

Package 'ProjectTemplate'

May 13, 2012

Type Package

Title Automates the creation of new statistical analysis projects.

Version 0.4-2

Date 2012-05-12

Author John Myles White

Maintainer John Myles White <jmw@johnmyleswhite.com>

Description ProjectTemplate provides functions to automatically build a directory structure for a new R project. Using this structure, ProjectTemplate automates data loading, preprocessing, library importing and unit testing.

License Artistic-2.0

LazyLoad yes

Depends R (>= 2.7), testthat

Suggests foreign, reshape, plyr, stringr, ggplot2, lubridate, log4r, RMySQL, RSQLite, gdata, RODBC, tuneR, pixmap, data.table, RPostgreSQL

URL <http://projecttemplate.net>

Collate

'arff.reader.R' 'cache.name.R' 'cache.project.R' 'cache.R' 'clean.variable.name.R' 'create.project.R' 'csv.reader.R' 'db.read

Repository CRAN

Date/Publication 2012-05-13 08:35:22

R topics documented:

arff.reader	3
cache	3
cache.name	4
cache.project	5
clean.variable.name	5
create.project	6
csv.reader	6
db.reader	7
dbf.reader	8
epiinfo.reader	8
extensions.dispatch.table	9
file.reader	9
load.project	10
mp3.reader	10
mtp.reader	11
octave.reader	12
ppm.reader	12
ProjectTemplate	13
r.reader	13
rdata.reader	14
reload.project	15
run.project	15
show.project	16
spss.reader	16
sql.reader	17
stata.reader	18
stub.tests	18
systat.reader	19
test.project	19
translate.dcf	20
tsv.reader	20
url.reader	21
wsv.reader	22
xls.reader	22
xlsx.reader	23
xport.reader	24

arff.reader	<i>Read the Weka file format.</i>
-------------	-----------------------------------

Description

This function will load a data set stored in the Weka file format into the specified global variable binding.

Usage

```
arff.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
## Not run: arff.reader('example.arff', 'data/example.arff', 'example')
```

cache	<i>Cache a data set for faster loading.</i>
-------	---

Description

This function will store a copy of the named data set in the cache directory. This cached copy of the data set will then be given precedence at load time when calling [load.project](#). Cached data sets are stored as .RData files.

Usage

```
cache(variable)
```

Arguments

variable	A character vector containing the name of the variable to be saved.
----------	---

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')
## Not run: create.project('tmp-project')

setwd('tmp-project')

dataset1 <- 1:5
cache('dataset1')

setwd('.')
unlink('tmp-project')
## End(Not run)
```

cache.name	<i>Translate a variable name into a file name for caching.</i>
------------	--

Description

This function will translate a variable name into a form that is suitable as a filename on most OS's.

Usage

```
cache.name(data.filename)
```

Arguments

data.filename The variable name to be translated into a filename.

Value

A translated variable name.

Examples

```
library('ProjectTemplate')

## Not run: cache.name('example.1')
```

cache.project	<i>Cache a project's data sets in binary format.</i>
---------------	--

Description

This function will cache all of the data sets that were loaded by the `load.project` function in a binary format that is easier to load quickly. This is particularly useful for data sets that you've modified during a slow munging process that does not need to be repeated.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')
## Not run: load.project()

cache.project()
## End(Not run)
```

clean.variable.name	<i>Translate a file name into a valid R variable name.</i>
---------------------	--

Description

This function will translate a file name into a name that is a valid variable name in R. Non-alphabetic characters on the boundaries of the file name will be stripped; non-alphabetic characters inside of the file name will be replaced with dots.

Usage

```
clean.variable.name(variable.name)
```

Arguments

`variable.name` A character vector containing a variable's proposed name that should be standardized.

Value

A translated variable name.

Examples

```
library('ProjectTemplate')

## Not run: clean.variable.name('example_1')
```

`create.project` *Create a new project.*

Description

This function will create all of the scaffolding for a new project. It will set up all of the relevant directories and their initial contents. For those who only want the minimal functionality, the `minimal` argument can be set to `TRUE` to create a subset of ProjectTemplate's default directories. For those who want to dump all of ProjectTemplate's functionality into a directory for extensive customization, the `dump` argument can be set to `TRUE`.

Usage

```
create.project(project.name = "new-project", minimal =
FALSE, dump = FALSE)
```

Arguments

<code>project.name</code>	A character vector containing the name for this new project. Must be a valid directory name for your file system.
<code>minimal</code>	A boolean value indicating whether to create a minimal project or a full project. A minimal project contains only the directories strictly necessary to use ProjectTemplate and does not provide template code for profiling, unit testing or documenting your project.
<code>dump</code>	A boolean value indicating whether the entire functionality of ProjectTemplate should be written out to flat files in the current project.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: create.project('MyProject')
```

`csv.reader` *Read a comma separated values (.csv) file.*

Description

This function will load a data set stored in the CSV file format into the specified global variable binding.

Usage

```
csv.reader(data.file, filename, variable.name)
```

Arguments

data.file The name of the data file to be read.
filename The path to the data set to be loaded.
variable.name The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: csv.reader('example.csv', 'data/example.csv', 'example')
```

db.reader	<i>Read a SQLite3 database with a (.db) file extension.</i>
-----------	---

Description

This function will load all of the data sets stored in the SQLite3 database into the global environment. If you want to specify a single table or query to execute against the database, move it elsewhere and use a .sql file interpreted by [sql.reader](#).

Usage

```
db.reader(data.file, filename, variable.name)
```

Arguments

data.file The name of the data file to be read.
filename The path to the data set to be loaded.
variable.name The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: db.reader('example.db', 'data/example.db', 'example')
```

dbf.reader *Read an XBASE file with a .dbf file extension.*

Description

This function will load all of the data sets stored in the specified XBASE file into the global environment.

Usage

```
dbf.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: dbf.reader('example.dbf', 'data/example.dbf', 'example')
```

epiinfo.reader *Read an Epi Info file with a .rec file extension.*

Description

This function will load all of the data sets stored in the specified Epi Info file into the global environment.

Usage

```
epiinfo.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: epiinfo.reader('example.rec', 'data/example.rec', 'example')
```

extensions.dispatch.table

Maps file types to the reader functions used to autoload them.

Description

This list stores a mapping from regular expressions that match file extensions for the file types supported by ProjectTemplate to the reader functions that implement autoloading for those formats. Any new file type must be appended to this dispatch table.

file.reader

Read an arbitrary file described in a .file file.

Description

This function will load all of the data sets described in the specified .file file into the global environment. A .file file must contain DCF that specifies the path to the data set and which extension should be used from the dispatch table to load the data set.

Usage

```
file.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Details

Examples of the DCF format and settings used in a .file file are shown below:

path: http://www.johnmyleswhite.com/ProjectTemplate/sample_data.csv extension: csv

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: file.reader('example.file', 'data/example.file', 'example')
```

load.project	<i>Automatically load data and packages for a project.</i>
--------------	--

Description

This function automatically load all of the data and packages used by the project from which it is called.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: load.project()
```

mp3.reader	<i>Read an MP3 file with a .mp3 file extension.</i>
------------	---

Description

This function will load the specified MP3 file into memory using the tuneR package. This is useful for working with music files as a data set.

Usage

```
mp3.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: mtp3.reader('example.mp3', 'data/example.mp3', 'example')
```

mtp.reader	<i>Read a Minitab Portable Worksheet with an .mtp3 file extension.</i>
------------	--

Description

This function will load the specified Minitab Portable Worksheet into memory.

Usage

```
mtp.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: mtp.reader('example.mtp', 'data/example.mtp', 'example')
```

octave.reader *Read an Octave file with a .m file extension.*

Description

This function will load the specified Octave file into memory.

Usage

```
octave.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: octave.reader('example.m', 'data/example.m', 'example')
```

ppm.reader *Read a PPM file with a .ppm file extension.*

Description

This function will load the specified PPM file into memory using the pixamp package. This is useful for working with image files as a data set.

Usage

```
ppm.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: ppm.reader('example.ppm', 'data/example.ppm', 'example')
```

ProjectTemplate	<i>Automates the creation of new statistical analysis projects.</i>
-----------------	---

Description

ProjectTemplate provides functions to automatically build a directory structure for a new R project. Using this structure, ProjectTemplate automates data loading, preprocessing, library importing and unit testing.

References

This code is inspired by the skeleton structure used by Ruby on Rails.

Examples

```
library('ProjectTemplate')

## Not run: create.project('project_name')

setwd('project_name')
load.project()
## End(Not run)
```

r.reader	<i>Read an R source file with a .R file extension.</i>
----------	--

Description

This function will call source on the specified R file, executing the code inside of it as a way of generating data sets dynamically, as in many Monte Carlo applications.

Usage

```
r.reader(data.file, filename, variable.name)
```

Arguments

`data.file` The name of the data file to be read.
`filename` The path to the data set to be loaded.
`variable.name` The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: r.reader('example.R', 'data/example.R', 'example')
```

rdata.reader *Read an RData file with a .rdata or .rda file extension.*

Description

This function will load the specified RData file into memory using the [load](#) function. This may generate many data sets simultaneously.

Usage

```
rdata.reader(data.file, filename, variable.name)
```

Arguments

`data.file` The name of the data file to be read.
`filename` The path to the data set to be loaded.
`variable.name` The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: rdata.reader('example.RData', 'data/example.RData', 'example')
```

reload.project	<i>Reload a project from scratch.</i>
----------------	---------------------------------------

Description

This function will clear the global environment and reload a project from scratch. This is useful when you've updated your data sets or changed your preprocessing scripts.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: load.project()  
  
reload.project()  
## End(Not run)
```

run.project	<i>Run all of the analyses in the src directory.</i>
-------------	--

Description

This function will run each of the analyses in the src directory in separate processes. At present, this is done serially, but future versions of this function will provide a means of running the analyses in parallel.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: run.project()
```

show.project	<i>Show information about the current project.</i>
--------------	--

Description

This function will show the user all of the information that ProjectTemplate has about the current project. This information is gathered when `load.project` is called. At present, ProjectTemplate keeps a record of the project's configuration settings, all packages that were loaded automatically and all of the data sets that were loaded automatically. The information about autoloading data sets is used by the `cache.project` function.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: load.project()  
  
show.project()  
## End(Not run)
```

spss.reader	<i>Read an SPSS file with a .sav file extension.</i>
-------------	--

Description

This function will load the specified SPSS file into memory. It will convert the resulting list object into a data frame before inserting the data set into the global environment.

Usage

```
spss.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: spss.reader('example.sav', 'data/example.sav', 'example')
```

sql.reader	<i>Read a database described in a .sql file.</i>
------------	--

Description

This function will load data from a SQL database based on configuration information found in the specified .sql file. The .sql file must specify a database to be accessed. All tables from the database, one specific tables or one specific query against any set of tables may be executed to generate a data set.

Usage

```
sql.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Details

Examples of the DCF format and settings used in a .sql file are shown below:

Example 1 type: mysql user: sample_user password: sample_password host: localhost dbname: sample_database table: sample_table

Example 2 type: sqlite dbname: /path/to/sample_database table: sample_table

Example 3 type: sqlite dbname: /path/to/sample_database query: SELECT * FROM users WHERE user_active == 1

Example 4 type: sqlite dbname: /path/to/sample_database table: *

Example 5 type: postgres user: sample_user password: sample_password host: localhost dbname: sample_database table: sample_table

Example 6 type: odbc dsn: sample_dsn user: sample_user password: sample_password dbname: sample_database query: SELECT * FROM sample_table

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: sql.reader('example.sql', 'data/example.sql', 'example')
```

stata.reader	<i>Read a Stata file with a .stata file extension.</i>
--------------	--

Description

This function will load the specified Stata file into memory.

Usage

```
stata.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: stata.reader('example.stata', 'data/example.stata', 'example')
```

stub.tests	<i>Generate unit tests for your helper functions.</i>
------------	---

Description

This function will parse all of the functions defined in files inside of the lib directory and will generate a trivial unit test for each function. The resulting tests are stored in the file tests/autogenerated.R. Every test is expected to fail by default, so you should edit them before calling [test.project](#).

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: stub.tests()
```

systat.reader	<i>Read a Systat file with a .sys or .syd file extension.</i>
---------------	---

Description

This function will load the specified Systat file into memory.

Usage

```
systat.reader(data.file, filename, variable.name)
```

Arguments

data.file	The name of the data file to be read.
filename	The path to the data set to be loaded.
variable.name	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: systat.reader('example.sys', 'data/example.sys', 'example')
```

test.project	<i>Run all unit tests for this project.</i>
--------------	---

Description

This function will run all of the testthat style unit tests for the current project that are defined inside of the tests directory. The tests will be run in the order defined by the filenames for the tests: it is recommend that each test begin with a number specifying its position in the sequence.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: load.project()

test.project()
## End(Not run)
```

translate.dcf	<i>Read a DCF file into an R list.</i>
---------------	--

Description

This function will read a DCF file and translate the resulting data frame into a list. The DCF format is used throughout ProjectTemplate for configuration settings and ad hoc file format specifications.

Usage

```
translate.dcf(filename)
```

Arguments

filename A character vector specifying the DCF file to be translated.

Value

Returns a list containing the entries from the DCF file.

Examples

```
library('ProjectTemplate')

## Not run: translate.dcf(file.path('config', 'global.dcf'))
```

tsv.reader	<i>Read a tab separated values (.tsv or .tab) file.</i>
------------	---

Description

This function will load a data set stored in the TSV file format into the specified global variable binding.

Usage

```
tsv.reader(data.file, filename, variable.name)
```

Arguments

`data.file` The name of the data file to be read.
`filename` The path to the data set to be loaded.
`variable.name` The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: tsv.reader('example.tsv', 'data/example.tsv', 'example')
```

url.reader	<i>Read a remote file described in a .url file.</i>
------------	---

Description

This function will load data from a remote source accessible through HTTP or FTP based on configuration information found in the specified .url file. The .url file must specify the URL of the remote data source and the type of data that is available remotely. Only one data source per .url file is supported currently.

Usage

```
url.reader(data.file, filename, variable.name)
```

Arguments

`data.file` The name of the data file to be read.
`filename` The path to the data set to be loaded.
`variable.name` The name to be assigned to in the global environment.

Details

Examples of the DCF format and settings used in a .url file are shown below:

Example 1 url: http://www.johnmyleswhite.com/ProjectTemplate/sample_data.csv separator: ,

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: url.reader('example.url', 'data/example.url', 'example')
```

wsv.reader *Read a whitespace separated values (.wsv or .txt) file.*

Description

This function will load a data set stored in the WSV file format into the specified global variable binding.

Usage

```
wsv.reader(data.file, filename, variable.name)
```

Arguments

data.file The name of the data file to be read.
filename The path to the data set to be loaded.
variable.name The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')

## Not run: wsv.reader('example.wsv', 'data/example.wsv', 'example')
```

xls.reader *Read an Excel 2004 file with a .xls file extension.*

Description

This function will load the specified Excel file into memory using the gdata package. Each sheet of the Excel workbook will be read into a separate variable in the global environment.

Usage

```
xls.reader(data.file, filename, workbook.name)
```

Arguments

<code>data.file</code>	The name of the data file to be read.
<code>filename</code>	The path to the data set to be loaded.
<code>workbook.name</code>	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: xls.reader('example.xls', 'data/example.xls', 'example')
```

<code>xlsx.reader</code>	<i>Read an Excel 2007 file with a .xlsx file extension.</i>
--------------------------	---

Description

This function will load the specified Excel file into memory using the xlsx package. Each sheet of the Excel workbook will be read into a separate variable in the global environment.

Usage

```
xlsx.reader(data.file, filename, workbook.name)
```

Arguments

<code>data.file</code>	The name of the data file to be read.
<code>filename</code>	The path to the data set to be loaded.
<code>workbook.name</code>	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: xlsx.reader('example.xlsx', 'data/example.xlsx', 'example')
```

`xport.reader`*Read an XPort file with a .xport file extension.*

Description

This function will load the specified XPort file into memory.

Usage

```
xport.reader(data.file, filename, variable.name)
```

Arguments

<code>data.file</code>	The name of the data file to be read.
<code>filename</code>	The path to the data set to be loaded.
<code>variable.name</code>	The name to be assigned to in the global environment.

Value

No value is returned; this function is called for its side effects.

Examples

```
library('ProjectTemplate')  
  
## Not run: xport.reader('example.xport', 'data/example.xport', 'example')
```

Index

arff.reader, 3

cache, 3
cache.name, 4
cache.project, 5, 16
clean.variable.name, 5
create.project, 6
csv.reader, 6

db.reader, 7
dbf.reader, 8

epiinfo.reader, 8
extensions.dispatch.table, 9

file.reader, 9

load, 14
load.project, 3, 5, 10, 16

mp3.reader, 10
mtp.reader, 11

octave.reader, 12

ppm.reader, 12
ProjectTemplate, 13

r.reader, 13
rdata.reader, 14
reload.project, 15
run.project, 15

show.project, 16
spss.reader, 16
sql.reader, 7, 17
stata.reader, 18
stub.tests, 18
systat.reader, 19

test.project, 18, 19
translate.dcf, 20

tsv.reader, 20

url.reader, 21

wsv.reader, 22

xls.reader, 22
xlsx.reader, 23
xport.reader, 24