-series

Author(s)

P. Chevallier - Jan 2019 - Oct 2021

See Also

`p_gaps`.

hs_tstep *hts time series with fixed timestep*

Description

Computes time-series with a fixed timestep from infra-daily to monthly within a shiny web page.

Usage

```
hs_tstep()
```

Details

First of all, one must select a "starting" hts file, instantaneous or already with a fixed timestep.

Then one must choose the computing time-step and mode, between the possible choices. Note that the timezone considered is the timezone of the "starting" file.

Possible time-steps are: 5, 10 or 30 minutes, 1, 2, 3, 6 or 12 hours, 1 day, 1 month. It shall be noted that when computing the monthly time step, the daily time step is previously computed.

Possible modes are: average, sum, max or min. For monthly time step, max and min offers two options: daily max averages, respectively min, or absolute, respectively min.

In the case of a daily timestep, a shift value (in hours) allows to shift the time interval. For example if shift = 6, the date is computed from 6am until 6am the following day. The result is dated in the middle of the interval, i.e. if shift = 6; the datetime is 18.

In the case of a monthly timestep, associated additional time series can be optionally computed:

- A mean monthly climatology, taking into account or not the missing daily values with the option "remove NA". Climatology files are by convention awarded to year 2000.
- Excel files: with a calendar presentation (days in rows, months in columns, years in sheets): option caledit_j ; with the monthly means (or sums): option caledit_m.
- Missing values can be replaced by the mean of the existing values for other years: option gapfill.
- Extract year stat

The output files are written in same folder as the starting hts file.

Value

hts files at the requested timestep with a suffix giving the timestep in minutes, i.e. 1440 for the daily timestep. In the case or monthly timestep, the suffixes are: M for the current case, C for the climatology, G for the gapfilled file.

Optionally, two Excel files with values in "calendar form": one with daily data and one with monthly data, the fist one with a ad_ prefix and the second one with the am_ prefix.

Author(s)

P. Chevallier - Oct 2017 - Sep 2023

h_addna *Add NA values within a time series*

Description

Add NA values within a time series

Usage

```
h_addna(file, add)
```

Arguments

file	File name to proceed
add	List of dates with NA values to be added

Details

The function adds records with NA in a time series at given dates. If the date already exists, the value is replaced by NA

The output file is named with a nap_ prefix.

Author(s)

P. Chevallier - November 2022

Examples

```
## Not run:

f <- h_addna (f, add = c("2021-01-01 12:00:00 UTC", "2031-01-01 12:00:00 UTC"))

## End(Not run)
```

h_adjust *Adjust a time series to a statistical model*

Description

Adjust a time series to a statistical model

Usage

```
h_adjust(file, time_unit = "year")
```

Arguments

file	File to proceed
time_unit	to be chosen in: "100y", "year", "month", "day"

Details

The function adjust a time series with a statistical model. For instance it works only with a linear model.

"year" corresponds to an average year of 365.25 days and month to an average month of 30,4575 days.

Author(s)

P. Chevallier - January 2024

---

h_avday *Daily average over a sequence of several years*

Description

Daily average over a sequence of several years

Usage

```
h_avday(file, start = NA, end = NA, mhy = 1, precip = FALSE, dig = 1)
```

Arguments

file	File name to proceed
start	Starting date (default = NA)
end	Ending date (default = NA)
mhy	Starting month of the hydrological year (default = 1)
precip	Precipitation time series (default = FALSE)
dig	Number of significant digits for Value (default = 1)

Details

The function means the values of each calendar day over a period larger than 4 years (i.e. it includes at least one Feb 29 day). The result is transfered to the last possible hydrological year of the interval.

In the special case of precipitation, where the distribution is discontinuous over time, the original values of the last hydrological year are replaced by values corrected proportionately.

Author(s)

P. Chevallier - Nov 2022

Examples

```
## Not run:

f <- h_avday(f,   start=NA, end=NA, mhy=10, precip=TRUE, dig=1)

## End(Not run)
```

---

h_changetz *Change the time zone of a time series*

Description

Change the time zone of a time series

Usage

```
h_changetz(file, tz1 = "UTC", tz2 = "Europe/Paris")
```

Arguments

file	File name to proceed
tz1	original time zone (default = "UTC")
tz2	new time zone (default = "Europe/Paris")

Details

The output file is named with a tz prefix.

Author(s)

P. Chevallier - June 2023

---

h_common *Extract 2 (or more) time-series on their common period*

Description

The fonction extract the data of 2 (or more) hts time-series for the common date/time records (precision of the second).

Usage

```
h_common(files)
```

Arguments

files	List of file names to process.

Value

hts files resulting of the operation; their names are composed as: co_<original filename>

Author(s)

P. Chevallier - Oct 2017 - Oct 2023

Examples

```
## Not run:
f <- h_common(files = c("foo1.hts","foo2.hts"))

## End(Not run)
```

---

h_condition *Conditional extraction of a time-series regarding another one*

Description

The series to proceed is the first of the list, the conditional series the second. Only the common record dates are kept.

Usage

```
h_condition(files, condition)
```

Arguments

files	Liste de 2 file names
condition	Liste 3 objects : oper ("sup" or "inf" or "between"), thrhd1 < thrhd2 ; default is c("inf",0,NA)

Details

If the condition on the file 2 value is not respected, the value of file 1 is changed as NA.

The condition has 3 options : x< ("inf"), x>= ("sup"), < x <= ("between"). In case of error or by default, "inf" is considered. In the cases "inf" and "sup", only one threshold is used (thrhd1) ; in the case "between", two thresholds are needed (thrhd1 < thrhd2).

The output file is the name of the fist file with a cd_ prefix.

Author(s)

P. Chevallier - Oct 2017-Jan 2019

Examples

```
## Not run:

f <- h_condition(c(f1,f2), c("between", 0, 2))

## End(Not run)
```

---

h_cumul *Cumul of time-series*

Description

The function returns a time-series of cumulated values. If the value is negative, the absolute value is taken. It is possible to limit the computation time interval. NA values are ignored.

Usage

```
h_cumul(file, start = NA, end = NA)
```

Arguments

file	File name to proceed
start	Start date, default = NA
end	End date, default NA

Details

The output file is named with a cu_ prefix.

Author(s)

P. Chevallier - Oct 2017-Jan 2019

Examples

```
## Not run:

f <- h_cumul(f, start="2012-1-1", end = "2013-1-1")

## End(Not run)
```

h_gaperr *Replace errors with gaps in a time-series based on neighboring values*

Description

Replace errors with gaps in a time-series based on neighboring values

Usage

```
h_gaperr(file, nv = 1, itv0 = 43201, df)
```

Arguments

file	File name to proceed
nv	Number of below and above neighboring values to take into account, default = 1
itv0	Threshold of minimum time gap (see function h_gaprem_itv)
df	Deviation value factor for testing if a value is correct or not

Details

Replace errors with gaps in a time-series based on neighboring values

Value

a time-series file with the prefix eg_

Author(s)

P. Chevallier - Nov 2019

h_gapfill *Simple gapfilling in a time-series*

Description

Simple gapfilling in a time-series

Usage

```
h_gapfill(file, npdt)
```

Arguments

file	File name to proceed
npdt	Number of time-steps

Details

Replace the missing values with the linear interpolated value within the gap interval, when the time interval is less than a number of fixed time steps.

CAUTION! this operation is only possible when the time-series has a fixed time-step.

Value

a time-series file with the prefix gf_

Author(s)

P. Chevallier - Nov 2017 - Nov 2021

`h_gaprem_itv` *Remove gaps in a time-series with a time interval threshold*

Description

Remove gaps in a time-series with a time interval threshold

Usage

```
h_gaprem_itv(file, itv0 = 43201)
```

Arguments

`file`	File name to proceed
`itv0`	Time threshold in seconds, default = 43201 (i.e 12 hours)

Details

Remove the missing values when the time interval between the previous and the next record is less than a fixed threshold

Value

a time-series file with the prefix gr_

Author(s)

P. Chevallier - Nov 2019

h_nodata *Replace values with NA conditionally or in a time interval*

Description

Replace values with NA conditionally or in a time interval

Usage

```
h_nodata(file, threshold = NA, test = "=", start = NA, end = NA)
```

Arguments

file	File name to proceed
threshold	Threshold value (default = NA)
test	Test "=" (default);"<";"<=";">";">="
start	Start date/time (included) of POSIXct class (default = NA)
end	End date/time (excluded) of POSIXct class (default = NA)

Details

The function replace values with NA conditionally or introduce a gap for a given interval.

For the conditional option, the start parameter must be NA. A conditional test is applied on the values (= ; > ; >= ; < ; <=) with a fixed threshold returning NA if the test is verified.

For the gap option, the threshold parameter must be NA. All the values of the records within the interval start end are replaces by NA.

CAUTION ! At least one of both parameters threshold or start must not be NA. NA.

The output file is named with a na_ prefix.

Author(s)

P. Chevallier - Oct 2017-Jan 2019

Examples

```
## Not run:

f <- h_nodata(f, threshold=10., test= "<=", start=NA)

## End(Not run)
```

`h_rainsnow` *Share the solid and liquid precipitations with a temperature criteria*

Description

The precipitations are shared with a linear bevel between two temperature values

Usage

```
h_rainsnow(fpr, fta, ta0, ta1, sta = NA)
```

Arguments

`fpr`	Precipitation file name
`fta`	Temperature file name
`ta0`	Low temperature threshold
`ta1`	High temperature threshold
`sta`	Station id. (default = NA)

Details

The two time-series must be previously restricted to the same interval of time.

The two temperature thresholds can be equal.

The temperature time-series must be complete with no gap. Gaps are allowed in the precipitation time-series.

Is the station id is NA, the station id of the file fta is used.

Value

2 hts files, one with the liquid precipitation (prefix rn_) and one with the solid precipitation (prefix sn_).

Author(s)

P. Chevallier - Oct 2017- Feb 2019

h_rbind *Bind 2 time-series on consecutive periods*

Description

The fonction binds the data of 2 hts time-series for consecutive date/time records (precision of the second) of the same station.

Usage

```
h_rbind(files, sensor = "NewS", gap = TRUE)
```

Arguments

files	List of char, File names to process.
sensor	New sensor name of the resulting hts file (default ="NewS")
gap	Introduce or not a gap between both series (default = TRUE)

Details

In the list, the files must be ordered from the oldest to the newest. If gap is TRUE, a gap is introduced between both series.

Value

hts file resulting of the operation; its names are composed as: <sensor>_<station>.hts, with the prefix na, if a gap has been introduced.

Author(s)

P. Chevallier - Mar-Nov 2020

Examples

```
## Not run:

f <- h_bind(files = c("foo1.hts","foo2.hts"), sensor = "NewOne")

## End(Not run)
```

h_replace *Replace a value by another*

Description

Replace a value by another

Usage

```
h_replace(file, old.val, new.val)
```

Arguments

file	File name to proceed
old.val	Value to be replaced
new.val	New value

Details

The output file is named with a re_ prefix.

Author(s)

P. Chevallier - Oct 2017- Nov 2020

Examples

```
## Not run:

f <- ts_remplace_ts(f,  NA, 0)

## End(Not run)
```

h_restrict *Restrict a series between 2 dates*

Description

Restrict a series between 2 dates

Usage

```
h_restrict(file, start = NA, end = NA)
```

Arguments

file	File name to proceed
start	Start date/time (included) of POSIXct class (default = NA)
end	End date/time (excluded) of POSIXct class (default = NA)

Details

The output file is named with a rs_ prefix.

Author(s)

P. Chevallier - Nov 2017-Jan 2019

---

h_rollav *Rolling average of a daily time-series*

Description

The function compute a rollong average of daily time-series values. NA values are removed.

Usage

```
h_rollav(file, ti = 7, position = "central")
```

Arguments

file	File name to proceed
ti	Time interval of computation in days (default = 7)
position	Position "central" or "right"

Details

The output file is named with a ro_ prefix. The computation can considers the values before and after the current time step (position = "central") or the values before the current time step. If the position is "central", the position must be an odd integer.

Author(s)

P. Chevallier - Apr 2020

h_season *Seasonal selection*

Description

The function provides seasonal time-series.

Usage

```
h_season(file, monthstart)
```

Arguments

file	Full file name to proceed
monthstart	List of 2 to 4 integers (between 1 and 12) giving the starting month of each season.

Details

2 to 4 seasons can be selected. For each season, the prefix sx_ where x is the season is added to the file name.

Value

list of file names for each seasonal time-series.

Author(s)

P. Chevallier - Oct 2017 - Mar 2020

Examples

```
## Not run:

files <- h_season("foo.hts", monthstart=c(3,6,9,12))

## End(Not run)
```

h_stat_basic *Basic statistics of a time-series*

Description

Compute the main statistic parameters of a time-series

Usage

```
h_stat_basic(files)
```

Arguments

files	vector of file names to process

Value

a tibble with the basic stats of the files.

Author(s)

P. Chevallier - Oct 2017 - Feb 2022

Examples

```
## Not run:
simplestat <- h_stat_basic(c("foo1.hts", "foo2.hts")

## End(Not run)
```

h_substitute *Subtitute the missing values in a series by existing values of another series*

Description

The series to proceed (first in file list) contents missing values or gaps to be replaced by those of the second series (second in file list).

The function only works on the common dates of both series.

Usage

```
h_substitute(files)
```

Arguments

files	List of two file names

Details

The output file is named with a sb_ prefix.

Author(s)

P. Chevallier - Feb 2017 - Mar 2020

Examples

```
## Not run:

f <- h_substitute(c(f1, f2))

## End(Not run)
```

---

h_weightedsum *Weighted sum of time-series*

Description

The function only works on the common period of the files without NA values. It operates weighted sums on one or several time-series. It is also possible to add a constant.

Usage

```
h_weightedsum(files, weights, constant = 0)
```

Arguments

files	List of file names to proceed
weights	List of weights (must have the same length as files)
constant	Constant to add (default = 0)

Details

For averaging n time-series one can use n weights wit a value of 1/n and constant = 0.

Value

The function returns + n hts files with the extracted common period + 1 hts file named as the first file of the list with the prefix w_. The sensor id is automatically set to "weighted".

Author(s)

P. Chevallier - Oct 2017-Oct 2021

Examples

```
## Not run:

# choose time-series f1, f2, f3
f1 <- "foo1.hts" ; f2 <- "foo2.hts" ; f3 <- "foo3.hts"
# the new f time-series contains records f[i] = f1[i] - (0.5 * f2[i]) + (0.5 * f3[i]) + 5
f <- h_weightedsum(c(f1,f2,f3), c(1,-0.5,0.5)), 5)
# the new f time-series contains records f[i] = (1.12 * f1[i]) + 3
f <- h_weightedsum(f1, 1.12, 3)

## End(Not run)
```

---

h_wl_di *Computation of the discharges from water-levels*

Description

Computes a discharge time-series from water levels data and calibration curves

Usage

```
h_wl_di(fsq, sta, seni, seno, dstart = NA, dend = NA, dbo = TRUE)
```

Arguments

fsq	htsr data base
sta	Station Id.
seni	Input sensor Id (water levels)
seno	Output sensor Id (discharges)
dstart	Start date (NA by default)
dend	End date (NA by default)
dbo	Includes the result in the data base (TRUE by default)

Details

Calibration curves must exist in the data base.

If 'dbo' is TRUE, a discharge table "DI" and the sensor 'seno' must exist in the data base. The new discharge time-series overwrites the already existing data ; however, it is asked to confirm the operation. In any case the data base is previously backed up.

Value

Writes an hts file with the resulting discharges and optionally includes it in the data base.

Author(s)

P. Chevallier - Dec 2020 - Sep 2023

See Also

The functions ds_exp_hts and d_imp_htsare used for export the water levels, respectively import the discharges within the data base. The function u_exp_discalib included in p_discalib is used for loading the calibration curves.

---

h_year *Annual time series*

Description

Annual time series

Usage

```
h_year(file, mhy = 1, op = "M", dig = 1)
```

Arguments

file	File name to proceed
mhy	Starting month of the hydrological year (default = 1)
op	Sum (S) or Mean (M) (default = "M")
dig	Number of significant digits for Value (default = 1)

Details

The function computes an annual time-series using the annual mean or the annual sum of daily values. It allows the use of hydrological years. The date corresponds to the middle of the year, i.e. the 182th day.

Value

The function returns a time-series of annual values.

Author(s)

P. Chevallier - Nov 2022

`ps_plothts` *Plot hts files*

Description

This function allows to plot one or several time series files using a shiny web page

Usage

```
ps_plothts()
```

Details

When launched, a shiny window is open. Follow the instructions, divided in 5 steps.

1. Select hts files (8 max) pressing "File select". They must be located in the same folder. When done, press "Enter file settings"
2. For each file, if needed, use the "Edit" tab to configure label, line.type, line.with, point.shape and point.size. (The values follows the ggplot2 package conventions). When done, press "Save file settings"
3. Configure the general layout of the file, entering Title and y-Axis label and choosing a color palette. Several options are available: set y-Axis scale, set time interval, point plot(*), display normalized values, draw a trend line, or display the plot as horizontal facets. When done, press "Save plot settings"
4. Pressing "Plot" displays the graph. You can chose a line or bar graph. When the graph is finalized, check the box "save plot". Three formats are allowed: .png, .jpeg or pdf. The resolution is 300 dpi. Then, press "Save plot settings". The plot is saved in the folder of the selected files.
5. When finished, press "Done".

Items 2 and 3 can be performed and repeated in any order. Once they have been validated once, item 4 can be executed as often as desired.

(*) When point plot is selected, the points overlay the line (point plot doesn't work with bar). If you want only the points on the plot, configure "line.type" and "line.width" = 0.

Author(s)

P. Chevallier - Apr 2015 - Sep 2023

p_box_month *Boxplot of the 12 months of a time-series.*

Description

Boxplot of the 12 months of a time-series.

Usage

```
p_box_month(
  file,
  title = "Title",
  axeY = "Y-axis",
  savefig = FALSE,
  fileo = "plot.png",
  width = 8,
  height = 6
)
```

Arguments

file	File name of the time-series
title	Title plot (default = Title)
axeY	Title of y-axis (default Y-axis)
savefig	Save plot file TRUE / FALSE (default)
fileo	Name of the plot file with extension png, jpg or pdf
width	Plot width (x 100 pixels), default = 8
height	Plot heights (x 100 pixels), default = 6

Value

A ggplot2 object

Author(s)

P. Chevallier - Nov 2017 -Feb 2019

p_clim *Plot climatologies in hydrological year*

Description

This function processes climatology hts files created with hs_tstep.

Usage

```
p_clim(
  files,
  type = "line",
  hydro.month = 1,
  title = "Title",
  yaxis = "Value",
  y.down = NA,
  y.up = NA,
  rpal = FALSE,
  pal = mapalette,
  legend.l = NA
)
```

Arguments

files	List of climatology file names
type	Type: "line" (default) or bar"
hydro.month	Starting month or the hydrological year (default = 1)
title	Title of the plot (default = "Title")
yaxis	Title of y-axis (default = "Value")
y.down	Down limit of y-axis (default = NA)
y.up	Up limit of y-axis (default = NA)
rpal	Choice of a color palette TRUE/FALSE(default)
pal	Color choice or mapalette (default)
legend.l	List of text to be displayed in the plot legend (default = NA)

Details

The parameter tyoe allows to display a line graph or a bar graph.

The parameter hydro.mont fixes the starting month of the hydrological year.

The y-axis scale can be fixed with y.down and y.up.

By default, the color palette is the R one. It can be change with a color list in the pal parameter or choosing mapalette (default in pal)

Par default station_sensor ids are displayed in the legend.l list. But it can be changed entering a list of texts in legend.l,, which must have the same length as the file number.

Value

A ggplot2 object.

Author(s)

P. Chevallier - Feb 2017 - Sep 2023

`p_discalib` *Plot calibration curves water levels vs discharges*

Description

Experimental function, which is for instance limited to only two calibration curves on the same plot.

The function plot the discharges measurements and the corresponding calibration curves starting.

Only the "active" discharge measurements are plotted. The parameter plotdism displays them or not.

One can zoom on a subpart of the plot using the limit values on the x and y axis.

The savefig (default = FALSE by default) parameter allows to save the result i a png, jpg or pdf file, according to the extension of fout.

Usage

```
p_discalib(
  fsq,
  sta,
  sen = "IH",
  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
)
```

Arguments

fsq	Data base file name
sta	Station Id.
sen	Sensor Id. (default = "IH")
plotcalib	Plot calibrations TRUE (default) / FALSE
plotdism	Plot discharge measurements TRUE (default) / FALSE
title	Plot title (default: Title)
savefig	Save plot in a png file TRUE (default) / FALSE
width	Plot width (x 100 pixels) (default = 8)
height	Plot height (x 100 pixels) (default = 6)
fout	Plot file name (default = "plot.png")
limx	Limit x axis TRUE / FALSE (default)
limy	Limit y axis TRUE / FALSE (default)
xinf	Low value for x (default = NA)
xsup	High value for x (default = NA)
yinf	Low value for y (default = NA)
ysup	High value for y (default = NA)

Author(s)

P. Chevallier - Sep 2017 - Dec 2020

---

p_gaps *Plot of data inventory*

Description

This function plot an inventory of the data from one or several station(s)-sensor(s). It is based on the .gap files provided by the function f_properties. It allows to highlight the gaps in time-series.

Usage

```
p_gaps(files, title = "Inventory", BW = FALSE, margin = 0.1)
```

Arguments

files	List of series to plot (hts files)
title	Plot title, default is "Inventory"
BW	Black & white plot TRUE / FALSE (default)
margin	Reserved space for label writing - default is 0.1

Details

The inventories are represented with lines displayed bottom-up in the order of the files list. They are labeled with the station_sensor ids.

Colors are the default colors of ggplot2. For a black & white plot, precise BW = TRUE

The margin value is a reserved space for writing the label at the end of each line. Default value is 0.1 of the difference between the minimum and the maximum date. It shall be adjusted following the length of the labels.

Value

A ggplot2 object

Author(s)

P. Chevallier - Nov 2017 - Sep 2023

See Also

f_properties.

---

p_hypso *Plot the hypsometry curve of one or more basins*

Description

Plot the hypsometry curve of one or more basins

Usage

```
p_hypso(
  file,
  abbrev,
  prop = FALSE,
  range = 50,
  fact = 5,
  title = "Title",
  savefig = FALSE,
  width = 8,
  height = 6,
  fileo = "plot.png"
)
```

Arguments

file	Raster file list of elevation model of basin(s)
abbrev	List of abbreviated basin name(s)
prop	TRUE / FALSE (default) plot a proportion curve of altitude ranges
range	Width of altitude range (default = 50m)
fact	Exagerating factor of the areas (default=5)
title	Title of the plot (default = Title)
savefig	Save the plot in png (default FALSE)
width	Plot width (x 100 pixels) (default = 8)
height	Plot height (x 100 pixels) (default = 6)
fileo	Name of plot file with extension (default = "plot.png")

Value

An object of ggplot2 class

Author(s)

P. Chevallier - Sep 2017- Jun 2023

---

p_scatter *Scatter plot of 2 or more time-series*

Description

The reference time-series is the first of the list. The scatter plot regards only the common dates of the series. In addition to the plot, a linear function is adjusted forcing or not the interception by the origin.

Usage

```
p_scatter(
  files,
  intercept.zero = FALSE,
  remove.zero = FALSE,
  lg.axis = c(NA, NA),
  title = "Title"
)
```

Arguments

files	List of file names to proceed
intercept.zero	TRUE/FALSE (default) force the interception by origin
remove.zero	TRUE / FALSE (default) remove the records with Value = 0 (e.g. precipitations)
lg.axis	Legend list for axis x & y (default = NA)
title	Title of the plot (default: Title)

Value

a table named "result" with 5 columns : variable name, size of the sample, correlation coefficient, regression line slope, interception

Author(s)

P. Chevallier - Oct 2017-Apr 2023

Examples

```
## Not run:

result <- p_scatter(files = c("foo1.RData","foo2.RData"),
          intercept.zero = TRUE)

## End(Not run)
```

---

p_wind *Plot wind roses*

Description

Plot wind roses

Usage

```
p_wind(
  fsq,
  sta,
  swd,
  swv,
  ws.int = 0.5,
  angle = 45,
  grid.line = 10,
  type = "default",
  breaks = 5,
  offset = 5,
  paddle = FALSE,
  key.position = "right"
)
```

Arguments

fsq	Full name of the htsr data base
sta	Station id
swd	Id of wind direction sensor

swv	Id of wind speed sensor
ws.int	Size of speed intervals
angle	Value in percent of the range unit
grid.line	Value in percent of the grid line frequency
type	Type of plot: "default", "year" or "month"
breaks	Number of speed intervals
offset	Size in percent of the central hole
paddle	Shape of the basic elements: if FALSE, polar, if TRUE, rectangular
key.position	Position of the legend

Details

For a detailed description of all parameters see windRose

Value

A wind rose plot

Author(s)

P. Chevallier - Dec 2019 - Sep 2023

See Also

windRose

---

u_index *Compute an index of community*

Description

Compute an index of community

Arguments

nz	length of the concatenated time-series
yd	initial vector of datetimes (in sec)

Details

the function compute an index, which the number of apparition of the same datetime in a time-series

Value

vector of indexes

Author(s)

P. Chevallier - Apr - Oct 2023

`u_timestep` *Compute values in a time-series with a fixed timestep*

Description

Compute values in a time-series with a fixed timestep

Arguments

`te`	time end (in sec)
`yd`	initial vector of datetimes (in sec)
`yv`	initial vector of values
`tst`	timestep (in mn)
`iop`	operation index

Details

iop = 1 for sum; 0 for mean; -2 for min and +2 for max

Value

vector of values with fixed timestep

Author(s)

P. Chevallier - June / Oct 2023

`w_atmp_alt` *Compute atmospheric pressure, function of altitude*

Description

Compute atmospheric pressure, function of altitude

Usage

```
w_atmp_alt(f_atmp, f_temp, alt)
```

Arguments

f_atmp	File name of the known atmospheric pressure ts (mb)
f_temp	File name of the air temperature at the known altitude (°C)
alt	Altitude of the computed air- temperature ts (m)

Details

The function computes an atmospheric pressure time-series at a given altitude, based on a known atmospheric pressure time-series at the sea level. It also needs the air temperature time-series at the sea level for the same times.

In order to verify that both time-series correspond, it is strongly recommended to run previously the function h_common.

Value

An hts file with the suffixe _<alt>

Author(s)

P. Chevallier - Nov 2021 / Nov 2022

---

w_etp *Compute the potential evapotranspiration with several methods*

Description

ETP calculation

Usage

```
w_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
)
```

Arguments

method	Method "Turc", "Penman-Monteith", "Priestley-Taylor", "Makkink", "Heargraves-Samani"
freq	Frequency "day", "month"
f_temp	File of air temperature in degC, mandatory
f_relh	File of relative humidity in percent, mandatory
f_radg	File of global radiation in W/m2
f_radn	File of net radiation in W/m2
f_atmp	File of atmospheric pressure in hPa
f_wvel	File of wind velocities in m/s
f_tmin	File of air min temperature in degC
f_tmax	File of air max temperature in degC
lat	Latitude in deg
alt	Altitude in m
albedo	Albedo
z	Anemometer high in m

Details

f_temp and f_relh are mandatory in all cases.

For the Turc method, f_radg is needed.

For the Penman-Monteith method, f_atmp, f_wvel, h and z are needed. If f_radn is not avalaible, lat, f_tmin and ftmax are also needed.

The Turc method only works with a monthly frequence.

Value

An hts files resulting of the operation with a name composed as:

<J or M><EtpTu>_<Station_id>.hts for the Turc method,

<J or M><EtpPM>_<Station_id>.hts for the Penman-Monteith method,

<J or M><EtpPT>_<Station_id>.hts for the Priestley-Taylor method

<J or M><EtpMa>_<Station_id>.hts for the Makkink method

<J or M><EtpHS>_<Station_id>.hts for the Heargraves-Samani method

Author(s)

P. Chevallier - April 2020-Nov2022

Source

Hingray, B., Picouet, C., Musy A., Hydrologie, une science pour l'ingénieur, Presses Plolytechniques et Universitaires Romandes, 2008,

Allen, R.G., L.S. Pereira, D. Raes, and M. Smith. 1998. Crop Evapotranspiration. Guidelines for Computing Crop Water Requirements. FAO Irrigation and Drainage Paper 56. 300p

Er-Raki, S., A. Chehbouni, S. Khabba, V. Simonneaux, L. Jarlan, A. Ouldbba, J. C. Rodriguez, and R. Allen. 2010. "Assessment of Reference Evapotranspiration Methods in Semi-Arid Regions: Can Weather Forecast Data Be Used as Alternate of Ground Meteorological Parameters?" Journal of Arid Environments 74 (12): 1587–96. https://doi.org/10.1016/j.jaridenv.2010.07.002.

`w_spechum2relhum` *Convert specific humidity to relative humidity*

Description

Convert specific humidity to relative humidity

Usage

```
w_spechum2relhum(f_spechum, f_temp, f_atm)
```

Arguments

`f_spechum`	file of specific humidity, dimensionless (e.g. kg/kg) ratio of water mass / total air mass
`f_temp`	file of temperature degrees C
`f_atm`	file of atmospheric pressure in mb

Details

Converting specific humidity into relative humidity. from Bolton 1980 The computation of Equivalent Potential Temperature

Value

a file of relative humidity, ratio of actual water mixing ratio to saturation mixing ratio

Author(s)

P. Chevallier - Nov 2022

Source

David LeBauer - 2014

from Bolton 1980 The computation of Equivalent Potential Temperature

https://earthscience.stackexchange.com/questions/2360/how-do-i-convert-specific-humidity-to-relativ

`w_temp_alt` *Compute temperature, function of altitude*

Description

Compute temperature, function of altitude

Usage

```
w_temp_alt(file, alt0 = 0, alt, grad = -0.0065)
```

Arguments

`file`	File name of the known air temperature ts (°C)
`alt0`	Altitude of the known air temperature ts - default = 0 (m)
`alt`	Altitude of the computed air- temperature ts (m)
`grad`	Temperature gradient vs elevation - default = -0.0065 (°C/m)

Details

The function computes an air temperature time-series at a given altitude, based on a known air temperature time-series at a known altitude.

Value

An hts file with the suffix _<alt>

Author(s)

P. Chevallier - Nov 2021

`z_coord` *Coordinate utility*

Description

Convert numeric coordinates in character coordinates

Usage

```
z_coord(ncoord = NA, ccoord = NA, type)
```

Arguments

ncoord	Numeric coordinate
ccoord	Character coordinate
type	Lat / Lon

Details

Only one of both parameters ncoord (numeric) and ccoord (character) must be filled, the other one remaining NA. The type of coordinate (Lat or Lon) is compulsory.

The character coordinate must be organized in one string with 4 fields (degrees, minutes, seconds, direction) separated with blanks (space or tab). Within each field, no blanks are allowed to share the numeric value and the unit character. For the unit character, the only following letters are allowed: letter d/m/s. For direction, the only the following letters are allowed: N/n/W/w/S/s/E/e.

Example: "25d 18m 56.2s S"

Value

Coordinates in characters

Author(s)

P. Chevallier - Jan 2019 / Nov 2020

Index

d_backup, 6
d_compact, 7
d_convert_eaufrance, 7
d_convert_hydraccess, 8
d_convert_meteofrance_d, 9
d_convert_meteofrance_h, 10
d_convert_weewx, 10
d_create, 11
d_exp_discalib, 12
d_imp_hts, 13, *36*
d_rem_hts, 13
d_table, 14
ds_exp_hts, 3, *9*, *12*, *15*, *36*
ds_inventory_sensor, 4, *9*
ds_inventory_station, 4, *9*, *15*
ds_sensor, 5
ds_station, 6

f_change_id, 15
f_convert, 16
f_csv_multivar, 17
f_month2day, 17
f_properties, 18, *41*, *42*
fc, 15

h_addna, 20
h_adjust, 20
h_avday, 21
h_changetz, 22
h_common, 22, *47*
h_condition, 23
h_cumul, 24
h_gaperr, 25
h_gapfill, 25
h_gaprem_itv, 26
h_nodata, 27
h_rainsnow, 28
h_rbind, 29
h_replace, 30
h_restrict, 30
h_rollav, 31
h_season, 32
h_stat_basic, 33
h_substitute, 33
h_weightedsum, 34
h_wl_di, 35
h_year, 36
hs_tstep, 19, *39*

p_box_month, 38
p_clim, 39
p_discalib, *36*, 40
p_gaps, *18*, 41
p_hypso, 42
p_scatter, 43
p_wind, 44
ps_plothts, 37

u_index, 45
u_timestep, 46

w_atmp_alt, 46
w_etp, 47
w_spechum2relhum, 49
w_temp_alt, 50
windRose, *45*

z_coord, 50