WeaAna Vignette

Bangyou ZHENG (Bangyou.Zheng@csiro.au)

April 15, 2011

Abstract

The purpose of the WeaAna Vignette is to show the basic methods how to run WeaAna. Please find the help pages and manuals for detailed information.

WeaAna is a R package to analysis weather records, easily to show analysis results, and to access statistics results.

Contents

1 Features

1. Multiple weather stations manage and analysis.
2. Auto load weather records when they are needed to save memory.
3. Handy functions to calculate extra daily weather variables and analysis weather conditions.

2 Read weather records

2.1 readWeatherRecords

Currently, WeaAna package supports weather records files with APSIM format with extension “**met**” or “**MET**” and “**RDATA**”. For “**RDATA**” format. Please see demo files for the detailed format about “**met**” files. RData files are saved variables which class are WeaAna class to save storing space and loading speed. One RData file only could contain the weather records of one station. Other results and other information are omitted. Function “**Met2RData**” would be the best way to convert met file to RData file. Other formats would be gradually added if it is necessary.

Function “**readWeatherRecords**” is used to read weather records from files. For example:

```
records <- readWeatherRecords( dataFolders = c( "MetFiles" ),
                               dataFiles = c( "MetFiles/10124.met" ) )
records <- readWeatherRecords( dataFolders = "MetFiles",
                               dataFiles = c( "MetFiles/10124.RData" ),
                               dataFormat = "RDATA" )
```

The first two lines read weather records from APSIM met files and the second two lines from RData files. Weather record files could be specified by parameters “**dataFolders**” and “**dataFiles**”, which could be vectors to list all files or folders. The duplicate files are omitted. Function “**readWeatherRecords**” is return a S4 class WeaAna which contained all weather records and other information. All other functions are based on the S4 class WeaAna.

2.2 Load weather records later

If many weather stations are analyzed at the same time, your computer would be out of memory. “**weaana**” package is designed to deal with this problem. All weather records will not be really loaded when “**readWeatherRecords**” is called with “**load.later = TRUE**”. Weather records will be loaded when some functions need them, then these records will be removed from memory except station name, number, latitude, and longitude. So, all station information are available when some statistics or plot functions are called. Function “**siteInfor**” with argument “**load.now = TRUE**” will read site basic information and store them into WeaAna class.

```
records <- readWeatherRecords( dataFolders = c( "MetFiles" ), load.later = TRUE )
# Collect information of weather stations
siteInfor( records, load.now = TRUE )
```

2.3 Extra marker

Some extra markers could be added to identify each site. Function “**addMarkers**” and “**delMarkers**” are used to add and delete markers, respectively.

```
addMarkers( records, model = c( "SILO", "A2", "B2" ) )
delMarkers( records, name = "model" )
```

3 Access information of WeaAna class

S4 class WeaAna are used to store information of multiple sites. The real weather records and statistics results are stored another S4 class WeaAnaSite which can not access by user. A pointer of WeaAnaSite is stored in WeaAna variable for each site to reduce memory usage. Consequently, any change of the a subset of WeaAna variable will be stored and reloaded after saving to RData file with R function “**save**”.

The operator [is used to get a subset of WeaAna class. A numeric vector is needed to specify positions. Examples as follows.

```
new.record <- records[1]
new.record <- records[1:10]
new.record <- records[c(1,3,10)]
```

Function "**print**", "**show**" or just input variable of WeaAna class show basic information of all weather stations (**name, number, latitude and longitude**). The daily weather records are skipped.

```
                 Name Number Latitude Longitude
1           NANGEENAN 010124   -31.51    118.18
2             NYABING 010619   -33.54    118.15
3 SCADDAN POST OFFICE 012073   -33.44    121.72
4              RUDALL 018075   -33.69    136.27
```

Function "**siteInfor**" is returned the same result, but is a data frame which have four columns for **Name, Number, Latitude and Longitude**.

```
siteInfor( records )
```

All statistics results and figures are stored at S4 class **WeaAnaSite** when calculating and plotting. Each result could be accessed through function "**getResults**" with result name. All available results can be listed through function **resultsInfor**.

```
resultsInfor( records )
getResults( records, "firstHeatDay" )
```

This is an example result for first heat day.

```
                 Name Number Latitude Longitude firstHeatDay
1           NANGEENAN 010124   -31.51    118.18        304.7
2             NYABING 010619   -33.54    118.15        319.0
3 SCADDAN POST OFFICE 012073   -33.44    121.72        308.0
4              RUDALL 018075   -33.69    136.27        296.4
```

4 Station position information

Function "**sites2KML**" convert site positions to KML file. The station names could not be output with "**name** = **FALSE**".

```
sites2KML( records, file = "sites.kml", name = FALSE )
```

Function "**plotSites**" shows positions of all weather stations in a map (Figure 4.1).

```
plotSites( records, lines = ozRegion(), sitename = FALSE, col = "red", pch = 24 )
```

The default map is Australia map from oz package. The map can be specified by parameter "**lines**" which is a list of latitude and longitude of all lines. The site name, colour and symbol style could be set by other parameters.

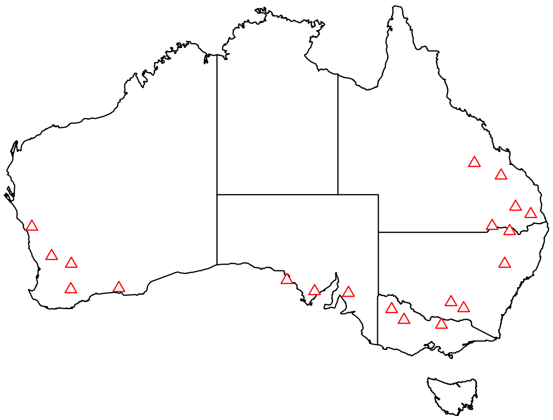

Figure 4.1: Example to show weather station positions with function “**plotSites**”

5 Parameters setting

WeaAna package uses global parameters setting to share many parameters among functions. All parameters and default values are listed by Appendix A.

All stored parameters could be accessed through function “**getPara**”. All parameters would be returned in default way, but one specific parameter could be returned if argument “name” is specified.

```
getPara() # All parameters will be returned
getPara( "data.format" ) # Parameter "dataFormat" will be returned
```

Function “**setPara**” could be used to set several parameters at the same time. Please pay attention to parameter value, as validity of new value is not checked.

```
setPara( data.format = "APSIM", mov.window = 10 )
```

Functions related with weather analysis have arguments to specify function parameters. But these parameters are not stored to global parameters. Function “setPara” must be called to plot figures if some specific parameters are used as all plot functions don’t have arguments to set parameters.

6 Weather analysis

6.1 Add weather variables

Function “**wcal**” is a flexible function to calculate weather variables through functions or string formula. This function could be used to calculate extra variables according weather records and knowing variables. Function “**getAvaiVars**” could be used to obtain all knowing variables.

Function “**wcal**” is string formula mode if FUN is NULL. String formula which contains variable name would be explained for each site. The argument name will be stored as variable names. The following codes are used to calculate average temperature.

```
library(weaana)
data( "WeatherRecordsDemo" )
# Daily mean temperature
wcal( records, avgt = "(␣maxt␣+␣mint␣)␣/␣2" )
```

If FUN is specified, function "**wcal**" is function mode. FUN will be called with args and optional arguments. The following codes are used to calculate moving average temperature with function **mov.avg**.

```r
library(weaana)
data( "WeatherRecordsDemo" )
# Moving average temperature
wcal( records, FUN = mov.avg, args = "avgt", k = 5, shift = "begin", var.name = "mov.avg" )
# The codes get the same results
wcal( records, mov.avg = "mov.avg(␣avgt,␣k␣=␣5,␣shift␣=␣\"begin\"␣)" )
```

If "**load.later**" is TRUE when **readWeatherRecords** is called, "**wcal**" will not really be executed. These functions will be evaluated when weather records are needed.

6.2 Analysis weather variables

Function "**wapply**" is a powerful function to statistics weather records by a certain period and functions. Most of other weather analysis functions are based on wapply. This are some simple examples to use "**wapply**":

```r
library(weaana)
data( "WeatherRecordsDemo" )
# daily mean temperature
wapply( records, vars = "avgt", period = "day", FUN = mean )
# yealy total rainfall
wapply( records, vars = "rain", period = "year", FUN = sum )
# yearly mean temperature
wapply( records, vars = "avgt", period = "year", FUN = mean )
# monthly mean temperature
wapply( records, vars = "avgt", period = "month", FUN = mean )
# 10 days mean temperature
wapply( records, vars = "avgt", period = 10, FUN = mean )
```

Many variables could be calculated with different functions at the same time. **FUN** will be replicated if length of **FUN** is less than that of **vars**. The following codes are used to calculate yearly average temperature, total rainfall and average radiation at one call.

```r
library(weaana)
data( "WeatherRecordsDemo" )
# yearly mean temperature, total raifall and mean radiation.
wapply( records, vars = c( "avgt", "rain", "radn" ),
                period = "year",
                FUN = c( mean, sum ),
                res.name = c( "mean.avgt", "total.rain", "mean.radn" ) )
```

Difference parameters could be set for each level if FUN need more optional parameters. The following codes are used to calculate thermal time from last frost day to first heat day from 1960 to 2009.

6.3 Climatic extreme

7 Plot results

Function "**wplot**" is a convenient way to plot all figures at the same time.

```r
library(weaana)
data( "WeatherRecordsDemo" )
wplot( records )
```

Function “**setPara**” must be called if some specific parameters are used as all plot functions don’t have arguments to set parameters. However, it is unnecessary to call each weather analysis function before any plot functions are called. These weather analysis functions will be automatically called by plot functions.

A Default parameters of weaana package

Name	Value	Description
data.format	APSIM	The data fromat of weather records
load.later	FALSE	Whether load weather records later when they are needed
yrange	c(1799, this year)	The year range to analysis. Default value is wide enough to include all range
records.index	c("year", "day")	The index for each record in the weather record files
records.vars	c("radn", "maxt", "mint", "rain", "evap", "vp")	The variables for each record in the weather record files
base.temperature	0	The base temperature to calculate the degree days
key.degree.days	400	The key degree days to calculate the extreme temperature in this periods
day.begin	1	Days start to calculate
day.end	365	Days end to calculate
extreme		