The sinew gitbook

Metrum Research Group

2017-09-14

Contents

Preface

At the core of creating reproducible research is the daunting process of documentation and working with R presents no exception. The standardized R package documentation is part of the appeal for new (and experienced) users, where concise, clear instructions are cherished. Packages are great for reusing scripts in a consistent fashion and for developers, package creation helps organizing and maintaining their ideas. The barrier to entry for package maintenance and documentation, however, may be too high for R users embarking on the task of package creation, thereby discouraging them from development. The out-of-the-box roxygen2 skeleton supplied by RStudio gives the bare bones road map of what should be part of function documentation.

The goal of Sinew is to automate nearly all of the manual tasks needed to document functions, properly set up the import fields for oxygenation, and make it easier to attain documentation consistency across functions and packages.

Once the initial documentation is set, Sinew can be used as part of the natural maintenance workflow to automatically change and append updated parameters, definitions, defaults, and dependencies, making your documentation great - again!

Sinew is built for both command line and interactive users through a Shiny gadget that can be called from the addin menu. The addin can be used on any function either installed, in the search path, one you just wrote, or one your need to update.

Chapter 1

Installation

1.1 CRAN

The CRAN package homepage

```
install.packages('sinew')
```

For a quick example, run the following:

```
sinew::makeOxygen('lm')
```

1.2 DEV

The package is currently under continuous development for the most up to date version run the script below.

The github package homepage

```
devtools::install_github("metrumresearchgroup/sinew")
```

Chapter 2

tl;dr

Sinew can be used to solve a number of tasks with simple one liners

- Create import calls for DESCRIPTION, NAMESPACE and Roxygen2

```
makeImport('R',format = 'description')

makeImport('R',format = 'namespace')

makeImport('R',format = 'oxygen')
```

- Create skeleton for roxygen2 documentation populated with information scraped from within the package function scripts.

Function or data.frame objects, names

```
makeOxygen(lm)

makeOxygen('lm')

makeOxygen(stats::lm)

makeOxygen(iris,add_fields = c('source'))
```

Writing directly to file(s) in a directory (batch mode)

```
makeOxyFile(<pathtofiles>)
```

- Update existing roxygen2 documentation in a file

```
moga(<pathtofile.R>)
```

- Split a long R file with multiple function definitions into multiple single function files

```
untangle(<pathtofile.R>, dir.out=<path for output>)
```

- Create an R file of all the unique roxygen parameter fields in a package R subdirectory

```
#preview
makeDictionary('R')

#write to file man-roxygen subdirectory of package
makeDictionary('R',save_path = TRUE)
```

Chapter 3

Session level options

Sinew options can be set to remain persistent across sessions or at a global level within a session