

Package ‘apsimr’

October 12, 2015

Type Package

Title Edit, Run and Evaluate APSIM Simulations Easily Using R

Version 1.2

Date 2015-10-09

Author Bryan Stanfill

Maintainer Bryan Stanfill <bstanfill2003@gmail.com>

Description The Agricultural Production Systems sIMulator (APSIM) is a widely used simulator of agricultural systems. This package includes functions to create, edit and run APSIM simulations from R. It also includes functions to visualize the results of an APSIM simulation and perform sensitivity/uncertainty analysis of APSIM either via functions in the sensitivity package or by novel emulator-based functions. For more on APSIM including download instructions go to [\{\}url{www.apsim.info}](http://www.apsim.info).

License GPL (>= 3)

Imports lubridate, reshape2, MASS, mgcv

Depends R (>= 3.0), ggplot2, XML

Suggests knitr

VignetteBuilder knitr

BugReports <https://github.com/stanfill/apsimr/issues>

Collate 'editAPSIM.R' 'helperFns.R' 'runAPSIM.R' 'sensitivityAPSIM.R'
'visualize.R' 'emulateAPSIM.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-12 08:58:48

R topics documented:

apsimr-package	2
apsim	3

apsim_emul_sa	4
apsim_vector	5
edit_apsim	7
edit_sim_file	8
example_apsim	9
plot.apsim	10
plot.gamSA	11
print.gamSA	12

Index	13
--------------	-----------

apsimr-package	<i>Run, edit, visualize APSIM from R</i>
----------------	------------------------------------------

Description

The **Agricultural Production Systems sIMulator** (APSIM) is a widely used simulator of agricultural systems. This package includes functions to create, edit and run APSIM simulations from R. It also includes functions to visualize the results of an APSIM simulation and perform sensitivity/uncertainty analysis of APSIM either via functions in the `sensitivity` package or by novel emulator-based functions. An installation of APSIM is required for this package to be of any use. Assuming non-commercial use, APSIM can be downloaded for free from <http://www.apsim.info/Products/Downloads.aspx>.

Details

Package: apsimr
 Type: Package
 Version: 1.2
 Date: 2015-10-09
 License: GPL(>=3)

Author(s)

Bryan Stanfill
 Maintainer: Bryan Stanfill <bstanfill2003@gmail.com>

References

See <http://www.apsim.info/> for information about APSIM.

See Also

[APSIMBatch](#), [sensitivity](#)

Examples

```
## Not run:
apsimExe <- "C:/Program Files (x86)/Apsim76-r3376/Model/Apsim.exe"
apsimWd <- "~/APSIM"
apsimResults <- apsim(exe = apsimExe, wd = apsimWd, files = "Centro.apsim")
plot(apsimResults)

## End(Not run)
```

apsim

Run APSIM Simulations from R

Description

This function will run one or many APSIM simulations and read the output into R in the form of a list of data frames. If the simulation does not run for some reason then an error is returned.

Usage

```
apsim(exe, wd = getwd(), files = NULL)
```

Arguments

exe	path to the APSIM executable
wd	working directory containing the .apsim files to be run; defaults to the current working directory
files	.apsim files to be run; if left empty all .apsim files in wd will be run

Details

The only required input is the path to the APSIM executable (APSIM.exe) usually found in the "Model" subfolder of the APSIM installation. By default, it is assumed the current working directory contains the .apsim file(s) to be run. If that is not the case then the directory containing the .apsim file(s) to be run should be specified by the wd argument. One can specify a list of .apsim files to be run within the directory wd using the files argument. If the files argument is left blank then all .apsim files within the directory specified by wd are run. The results for each .apsim file is saved as a data frame which are compiled into a list. Each element of the list is of the class "apsim", which has its own print and plot routines.

Value

list of output files; each element of the list corresponds to an output file specified by the .apsim files executed

Examples

```
## Not run:
apsimExe <-"C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"
apsimWd <- "~/APSIM"
toRun <- c("Centro.apsim", "Continuous Wheat.apsim")
results <- apsim(exe = apsimExe, wd = apsimWd, files = toRun)
results
plot(results$Centro)

## End(Not run)
```

apsim_emul_sa

Emulator-Based Sensitivity Analysis

Description

This is a generic function that can be used to estimate the sensitivity indices for complex computer models using GAM-based emulators.

Usage

```
apsim_emul_sa(model, X, boot = 1000, conf = 0.95, y = NULL, method, ...)
```

Arguments

model	a function that specifies the model of interest
X	n-by-p matrix of input values; n runs, p inputs
boot	number of bootstrap replicates
conf	confidence level of bootstrap calibrated intervals
y	optional vector of model evaluations that can be used in place of the model statement
method	method to use to emulate the model; "singleGAM" or "separateGAM"
...	additional arguments passed to method

Details

Each column of the matrix X corresponds to a different input and each row corresponds to a different run of the computer model. Currently the method choices are: "singleGAM" and "separateGAM". The "singleGAM" method builds a single generalized additive model (GAM) with terms for all main effects of inputs plus all two-way interactions between inputs. With this method all first-order and total sensitivity indices are estimable provided at least $8p^2 - 4p + 1$ runs of the computer model are available where p is the number of parameters included in the analysis. The "separateGAM" method builds a separate GAM emulator for each input to estimate first-order indices for each input. Since this method emulates the marginal distribution of y given each of the inputs, total order indices cannot be computed. Both methods can be bootstrapped in order to estimate standard errors and to calibrate confidence regions for the sensitivity index estimates. The argument boot specifies the number of bootstrap replicates and conf is the confidence level of the bootstrap calibrated confidence region.

Value

A data frame of sensitivity measures where the exact form depends on the method, see details.

Examples

```
## Not run:
meanYield<-function(x){
  return(mean(x$lai_cowpea))
}
n <- 75
parValues <- data.frame(SoilCN = runif(n, 5, 25),
                        DiffusConst = runif(n, 20, 50), CNCov = runif(n, 0, 1))
apsimExe <- "C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"
apsimWd <- "~/APSIM"
apsimVars <- c("SoilOrganicMatter/SoilCN", "SoilWater/DiffusConst", "SoilWater/CNCov")
apsimFile <- "Canopy.apsim"
emulRes <- apsim_emul_sa(model = apsim_vector, X = parValues, method = "singleGAM",
                        exe = apsimExe, wd = apsimWd, vars = apsimVars, to.run = apsimFile,
                        g = meanYield, overwrite = TRUE)

emulRes
plot(emulRes)

## End(Not run)
```

apsim_vector

*Edit and Run an APSIM Simulation***Description**

This function will edit then run an APSIM simulation. It can be thought of as a vectorized version of [apsim](#).

Usage

```
apsim_vector(X, exe, wd, vars, to.run, to.edit = to.run, overwrite = FALSE,
            g)
```

Arguments

X	N-by-p matrix of inputs with rows corresponding to runs and columns for variables
exe	where to find the APSIM executable
wd	directory where the .apsim file lives and where the results will be saved
vars	names of the variables, must be of the same length as 'X' has columns
to.run	the .apsim file in wd to run

to.edit	the .apsim file or .xml file to be edited
overwrite	logical; passed to the edit_apsim and edit_sim_file functions
g	a function of the output returned by apsim - must give vector or scalar result

Details

In order to link between the [sensitivity](#) package and APSIM, there needs to be a function that can edit then run APSIM that produces a univariate output. The `apsim_vector` function satisfies these conditions and can return multivariate output if a multivariate sensitivity analysis is of interest. See the examples section or the package vignette for details.

Value

a vector of length N

Examples

```
## Not run:
meanCowpea<-function(X){
  return(mean(X$lai_cowpea))
}

apsimWd <- "~/APSIM"
apsimVar <- c(rep("SoilWater/Thickness",11), "SoilOrganicMatter/SoilCN")
apsimValue <- matrix(c(rep(200, 2), rep(300, 9), 10,
                      rep(350, 2), rep(350, 9), 5),nrow=2,byrow=T)
apsimExe <- "C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"
apsimFile <- "Canopy.apsim"

#Run APSIM at each new parameter vector specified by apsimVar and only return the mean cowpea value
uniRes <- apsim_vector(X = apsimValue, exe = apsimExe, wd = apsimWd, vars = apsimVar,
                      to.run = apsimFile, to.edit = apsimFile, g = meanCowpea)
uniRes

allCowpea <- function(x){
  return(x$lai_cowpea)
}

#Use allCowpea function to return all the cowpea values
multiRes <- apsim_vector(X = apsimValue, exe = apsimExe, wd = apsimWd, vars = apsimVar,
                        to.run = apsimFile, to.edit = apsimFile, g = allCowpea)
multiRes

## End(Not run)
```

edit_apsim

Edit an APSIM Simulation

Description

This function allows you to edit an APSIM simulation file.

Usage

```
edit_apsim(file, wd = getwd(), var, value, overwrite = FALSE)
```

Arguments

file	file ending in .apsim to be edited
wd	directory containing the .apsim file to be edited; defaults to the current wd
var	vector of variables to be edited
value	list of new values for the specified variables
overwrite	logical; if TRUE the old file is overwritten, a new file is written otherwise

Details

The variables specified by `var` within the .apsim file specified by `file` in the working directory `wd` are edited. The old values are replaced with `value`, which is a list that has the same number of elements as the length of the vector `var`. The current .apsim file will be overwritten if `overwrite` is set to TRUE; otherwise the file *file-edited.apsim* will be created. If the file was successfully edited, then the name of the written file is returned.

Value

complete file path to edited .apsim file is returned as a character string

Examples

```
## Not run:
#The file I want to edit is called "Canopy.apsim" which is in the directory "~/APSIM"
apsimFile <- "Canopy.apsim"
apsimWd <- "~/APSIM"

#I want to change the Thickness of the Soilwater, the SoilCN of the SoilOrganicMatter and
#the state at which the simulation is being run.
apsimVar <- c("SoilWater/Thickness", "SoilOrganicMatter/SoilCN", "State")

#Change SoilWater-Thickness to 200,200,300x9
#Change SoilCN to 10
#Change "State" to "NSW"
apsimValue <- list(c(rep(200, 2), rep(300, 9)), 9, "NSW")

#Edit the apsim file without overwriting it
```

```

edit_apsim(file = apsimFile, wd = apsimWd, var = apsimVar, value = apsimValue, overwrite = FALSE)

#Run the edited simulation
apsimExe <- "C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"

results <- apsim(apsimExe, apsimWd, files = "Canopy-edited.apsim")

#Passing a simulation file to edit_apsim will give you a warning and redirect it to edit_sim_file
simFile <- "Soil.xml"
simValue <- list(abs(rnorm(1)), abs(rnorm(1)), c(0,2,2,1))
simVar <- c("nitrification_pot", "dnit_nitrif_loss", "wfnit_values")
edit_apsim(file = simFile, wd = apsimWd, var = simVar, value = simValue, overwrite = FALSE)

## End(Not run)

```

edit_sim_file	<i>Edit an APSIM Module File</i>
---------------	----------------------------------

Description

APSIM helper files, such as "Soil.xml", have a different format from .apsim files and are therefore handled separately

Usage

```
edit_sim_file(file, wd = getwd(), var, value, overwrite = FALSE)
```

Arguments

file	.xml module file to be edited
wd	directory containing the .xml file to be edited; defaults to the current wd
var	vector of variables to be edited
value	list of new values for the specified variables
overwrite	logical; if TRUE the old file is overwritten, otherwise a new file is written

Details

APSIM uses .xml files to dictate how certain processes are carried out. Similar to [edit_apsim](#) this function edits a file that will be used in an APSIM simulation. Unlike [edit_apsim](#) this function edits the .xml simulation files. The variables specified by var within the .xml file specified by file in the working directory wd are edited. The old values are replaced with value, which is a list that has the same number of elements as the vector var is long. The current .xml file will be overwritten if overwrite is set to TRUE; otherwise the file *file-edited.xml* will be created. If the file was successfully edited, then the name of the written file is returned.

Value

complete file path to edited simulation file is returned as a character string

Examples

```
## Not run:
#The file I want to edit is called "Soil.xml" which is the the directory "~/APSIM"
simFile <- "Soil.xml"
apsimWd <- "~/APSIM"

#I want to change the potential nitrification and N2O from nitrification
simVar <- c("nitrification_pot", "dnit_nitrif_loss", "wfnit_values")

#Change both to absolute values of random N(0,1)
simValue <- list(abs(rnorm(1)), abs(rnorm(1)), c(0,2,2,1))

#Edit Soil.xml without overwriting it
edit_sim_file(file = simFile, wd = apsimWd, var = simVar, value = simValue, overwrite = FALSE)

#Passing an .apsim file to edit_sim_file will give a warning and redirect it to edit_apsim
apsimFile <- "Canopy.apsim"
apsimValue <- list(c(rep(200, 2), rep(300, 9)), 9, "NSW")
apsimVar <- c("SoilWater/Thickness", "SoilOrganicMatter/SoilCN", "State")
edit_sim_file(file = apsimFile, wd = apsimWd, var = apsimVar, value = apsimValue, overwrite = FALSE)

## End(Not run)
```

example_apsim

Access Example APSIM Simulations

Description

Standard APSIM simulations are provided by the default APSIM installation. `apsim_example` moves those example files into the working directory `wd` so you can run them or edit them using `apsim` and `edit_apsim`, respectively. Generally the example simulations must be moved because the output file is written to the directory containing the `.apsim` file and the ability to write in the "Program Files" can be limited in some cases.

Usage

```
example_apsim(path, wd = getwd(), files = NULL, ...)
```

Arguments

<code>path</code>	path to the APSIM installation
<code>wd</code>	working directory containing the <code>.apsim</code> files to be copied; defaults to the current working directory
<code>files</code>	files to extract from the "Examples" folder
<code>...</code>	additional arguments passed to file.copy

Value

logical; if TRUE the corresponding file was successfully copied, FALSE otherwise

Examples

```
## Not run:
apsimPath <- "C:/Program Files (x86)/Apsim75-r3008/"
apsimWd <- "~/APSIM"
toRun <- "Canopy.apsim"
example_apsim(path = apsimPath, wd = apsimWd, files = toRun) #TRUE

toRun <- c("Canopy.apsim", "Continuous Wheat.apsim")
example_apsim(path = apsimPath, wd = apsimWd, files = toRun) #TRUE TRUE

apsimExe <- "C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"
results <- apsim(exe = apsimExe, wd = apsimWd, files = toRun)
plot(results[[1]])

## End(Not run)
```

plot.apsim

Visualize the Results of an APSIM Simulation

Description

Plot routine for "apsim" class objects

Usage

```
## S3 method for class 'apsim'
plot(x, y = NULL, ask = TRUE, one_plot = FALSE, ...)
```

Arguments

x	data frame of class "apsim" including the results of an APSIM simulation
y	variable to plot on y-axis; if left empty all variables will be plotted on separate plots
ask	logical; if TRUE, the user is asked before each plot, see par (ask=.)
one_plot	logical; if TRUE all variables are plotted on one faceted plot
...	additional arguments passed to qplot

Details

Similar to the [plot](#) for [lm](#) objects, `plot.apsim` will plot each response in the results of an APSIM simulation on its own `ggplot2` object. If the `one_plot` argument is set to TRUE then [facet_wrap](#) is used to plot all of the responses on one screen. Alternatively, one response can be plotted by setting `y` to one variable corresponding to a column in `x`.

Examples

```
## Not run:
apsimExe <- "C:/Program Files (x86)/Apsim75-r3008/Model/Apsim.exe"
apsimWd <- "~/APSIM"
toRun <- c("Centro.apsim", "Continuous Wheat.apsim")
results <- apsim(exe = apsimExe, wd = apsimWd, files = toRun)

#Look at all of the results as a function of time in separate plots
plot(results[[2]])

#Put all variables on one faceted plot
plot(results[[2]], one_plot = TRUE) + theme_bw()

#Plot just yield as a function of time
plot(results[[2]], y = 'yield') + geom_line(colour = 'red') + theme_bw()

## End(Not run)
```

plot.gamSA

Plot Sensitivity Analysis Results

Description

Plot routine for "gamSA" class objects

Usage

```
## S3 method for class 'gamSA'
plot(x, ...)
```

Arguments

x, ... Results of a successful call to apsim_emulator

Details

The default plot method for the results of a sensitivity analysis of APSIM using the single or separate GAM-based emulator. This function produces a bar chart where the heights of the bar represents the magnitude of the estimated sensitivity index estimate. Error bars represent bootstrap calibrated confidence regions.

Value

The results of the sensitivity analysis without the estimated output and residuals

`print.gamSA`*Print Sensitivity Analysis Results*

Description

The default print method for the results of a sensitivity analysis of APSIM using the single or separate GAM-based emulator.

Usage

```
## S3 method for class 'gamSA'  
print(x, ...)
```

Arguments

`x, ...` Results of a successful call to [apsim_emul_sa](#)

Value

The results of the sensitivity analysis without the estimated output and residuals

Index

*Topic **package**

- apsimr-package, [2](#)

- apsim, [3](#), [5](#), [9](#)
- apsim_emul_sa, [4](#), [12](#)
- apsim_vector, [5](#)
- APSIMBatch, [2](#)
- apsimr (apsimr-package), [2](#)
- apsimr-package, [2](#)

- edit_apsim, [7](#), [8](#), [9](#)
- edit_sim_file, [8](#)
- example_apsim, [9](#)

- facet_wrap, [10](#)
- file.copy, [9](#)

- lm, [10](#)

- par, [10](#)
- plot, [10](#)
- plot.apsim, [10](#)
- plot.gamSA, [11](#)
- print.gamSA, [12](#)

- qqplot, [10](#)

- sensitivity, [2](#), [6](#)