Package 'flowdurr'

May 12, 2014

Type Package

Title What the package does (short line)

Version 1.0

Date 2014-05-12

Author Who wrote it

Maintainer Who to complain to <yourfault@somewhere.net>

Description More about what it does (maybe more than one line)

License What license is it under?

R topics documented:

flowdurr-package *What the package does (short line)*

Description

More about what it does (maybe more than one line)

Details

Package:	flowdurr
Type:	Package
Title:	What the package does (short line)
Version:	1.0
Date:	2014-05-12
Author:	Who wrote it
Maintainer:	Who to complain to <yourfault@somewhere.net>
License:	What license is it under?

Author(s)

Who wrote it

add_wateryear *Add a wateryear column.*

Description

wrapper for get_wateryear() that adds a column named 'WYD' in the second column of a dataframe produced by get_waterdata() or get_forecast().

Usage

```
add_wateryear(d)
```

Arguments

d	a dataframe generated by get_forecast() or get_waterdata().

Value

the supplied dataframe, with 'WYD' in second column.

Author(s)

Who wrote it

Examples

```
#  d = data.frame(GMT=as.Date(c('2012-10-01', '2013-01-01', '2013-09-30')),
#                 x=c(10,20,30))
#  add_wateryear(d)
```

`clean_flowdata` *Clean up flow data.*

Description

Replace negative, 0 or NA entries in the flow data.

Usage

```
clean_flowdata(d, s = c(0, NA, -1), r)
```

Arguments

`d`	a dataframe produced by get_forecast() or get_waterdata().
`s`	the value targeted for substitution. s = -1 means negative values will be replaced. s = NA means NA values will be replaced. s = 0 means zero values will be replaced.
`r`	the replacement value, either NA or numeric.

Value

the same dataframe, with all zero values replaced.

Author(s)

Who wrote it

Examples

```
#
```

`convert_from_cfsd` *Flow to volume conversion.*

Description

convert from thousand cubic-feet/second days to cubic feet or acre-feet.

Usage

```
convert_from_cfsd(d, to = c("cf", "af"))
```

Arguments

d	a dataframe produced by get_forecast() or get_waterdata().
to	convert to cubic feet ('cf') or acre-feet ('af').

Details

CFSD stands for cubic feet per second (cfs) averaged over the day (D), i.e. "mean daily flow in cfs". Volume is calculated by multiplying flows by the 86400 seconds/day to achieve units of cubic feet. Conversion to acre-feet is then achieved by multiplying by the ratio of acres to square feet.

Value

the same dataframe, but with a conversion factor applied.

Author(s)

Who wrote it

Examples

```
#  d = data.frame(GMT=seq(0,10,1), WYD=seq(2014001, 2014011),
#                 x=seq(0, 10, 1), y=seq(0, 20, 2))
#  convert_from_cfsd(d, 'cf')
#  convert_from_cfsd(d, 'af')
```

---

get_excurves *Calculate exceedance curves.*

Description

generate fitted exceedance curves for plotting.

Usage

```
get_excurves(parameterdata, distr, probs = c(seq(0.002, 0.019,
    0.001), seq(0.02, 0.99, 0.01)))
```

Arguments

parameterdata	dataframe of parameters as generated by get_parameters().
distr	The distribution corresponding to the parameter estimates. Can be either log Pearson III ('lp3') or 3-parameter lognormal ('ln3').
probs	the quantiles used to generate the curve.

Value

a dataframe for plotting exceedance curves for each flow duration. Rows correspond to probability values contained in column 'p.exceedance'. Column names are the flow durations.

Author(s)

Who wrote it

Examples

```
#
```

`get_forecast` *Download ESP traces from CNRFC.*

Description

download forecast data from the internet and get it into R.

Usage

```
get_forecast(datestring)
```

Arguments

`datestring`	the date to download, represented by the string YYYY-MM-DD or R date class. e.g. October 12, 1980 is "1980-10-12"

Details

The California Nevada River Forecast Center provides daily flow predictions generated for ensemble forecasting. Daily flows are reported as daily average instantaneous flow in thousand cubic feet per second, or kcfsd (i.e. 1 kcfs = 1000 cfs (d)aily). Flow values are multiplied by a factor of 1,000 so that the data is returned in cfsd.

Value

a dataframe containing the trace data.

Author(s)

Who wrote it

Examples

```
#  get_forecast(as.Date(Sys.time()))
```

get_parameters *Parameter estimates for exceedance curve fitting.*

Description

estimate the parameters of a log Pearson III or 3-parameter lognormal distribution.

Usage

```
get_parameters(flowdata, distr = c("lp3", "ln3"), ...)
```

Arguments

flowdata	flow duration data to be used for fitting, as created by get_peakavg().
distr	distribution to be fitted, either log Pearson III ('lp3') or lognormal ('ln3').
...	other parameters to pass to pel-functions. For more information, see lmom::pelln3 and lmom::pelpe3.

Details

The function uses the lmom package to compute method of L-moment estimates. See lmom package documentation for information on definitions of the log Pearson III and 3-parameter lognormal distributions.

Value

a dataframe of similar construction to one generated by get_peakavg(). Column names are flow durations, named rows are the distribution parameters.

Author(s)

Who wrote it

Examples

```
#
```

get_peakavg *Calculate peak flow durations.*

Description

get the peak average flows for many flow records.

Usage

```
get_peakavg(d, flowduration = c(1, 3, 7, 15, 30, 60, 90, 120,
    183), na.rm = TRUE)
```

Arguments

d	a dataframe generated by get_forecast() or get_waterdata().
flowduration	a vector of window sizes for calculating moving averages.
na.rm	logical: remove NA values prior to calculating peak flow durations?

Details

the function uses the rollmean() function in package 'zoo' to compute flow duration timeseries. The maximum value of each averaged timeseries is the peak flow for that flow duration.

Value

a dataframe.

Author(s)

Who wrote it

Examples

```
#  d1 = add_wateryear(get_forecast(as.Date(Sys.time()) ))
#  d2 = split_by_calendaryear(add_wateryear(get_waterdata(startdate='2010-10-01')))
#  get_peakavg(d1, c(1,3,7,15,30,60,90))
#  get_peakavg(d2, c(1,3,7,15), na.rm=TRUE)
```

---

get_ppositions *Cunnane plotting positions.*

Description

calculate plotting positions for empirical cdf.

Usage

```
get_ppositions(pafdata, alpha = 0.4)
```

Arguments

pafdata	a data frame generated by get_peakavg().
alpha	shape parameter, default is 0.4.

Value

a data frame of size pafdata containing the plotting positions of observations for each record.

Author(s)

Who wrote it

References

Cunnane 1978, J. Hydrol. 37:205-222 Helsel and Hirsch, Statistical Methods in Water Resources, USGS.

Examples

```
#  d1 = add_wateryear(get_forecast(as.Date(Sys.time()) ))
#  d2 = split_by_calendaryear(add_wateryear(get_waterdata(startdate='2010-10-01')))
#  p1 = get_peakavg(d1, c(1,3,7,15,30,60,90))
#  p2 = get_peakavg(d2, c(1,3,7,15), na.rm=TRUE)
#  get_ppositions(p1)
#  get_ppositions(p2, 0.4)
```

---

get_waterdata *Download daily flow data from USGS.*

Description

download flowdata from a USGS gauge using the waterData package.

Usage

```
get_waterdata(startdate = NULL, enddate = NULL, locid = "11173200")
```

Arguments

startdate	start date of record, string of form "YYYY-MM-DD".
enddate	end date of record, same format as startdate.
locid	the station id, default is Arroyo Hondo '11173200'.

Details

flow data is downloaded using the waterData package for R. Daily mean flow data (parameter code 00060, stat code 00003) is specified.

Value

a continuous time series from the waterData package.

Author(s)

Who wrote it

Examples

```
#  get_waterdata()
#  get_waterdata(startdate='2011-10-01', enddate='2012-09-30')
```

plot_all_traces *Plot ESP traces.*

Description

plot traces in dataset, as well as mean and median flows.

Usage

```
plot_all_traces(d)
```

Arguments

d	a dataframe generated by get_forecast() or split_by_*year(get_waterdata()).

Value

a ggplot plot of all traces, the mean flow, and the median flow.

Author(s)

Who wrote it

Examples

```
#  d1 = add_wateryear(get_forecast(as.Date(Sys.time()) ))
#  d2 = split_by_calendaryear(add_wateryear(get_waterdata()))
#  plot_all_traces(d1)
#  plot_all_traces(d2)
```

plot_curves *Plot empirical and fitted exceedance curves.*

Description

plot the exceedance curves and empirical data.

Usage

```
plot_curves(flowdata, ppos, curvedata, distr, units = c("cf",
    "af"))
```

Arguments

flowdata	flow data generated by get_peakavg().
ppos	plotting positions generated by get_ppositions().
curvedata	fitted curves generated by get_excurves().
distr	the distribution used to generate the fitted curves.
units	the units to be used for displaying flow. Options are cubic feet ('cf') or acre-feet ('af').

Value

a ggplot2 plot.

Author(s)

Who wrote it

Examples

#

`plot_diagnostics` *Diagnostic plots for flow duration curves*

Description

Create two diagnostic plots for checking distribution fits

Usage

```
plot_diagnostics(dp, distr)
```

Arguments

`dp`	a dataframe of parameter values produced by get_parameters()
`distr`	the distribution of the supplied parameters. Options are log Pearson type III ('lp3') or 3-parameter lognormal ('ln3').

Author(s)

Who wrote it

Examples

#

`plot_empirical_exceedance`
Plot empirical exceedance curves.

Description

plot the empirical cumulative distribution function using plotting positions.

Usage

```
plot_empirical_exceedance(flows, ppos)
```

Arguments

flows	a dataframe produced by get_peakavg().
ppos	a dataframe produced by get_ppositions().

Value

a ggplot plot.

Author(s)

Who wrote it

Examples

```
#  d1 = add_wateryear(get_forecast(as.Date(Sys.time()) ))
#  d2 = split_by_calendaryear(add_wateryear(get_waterdata(startdate='2010-10-01')))
#  p1 = get_peakavg(d1, c(1,3,7,15,30,60,90))
#  p2 = get_peakavg(d2, c(1,3,7,15), na.rm=TRUE)
#  plot_empirical_exceedance(p1, get_ppositions(p1))
#  plot_empirical_exceedance(p2, get_ppositions(p2, 0.4))
```

---

plot_trace *Plot a single ESP trace.*

Description

plot an individual trace.

Usage

```
plot_trace(d, colname)
```

Arguments

d	a dataframe generated by get_forecast() or split_by_*year(get_waterdata()).
colname	the column name of the trace to be plotted.

Value

a ggplot plot.

Author(s)

Who wrote it

Examples

```
#  d1 = add_wateryear(get_forecast(as.Date(Sys.time()) ))
#  d2 = split_by_calendaryear(add_wateryear(get_waterdata()))
#  plot_trace(d1, tail(names(d1), 1))
#  plot_trace(d2, tail(names(d2), 1))
```

replace_zero_flows *Replace zero values in a dataframe.*

Description

replace numeric values of zero with a specified numeric placeholder.

Usage

```
replace_zero_flows(d, r)
```

Arguments

d	a dataframe generated by get_forecast() or get_waterdata().
r	the value to substitute, of type numeric.

Value

the same dataframe, with all zero values replaced.

Author(s)

Who wrote it

Examples

```
#  d = data.frame(x=c(1,2,3), y=c(0, 5, 6), z=c(0,0,9))
#  replace_zero_values(d, 0.001)
#  replace_zero_values(d, 10)
#  replace_zero_values(d, NA)
```

split_by_calendaryear *Split a timeseries by calendar year.*

Description

splits a timeseries downloaded using the waterData package by calendar year.

Usage

```
split_by_calendaryear(d)
```

Arguments

d	a dataframe generated by get_waterdata().

Value

a dataframe with columns 'GMT' and wyXXXX, where XXXX is a year in the record.

Author(s)

Who wrote it

Examples

```
#  d = get_waterdata()
#  split_by_calendaryear(d)
```

---

split_by_wateryear *Split a timeseries by water year.*

Description

splits a timeseries downloaded using the waterData package by water year.

Usage

```
split_by_wateryear(d)
```

Arguments

d	a dataframe generated by get_waterdata() and add_wateryear().

Value

a dataframe with each column containing data from each water year in the record. Missing values are NA.

Author(s)

Who wrote it

Examples

```
#  d = get_waterdata()
#  split_by_wateryear(d)
```

---

strip_na_cols *Remove NA columns.*

Description

remove columns that only contain NA values.

Usage

```
strip_na_cols(d)
```

Arguments

d	a dataframe.

Value

the supplied dataframe, but with NA columns removed.

Author(s)

Who wrote it

Examples

```
#  d = data.frame(w=seq(0, 10), x=c(rep(NA, 6), seq(5)),
#                 y=rep(NA, 11), z=seq(0, 20, 2))
#  strip_na_cols(d)
```

`subset_forecast` *Subset ESP trace data.*

Description

pull time interval and/or specific locations from forecast data.

Usage

```
subset_forecast(d, sc = c("GMT", "WYD"), startdate = NULL, enddate = NULL,
    location = NULL)
```

Arguments

`d`	a dataframe generated by get_forecast.
`sc`	name or index of column to use for date, e.g. 'GMT' or 'WYD'. Default is 'GMT'.
`startdate`	the start of interval, date format e.g. as.Date('2012-01-15') or string 'YYYY-MM-DD', or numeric water year date, e.g. October 1st, 2013 is 2014001.
`enddate`	the last entry in interval, same format as startdate.
`location`	the region identifier, e.g. Arroyo Hondo is 'AHOC1'.

Value

return subset of data.

Author(s)

Who wrote it

Examples

```
#  d = get_forecast(as.Date(Sys.time()))
#  d = add_wateryear(d)
#  subset_forecast(d, location='AHOC1')
#  subset_forecast(startdate='2014-04-01', enddate='2014-06-30')
#  subset_forecast(sc='WYD', enddate=2014365)
```

subset_waterdata *Subset downloaded USGS flow data.*

Description

subset waterdata by interval.

Usage

```
subset_waterdata(d, sc = c("WYD", "GMT"), startdate = NULL, enddate = NULL,
    na.cols = TRUE)
```

Arguments

d	a dataset generated by split_by_wateryear() or split_by_calendaryear().
sc	search column to use for subsetting by date.
startdate	the first day of the interval, either 'MM-DD' string or 3-digit WYD.
enddate	the last day of the interval, same format as startdate.
na.cols	Logical: remove columns with all values NA?

Value

a subset of the original dataframe.

Author(s)

Who wrote it

Examples

```
#  d = get_waterdata()
#  subset_waterdata(d, startdate='2013-10-01')
```

to_wateryear *Hydrologic date conversion.*

Description

converts a date to a water year.

Usage

```
to_wateryear(thedate)
```

Arguments

thedate	a date created by as.Date('YYYY-MM-DD') or a string of form 'YYYY-MM-DD'.

Value

an interger of form YYYYDDD, e.g. 2014-01-01 is 2014001.

Author(s)

Who wrote it

Examples

```
#  to_wateryear(as.Date(Sys.time()))
#  to_wateryear('2013-10-01')
#  to_wateryear('2014-09-30')
#  to_wateryear('2012-09-30')
```

Index