

Package ‘SWTools’

September 6, 2021

Type Package

Title Helper Tools for Australian Hydrologists

Version 0.2.1.0

Author Matt Gibbs [aut, cre]

Maintainer Matt Gibbs <gibbs.ms@gmail.com>

URL <https://github.com/matt-s-gibbs/SWTools>

BugReports <https://github.com/matt-s-gibbs/swtools/issues>

Description Functions to speed up work flow for hydrological analysis.

Focused on Australian climate data (SILO climate data), hydrological models (eWater Source) and in particular South Australia (<<https://water.data.sa.gov.au>> hydrological data).

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

Imports httr, zoo, RColorBrewer, dplyr, tidyr, tibble, forcats, stringr, rlang, rmarkdown, reshape2, ggplot2, ggmap, ggrepel, magrittr, hydroTSM, jsonlite, sf

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-09-06 09:00:14 UTC

R topics documented:

AQWPDownload	2
AQWPLoad	4
read_res.csv	4
SILOCumulativeDeviation	5
SILODoubleMass	6
SILODownload	7

SILOImport	8
SILOLoad	9
SILOMap	10
SILOMonthlyRainfall	10
SILOMortonQualityCodes	11
SILOQualityCodes	12
SILOReport	13
SILOSiteSummary	14
SILOThiessenShp	15
SILOWriteforSource	15
SWTools	16
VeneerGetInputSets	17
VeneerGetNodesbyType	18
VeneerGetPiecewise	18
VeneerGetTS	19
VeneerGetTSbyNode	20
VeneerGetTSbyVariable	20
VeneerGetTSVariables	21
VeneerlatestRunNumber	22
VeneerRunSource	22
VeneerSetFunction	23
VeneerSetPiecewise	24
WritepwttoIS	25

Index 26

AQWPDDownload	<i>Function to download data from https://water.data.sa.gov.au</i>
---------------	---

Description

For most inputs, valid options will be returned if an unexpected input is provided. The exception are **Location** and **Dataset**, if the location, or dataset for that location, don't exist no data will be returned. Browse the Export tab on <https://water.data.sa.gov.au> to find **Location** and **Dataset** that exists.

Usage

```
AQWPDDownload(
  Location,
  Dataset,
  Unit,
  file = "AQWP.json",
  Interval = "Daily",
  Calculation = "Aggregate",
  Calendar = "WATERDAY_9AM",
  Step = 1,
  DateRange = "EntirePeriodOfRecord",
```

```

    StartTime = NULL,
    EndTime = NULL
)

```

Arguments

Location	A string or vector of strings, with site numbers, e.g. "A4261001"
Dataset	A string or vector of strings, with dataset names, as expected by AWQP, e.g. "Tide Height.Best Available--Continuous"
Unit	A string or vector of strings, with units, e.g. "Metres" or "mg/L". If only 1 is provided it will be used for each site in Location
file	Location and name of json file to download. Defaults to "AQWP.json".
Interval	Interval of output, e.g. "PointsAsRecorded", or "Daily"
Calculation	For larger intervals, what calculation to do, e.g. "Aggregate" (average) or "Maximum"
Calendar	When to start the periods, e.g. "WATERDAY_9AM"
Step	How many intervals e.g. 15 with Interval="Minutely" returns 15 minute data.
DateRange	Period of data to return, e.g. "EntirePeriodOfRecord" or "Custom". "Years1" seems to not work on AWQP.
StartTime	Start Date and Time if DateRange="Custom", in a format that as.POSIXct will convert, e.g 2000-01-01 00:00
EndTime	End Date and Time if DateRange="Custom", in a format that as.POSIXct will convert, e.g 2001-01-02 00:00

Value

nothing to the environment. Saves a file to "file", that can then be read in with AQWPLoad()

Examples

```

## Not run:
Location=c("A4260633", "A4261209", "A4260572")
Dataset=rep("Tide Height.Best Available--Continuous", 3)
Unit=rep("Metres", 3)
S="2020-01-01 00:00"
E="2020-01-02 00:00"
AQWPDownload(Location, Dataset, Unit, DateRange="Custom",
  StartTime=S, EndTime=E, file=tempfile("AQWP", fileext=".json"))

## End(Not run)

```

AQWPLoad	<i>Function to load in an Aquarius json file, downloaded from https://water.data.sa.gov.au, possibly using AWQPDownload()</i>
----------	--

Description

Function to load in an Aquarius json file, downloaded from <https://water.data.sa.gov.au>, possibly using AWQPDownload()

Usage

```
AQWPLoad(filename, qual_codes = TRUE, long_format = TRUE)
```

Arguments

filename	A file downloaded from the Export Data tab on https://water.data.sa.gov.au , or using AWQPDownload()
qual_codes	TRUE/FALSE to return quality codes. Defaults to true
long_format	TRUE/FALSE to return data in long format, rather than wide (e.g. a spreadsheet). Long is useful for plotting with ggplot

Value

A tibble with the data in the file

Examples

```
## Not run:
AQWPLoad("AQWP.json")

## End(Not run)
```

read_res.csv	<i>Function to inport a Source .res.csv File Returns data (as a Data Frame, Zoo, or tibble) as a time Series with all Results Read Source .res.csv file into a data table or zoo time series</i>
--------------	--

Description

Function to inport a Source .res.csv File Returns data (as a Data Frame, Zoo, or tibble) as a time Series with all Results Read Source .res.csv file into a data table or zoo time series

Usage

```
read_res.csv(resFile, returnType = "df")
```

Arguments

resFile	A character string representing the full file path of the .res.csv file
returnType	A character string to set the return type: "z", "t", "df". If not matching "t" (tibble) or "z" (zoo), data frame returned.

Value

Data in the format selected with all data read in from the Source .res.csv file

Examples

```
## Not run:
X = read_res.csv("./SWTools/extdata/Scenario1.res.csv",returnType="t")

## End(Not run)
```

SILOCumulativeDeviation

Plot the cumulative deviation from the mean for each silo station on one plot

Description

Plot the cumulative deviation from the mean for each silo station on one plot

Usage

```
SILOCumulativeDeviation(SILO, filename = NULL, cols = pkg.env$cols)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	optional, filename to write the plot to, including extension (e.g. .png). Filename can include full path or sub folders.
cols	optional, a vector of colours to use for the plotting

Value

a ggplot plot of the cumulative deviation from the mean.

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")
p<-SILOCumulativeDeviation(X)

## End(Not run)
```

SILODoubleMass *Plot double mass curves of each rainfall site against each other*

Description

Plot double mass curves of each rainfall site against each other

Usage

```
SILODoubleMass(SILO, filename = NULL, plotsperpage = 4)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	optional, filename to write the plot to, including extension. Filename can include full path or sub folders.
plotsperpage	optional, number of plots to output per element of the list returned. Defaults to 4

Value

a list of ggplot objects that plot of the double mass curves of each station in the SILO list against each other. The double mass plot is on the bottom diagonal, and the slope of the line for each case in the upper diagonal. Each list element contains plotsperpage (default to 4) double mass plots, to allow them to be plotted on multiple pages

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")
p<-SILODoubleMass(X)

## End(Not run)
```

SILODownload	<i>Download SILO data</i>
--------------	---------------------------

Description

Download SILO data

Usage

```
SILODownload(  
  SiteList,  
  username = "noemail@net.com",  
  password = "gui",  
  path = getwd(),  
  startdate = "18890101",  
  enddate = NULL,  
  ssl = FALSE  
)
```

Arguments

SiteList	A station number or vector of station numbers, as a string (e.g. "24001")
username	SILO user name. Defaults to credentials used by https://www.longpaddock.qld.gov.au/silo/point-data/
password	SILO password
path	Where to save the output. Will default to getwd() if not specified
startdate	First day of data, in the format "YYYYMMDD". Will default to the first day of the record "18890101" if not specified
enddate	Last day of data, in the format "YYYYMMDD". Will default to yesterday if not specified
ssl	if true set ssl_cipher_list to "RC4-SHA" for file download. Seems to be necessary on some machines. default to FALSE

Value

A file for each station will be saved to path, named station number.txt. Nothing is returned to the R environment.

Examples

```
## Not run:  
SILODownload(c("24001", "24002", "24003"),  
  path=tempdir(),  
  startdate="20180101", enddate="20200101")  
  
## End(Not run)
```

 SILOImport

Import a SILO file

Description

Import a SILO file

Usage

```
SILOImport(station, path, startdate, enddate)
```

Arguments

station	Station number (e.g. "24001") to import. The function expects the file to be called "24001.txt".
path	Location where the file is located. Use "/" or "\" for folders. Defaults to getwd() if not specified.
startdate	Start date of data to load, in format "YYYY-MM-DD". Defaults to start of the file if not provided
enddate	End date of data to load, in format "YYYY-MM-DD". Defaults to end of the file if not provided

Value

a list of data from the file, with members:

tsd the raw data as a daily zoo object

Site the name of the site

Station the station number

Lon Longitude

Lat Latitude

start the first date with good quality rainfall data

end the last date with good quality rainfall data

goodpct the percentage of good quality coded rainfall data between start and end

SILOLoad

Import multiple SILO files

Description

Import multiple SILO files

Usage

```
SILOLoad(sites, path = getwd(), startdate, enddate)
```

Arguments

sites	a vector of Station numbers (e.g. c("24001","24002","24003")) to import. The function expects the file to be called "24001.txt".
path	Location where the file is located. Use "/" or "\\" for folders. Defaults to getwd() if not specified.
startdate	Start date of data to load, in format "YYYY-MM-DD". Defaults to start of the file if not provided
enddate	End date of data to load, in format "YYYY-MM-DD". Defaults to end of the file if not provided

Value

a list of data from the file, with members:

tsd the raw data as a daily zoo object

Site the name of the site

Station the station number

Lon Longitude

Lat Latitude

start the first date with good quality rainfall data

end the last date with good quality rainfall data

goodpct the percentage of good quality coded rainfall data between start and end

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")

## End(Not run)
```

SILOMap

Plot a map of the SILO station locations

Description

Plot a map of the SILO station locations

Usage

```
SILOMap(SILO, filename = NULL)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	optional, filename to write the plot to, including extension. Filename can include full path or sub folders.

Value

a google map of the SILO station locations

Examples

```
## Not run:  
X<-SILOLoad(c("24001", "24002", "24003"), path = "./SWTools/extdata")  
p<-SILOMap(X)  
  
## End(Not run)
```

SILOMonthlyRainfall*Plot a boxplot of monthly rainfall with mean monthly evaporation*

Description

Plot a boxplot of monthly rainfall with mean monthly evaporation

Usage

```
SILOMonthlyRainfall(  
  SILO,  
  evapcol = "Mwet",  
  filename = NULL,  
  cols = pkg.env$cols  
)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
evapcol	name of an evaporation column to plot, defaults to "MWet".
filename	optional, filename to write the plot to, including extension. Filename can include full path or sub folders.
cols	optional, a vector of colours to use for the plotting

Value

a ggplot of the monthly rainfall and evaporation.

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")
p<-SILOMonthlyRainfall(X, "Span", cols=c("black", "red", "#124734"))

## End(Not run)
```

SILOMortonQualityCodes

Plot the quality codes of the input data for Morton's Evap calculations

Description

Produces a tile plot displaying the quality codes for variables that are input to the calculation of Morton's evaporation equations, being maximum and minimum temperature, solar radiation and vapor pressure (derived from wet bulb temperature). Evaporation is also plotted, if the site has pan observations.

Usage

```
SILOMortonQualityCodes(SILO, filename = NULL)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	optional, filename to write a plot of the rainfall quality codes to, including extension (e.g. png). Filename can include full path or sub folders.

Value

a ggplot geom_tile plot of the rainfall quality codes

Examples

```
## Not run:  
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")  
p<-SILOMortonQualityCodes(X)  
  
## End(Not run)
```

SILOQualityCodes	<i>Plot the quality codes of the SILO rainfall data</i>
------------------	---

Description

Plot the quality codes of the SILO rainfall data

Usage

```
SILOQualityCodes(SILO, filename = NULL)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	optional, filename to save a plot of the rainfall quality codes to, including extension (e.g. .png). Filename can include full path or sub folders.

Value

a ggplot geom_tile plot of the rainfall quality codes

Examples

```
## Not run:  
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")  
p<-SILOQualityCodes(X)  
  
## End(Not run)
```

SILOReport	<i>Write SILO data report to word document. The report includes output from SILOSiteSummary(), SILOQualityCodes(), SILOMortonQualityCodes(). SILOMap(), SILOMonthlyRainfall(), SILOCumulativeDeviation() and SILODoubleMass().</i>
------------	--

Description

Write SILO data report to word document. The report includes output from SILOSiteSummary(), SILOQualityCodes(), SILOMortonQualityCodes(). SILOMap(), SILOMonthlyRainfall(), SILOCumulativeDeviation() and SILODoubleMass().

Usage

```
SILOReport(SILO, filename, path = getwd(), cols = pkg.env$cols)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
filename	filename to write the report to.
path	Optional. Folder to save the report to, defaults to current working directory
cols	Optional. vector of colours to use for the monthly rainfall and cumulative deviation plots. Must be at least as long as the number of sites in the SILO list.

Value

Nothing to the environment. A word document report is written to "filename".

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path = "/SWTools/extdata")
SILOReport(X, "MyReport.docx") #requires pandoc installed

## End(Not run)
```

SILOSiteSummary *Produce a table summarising SILO sites*

Description

Produce a table summarising SILO sites

Usage

```
SILOSiteSummary(SILO)
```

Arguments

SILO a list of sites with SILO data, as created by SILOLoad()

Value

a dataframe with the following columns

Site site name

Station station number

StartDate date of the first good quality rainfall data

EndDate date of the last good quality rainfall data

PctMissing percentage of days that do not have good quality code between StartDate and EndDate

AnnualRainfall Mean annual rainfall in mm

Latitude Latitude

Longitude Longitude

Examples

```
## Not run:  
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")  
d<-SILOSiteSummary(X)
```

```
## End(Not run)
```

SILOThiessenShp	<i>Function to generate Thiessen polygons from SILO sites</i>
-----------------	---

Description

Function to generate Thiessen polygons from SILO sites

Usage

```
SILOThiessenShp(SILOdata, path, shpname)
```

Arguments

SILOdata	- data loaded from SILO based on site list
path	- file path to save Thiessen polygon shapefile
shpname	- filename to save ESRI shapefile (no extension)

Value

A simple feature geometry (sf::sfc object) of the polygons created. Shape file saved to path \ shp-name

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path=". /SWTools/extdata")
p<-SILOThiessenShp(X, tempdir(), "Theissens")
a<-SILOSiteSummary(X)
ggplot(p)+geom_sf(aes(fill=AnnualRainfall))+
geom_point(data=a, aes(Longitude, Latitude))+
geom_text(data=a, aes(Longitude, Latitude, label=Site), nudge_y = 0.02)

## End(Not run)
```

SILOWriteforSource	<i>Write a SILO time series to a csv file in the format expected by eWater Source</i>
--------------------	---

Description

Write a SILO time series to a csv file in the format expected by eWater Source

Usage

```
SILOWriteforSource(SILO, col, filename, scalefactor = 1)
```

Arguments

SILO	a list of sites with SILO data, as created by SILOLoad()
col	Name of a column in a silo file to write out, e.g. Rain
filename	file to write to.
scalefactor	factor to scale the data by. Defaults to 1. Useful for Pan evap or rainfall scaling. Could also be a vector, with a value for each station in SILO

Value

Nothing to the R environment. SILO data is written to "filename".

Examples

```
## Not run:
X<-SILOLoad(c("24001", "24002", "24003"), path="./SWTools/extdata")
SILOwriteforSource(X, "Rain", tempfile("Rainfall", fileext=".csv"))

## End(Not run)
```

SWTools

SWTools: Helper Tools for Australian Hydrologists

Description

Functions to speed up workflow for hydrological analysis. Focused on Australian climate data (SILO climate data), hydrological models (eWater Source) and South Australian hydrological data (from [Water Data SA](#)).

SILO functions

SILO is a database of Australian climate data from 1889 to the present. It provides daily meteorological datasets for a range of climate variables in ready-to-use formats suitable for biophysical modelling, research and climate applications [SILO Website](#).

These functions allow SILO data to be downloaded from the [SILO Website](#), imported into R, calculate some basic statistics and undertake some quality assurance tests to easily visualise how much data has been interpolated, and to compare nearby sites to identify potential data issues. [SILODownload](#), [SILOLoad](#) and [SILOReport](#) functions allow a vector of SILO sites to be downloaded and summarised in a Microsoft Word report.

Source and Veneer functions

eWater Source is the Australia's national hydrological modelling platform, and is increasing in use around the world. Functions are included to write SILO climate data to the format expected for Source [SILOWriteforSource](#), and reading in model outputs, [read_res.csv](#).

Veneer is a RESTful API for interacting with Source models. Functions are included that are wrappers for Veneer, to build URLs to get or set data in the Source model, and process the json object returned.

Aquarius functions

South Australia's hydrological data is hosted on [Water Data SA](#). The [Export link](#) creates URLs that enable multiple datasets to be downloaded. [AQWPDownload](#) builds these URLs to download data in json format, and [AQWPLoad](#) loads this json file into the R interface.

VeneerGetInputSets	<i>Get vector of InputSets</i>
--------------------	--------------------------------

Description

Get vector of InputSets

Usage

```
VeneerGetInputSets(baseUrl = "http://localhost:9876")
```

Arguments

baseUrl URL of the Veneer server. Defaults to the veneer default.

Value

vector containing info on Input Sets in the model

Examples

```
## Not run:  
VeneerGetInputSets()  
  
## End(Not run)
```

VeneerGetNodesbyType *Get a vector of node names for a given type*

Description

Get a vector of node names for a given type

Usage

```
VeneerGetNodesbyType(NodeType, baseURL = "http://localhost:9876")
```

Arguments

NodeType	The node to return the names of. The icon in /network is searched for this name
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

vector of node names matching the specified node type

Examples

```
## Not run:
VeneerGetNodesbyType("Weir")

## End(Not run)
```

VeneerGetPiecewise *Get data from a Source piecewise table using Veneer*

Description

Get data from a Source piecewise table using Veneer

Usage

```
VeneerGetPiecewise(pw_table, baseURL = "http://localhost:9876")
```

Arguments

pw_table	The name of the piecewise linear variable, without the \$
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

a matrix with the data from the piecewise table.

Examples

```
## Not run:
VeneerGetPiecewise(data,"pw_table")

## End(Not run)
```

VeneerGetTS

Get a time series result from Source using Veneer

Description

Get a time series result from Source using Veneer

Usage

```
VeneerGetTS(TSURL, baseURL = "http://localhost:9876")
```

Arguments

TSURL, the URL of the time series to retrieve
baseURL URL of the Veneer server. Defaults to the veneer default.

Value

a zoo time series of the data

The URL of the time series must be specified, by interrogation using a browser or other analysis. By default Source returns SI units. Some conversion is undertaken: * Flow converted to ML/d * Volume converted to ML * Area converted to ha

Spaces are OK, like in the example below (dont need to insert

Examples

```
## Not run:
VeneerGetTS("/runs/latest/location/EndofSystem/element/Downstream Flow/variable/Flow")

## End(Not run)
```

VeneerGetTSbyNode *Get all time series recorded in Source for a given node*

Description

Get all time series recorded in Source for a given node

Usage

```
VeneerGetTSbyNode(Node, run = "latest", baseURL = "http://localhost:9876")
```

Arguments

Node	Name of node to retrieve Time Series for
run	Which run to retrieve from. Defaults to the latest
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

a zoo time series, with each variable as a column

Examples

```
## Not run:
VeneerGetTSbyNode("Storage 1")

## End(Not run)
```

VeneerGetTSbyVariable *Get all time series recorded in Source of a given variable type*

Description

Get all time series recorded in Source of a given variable type

Usage

```
VeneerGetTSbyVariable(
  variable = "Flow",
  run = "latest",
  baseURL = "http://localhost:9876"
)
```

Arguments

variable	Which variable to retrieve. Defaults to Flow.
run	Which run to retrieve from. Defaults to the latest
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

a zoo time series, with each output as a column

Examples

```
## Not run:
VeneerGetTSbyVariable() #returns all flow outputs recorded in the latest run
VeneerGetTSbyVariable("Water Surface Elevation",1)

## End(Not run)
```

VeneerGetTSVariables *Get a vector of the type of time series variables recorded*

Description

Get a vector of the type of time series variables recorded

Usage

```
VeneerGetTSVariables(run = "latest", baseURL = "http://localhost:9876")
```

Arguments

run	Which run to retrieve from. Defaults to the latest
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

a vector of variable types (e.g. Downstream flow, Downstream Flow Concentration, water surface elevation)

Examples

```
## Not run:
VeneerGetTSVariables()

## End(Not run)
```

VeneerlatestRunNumber *Get the number of the latest run*

Description

Get the number of the latest run

Usage

```
VeneerlatestRunNumber(baseUrl = "http://localhost:9876")
```

Arguments

baseUrl URL of the Veneer server. Defaults to the veneer default.

Value

integer of the latest run number

Examples

```
## Not run:  
VeneerlatestRunNumber()  
  
## End(Not run)
```

VeneerRunSource *Run Source using Veneer*

Description

Run Source using Veneer

Usage

```
VeneerRunSource(  
  StartDate = NULL,  
  EndDate = NULL,  
  InputSet = NULL,  
  baseUrl = "http://localhost:9876"  
)
```

Arguments

StartDate	Optional. Start date for simulation. Must be dd/mm/yyyy
EndDate	Optional. End date for simulation. Must be dd/mm/yyyy
InputSet	Optional. Input set to use
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

Nothing to the R environment.

If not set, the configuration parameters (StartDate, EndDate, InputSet), was is specified in the Source configuration in the GUI will be used.

The console will show any errors returned by Veneer.

Examples

```
## Not run:
VeneerRunSource()
VeneerRunSource("01/07/2017", "01/02/2018", "NoDams")

## End(Not run)
```

VeneerSetFunction	<i>Update a function value or expression. Function must exist before being updated.</i>
-------------------	---

Description

Update a function value or expression. Function must exist before being updated.

Usage

```
VeneerSetFunction(Name, Expression, baseURL = "http://localhost:9876")
```

Arguments

Name	Name of the function without the "\$", e.g. f_ScaleFactor
Expression	Expression to change it to, e.g. 1.2
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

Nothing to the R environment.

Examples

```
## Not run:
VeneerSetFunction("f_ScaleFactor",1.2)
VeneerSetFunction("f_TargetLevel","if($m_Flow<1000,3.2,3.5)")

## End(Not run)
```

VeneerSetPiecewise *Change a Source piecewise table using Veneer*

Description

Change a Source piecewise table using Veneer

Usage

```
VeneerSetPiecewise(data, pw_table, baseURL = "http://localhost:9876")
```

Arguments

data	A 2 column data.frame or matrix with the data to load into the piecewise table.
pw_table	The name of the piecewise linear variable, without the "\$".
baseURL	URL of the Veneer server. Defaults to the veneer default.

Value

Nothing to the R environment.

Examples

```
## Not run:
data<-data.frame(X=seq(1,5),Y=seq(1,5))
VeneerSetPiecewise(data,"pw_table")

## End(Not run)
```

WritepwtoIS*Write an input set line for a piecewise lookup table from a csv file*

Description

Write an input set line for a piecewise lookup table from a csv file

Usage

```
WritepwtoIS(folder, csvfiles, outputfile)
```

Arguments

folder	Folder for where are the csv files with the lookup tables
csvfiles	vector of files to turn into an input set line. File name should be the name of the pw table in Source, including the folder name if necessary, separated by "." (see example). The first row in the file should be column names, the same as used in Source, i.e. XValue and YValue
outputfile	text file to save the lines to

Value

Nothing to the R environment. Input set lines are written to "outputfile".

Examples

```
## Not run:  
folder<-"C:/Source/tables"  
csvfiles<-c("LowerLakes0ps.pw_LakeTarget.csv", "Operations.pw_NA_Lock5_16p8.csv")  
outputfile<-"inputset.txt"  
WritepwtoIS(folder, csvfiles, outputfile)  
  
## End(Not run)
```

Index

AQWPDDownload, [2](#), [17](#)

AQWPLoad, [4](#), [17](#)

read_res.csv, [4](#), [17](#)

SILOCumulativeDeviation, [5](#)

SILODoubleMass, [6](#)

SILODownload, [7](#), [16](#)

SILOImport, [8](#)

SILOLoad, [9](#), [16](#)

SILOMap, [10](#)

SILOMonthlyRainfall, [10](#)

SILOMortonQualityCodes, [11](#)

SILOQualityCodes, [12](#)

SILOReport, [13](#), [16](#)

SILOSiteSummary, [14](#)

SILOThiessenShp, [15](#)

SILOWriteforSource, [15](#), [17](#)

SWTools, [16](#)

VeneerGetInputSets, [17](#)

VeneerGetNodesbyType, [18](#)

VeneerGetPiecewise, [18](#)

VeneerGetTS, [19](#)

VeneerGetTSbyNode, [20](#)

VeneerGetTSbyVariable, [20](#)

VeneerGetTSVariables, [21](#)

VeneerlatestRunNumber, [22](#)

VeneerRunSource, [22](#)

VeneerSetFunction, [23](#)

VeneerSetPiecewise, [24](#)

WritepwtoIS, [25](#)