`Xmisc::ArgumentParser` (0.2.1)

A Simple Command Line Argument Parser by *R*

Xiaobei Zhao*[1]

[1]Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill

Modified: 2014-08-12 Compiled: 2014-8-12

You may find the latest version of *Xmisc* [1] and this documentation at,
http://CRAN.R-project.org/package=Xmisc

Keywords: command line options/arguments parser

Contents

*Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill, 450 West Dr, Chapel Hill, NC 27599, USA. xiaobei@binf.ku.dk

1 Introduction

This document illustrates the major functionality (**Section** 2) of the parser, `Xmisc::ArgumentParser`, and demonstrates it by a minimal example (**Section** 3).

1.1 Package installation

```
>   ## install Xmisc
>   install.packages("Xmisc")
```

2 `ArgumentParser`

`ArgumentParser`, an *R* reference class, makes it easy to write user-friendly command-line interfaces. The class defines methods to add and parse arguments (`add_argument`), usgae (`add_usage`) and description (`add_description`).

2.1 Create an instance of `ArgumentParser`

```
>   require(Xmisc)
>   parser <- ArgumentParser$new()
```

2.2 Add arguments

```
> ## add a character object
>   parser$add_argument(
+     '--a_name',type='character',
+     help='A a_name.'
+   )
> ## add an integer object with default
>   parser$add_argument(
+     '--a_int',type='integer',
+     default=1,
+     help='A integer.'
+   )
> ## add a numeric object with default
>   parser$add_argument(
+     '--a_num',type='numeric',
+     default=1,
+     help='A number.'
+   )
> ## add a logical object with default
>   parser$add_argument(
+     '--if.test',type='logical',
+     default=FALSE,
+     help='Whether it is a test?!'
+   )
```

2.3 Add "usage"

```
>   parser$add_usage('Xmisc-ArgumentParser.R [options]')
```

2.4 Add "description"

```
>   parser$add_description(
+     'An executable R script parsing arguments from Unix-like command line.')
```

2.5 Add options to get the help page

```
>   parser$add_argument(
+     '--h',type='logical',
+     action='store_true',
+     help='Print the help page'
+   )
>   parser$add_argument(
+     '--help',type='logical',
+     action='store_true',
+     help='Print the help page'
+   )
```

2.6 Review pasrsed arguments

Method `get_args` returns all variables defined by the parser, with either the default values or those passed from the command line.

```
>   parser$get_args()


$a_name
character(0)

$a_int
[1] 1

$a_num
[1] 1

$if.test
[1] FALSE

$h
logical(0)

$help
logical(0)
```

2.7 Get help

Method `helpme` invokes a help page with -h or –help options.

```
>   parser$helpme()
```

3 A minimal example

An executable R script, `Xmisc-ArgumentParser.R` is provided at the `bin` subdirectory under the top level package directory. A complete source code of this executable is in Appendix A.1 on page 8.

3.1 Get the path of the executable R script

Given the *Xmisc* package is installed, we can obtain the path of the executable *R* script at the *R* prompt:

```
> system.file('bin', 'Xmisc-ArgumentParser.R', package='Xmisc', mustWork=TRUE)


[1] "/tmp/Rtmph6xwzv/Rinst88ed350f66a9/Xmisc/bin/Xmisc-ArgumentParser.R"


> ## Or,
> Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R')


[1] "/tmp/Rtmph6xwzv/Rinst88ed350f66a9/Xmisc/bin/Xmisc-ArgumentParser.R"
```

Alternatively, we can extract this path at Unix-like command-line interface (CLI):

```
cmd=$(Rscript -e "cat(system.file('bin','Xmisc-ArgumentParser.R',package='Xmisc', mustWork=TRUE))")
echo ${cmd}

## Or,
cmd=$(Rscript -e "cat(Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R'))")
echo ${cmd}
```

3.2 Get help

```
${cmd} -h

## Or,
${cmd} --help
```

This will print the help page at the console,

```
Usage:
    Xmisc-ArgumentParser.R [options]

Description:
    An executable R script parsing arguments from Unix-like command line
.

Options:
    h       logical   Print the help page. [ NULL ]

    help    logical   Print the help page. [ NULL ]

    a_name  character A a_name. [ Xmisc ]

    a_int   integer   A integer. [ 1 ]

    a_num   numeric   A number. [ 1 ]

    if.test logical   Whether it is a test?!. [ FALSE ]
```

3.3 Parsing command line arguments

Let's parse the command line arguments.

```
cmd=$(Rscript -e "cat(Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R'))")

${cmd} --a_name=${USER} --a_int=2 --a_num=3.6 --if.test=TRUE
```

This produces the output,

```
Hello, xiaobei!
The integer is 2.
The number is 3.6.

(This is a test).

class(a_name): character
class(a_int): integer
class(a_num): numeric
class(if.test): logical
```

3.3.1 Convert the passed values

Usually, the parser tries to convert the passed values to the proper type. For instance, it produces an `NA` when converting a character object to an integer.

```
cmd=$(Rscript -e "cat(Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R'))")
${cmd} --a_name=${USER} --a_int="an integer"
```

This produces the output,

```
Hello, xiaobei!
The integer is NA.
The number is 1.

(This is not a test).

class(a_name): character
class(a_int): integer
class(a_num): numeric
class(if.test): logical
```

3.3.2 Fail to convert the passed values

However, the parser fails in case it is unable to make such conversion. For instance, it fails to convert a character object to a logical.

```
cmd=$(Rscript -e "cat(Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R'))")
${cmd} --a_name=${USER} --if.test="a test"
```

This raises an error,

```
Error in withCallingHandlers(expr, warning = function(w) invokeRestart("muffleWarning")) :
  ValueParser | invalid logical value
Calls: main ... initialize -> initialize -> <Anonymous> -> R5.value.parse
Execution halted
```

3.3.3 Required variables

In the executable *R* script (Appendix A.1 on page 8), we have asked the variable `a_name` to be required.

```
> ## add a required character object
>   parser$add_argument(
+     '--a_name',type='character',
+     required=TRUE,
+     help='A a_name.'
+   )
```

In case any of the required variables is not present, the parser issues an error.

```
cmd=$(Rscript -e "cat(Xmisc::get_executable('Xmisc','Xmisc-ArgumentParser.R'))")
${cmd} --a_int=2 --a_num=3.6 --if.test=TRUE
```

The error,

```
Error in .add_argument(name, ..., type = type, default = default, required = required,  :
  add_argument | argument (a_name) is required.
Calls: main -> PARSEME -> <Anonymous> -> .add_argument
Execution halted
```

A The source code

A.1 The command-line executable *R* script

Xmisc-ArgumentParser.R

```
#!/usr/bin/env Rscript

## ============================================================================
## This is an executable R script to illustrate `ArgumentParser'
## in CRAN/R package `Xmisc'.
##
##
##
## (c) Xiaobei Zhao
##
## Mon Aug 11 15:28:28 EDT 2014 -0400 (Week 32)
##
##
## Reference:
## CRAN/R package `Xmisc'
## http://CRAN.R-project.org/package=Xmisc
##
## Get help:
## Rscript Xmisc-argumentparser.R -h
## Runme:
## Rscript Xmisc-argumentparser.R --a_name=${USER} --a_int=2 --a_num=3.6 --if.test=TRUE
##
## ============================================================================

require(methods)
require(Xmisc)



PARSEME <- function(){
  parser <- ArgumentParser$new()

  parser$add_usage('Xmisc-argumentparser.R [options]')
  parser$add_description(
    'An executable R script parsing arguments from Unix-like command line.')

  parser$add_argument(
    '--h',type='logical',
    action='store_true',
    help='Print the help page'
  )

  parser$add_argument(
    '--help',type='logical',
    action='store_true',
    help='Print the help page'
  )

  parser$add_argument(
    '--a_name',type='character',
```

```
    required=TRUE,
    help='A a_name.'
  )

  parser$add_argument(
    '--a_int',type='integer',
    default=1,
    help='A integer.'
  )

  parser$add_argument(
    '--a_num',type='numeric',
    default=1,
    help='A number.'
  )

  parser$add_argument(
    '--if.test',type='logical',
    default=FALSE,
    help='Whether it is a test?!'
  )

  return(parser)
}


hello <- function(){
  message('Hello, ',a_name,'!')
  message('The integer is ',a_int,'.')
  message('The number is ',a_num,'.')
  message('')
  flag <- if (if.test) "" else "not "
  message('(This is ',flag,'a test).')
  message('')
}

info <- function(){
  message('class(a_name): ',class(a_name))
  message('class(a_int): ',class(a_int))
  message('class(a_num): ',class(a_num))
  message('class(if.test): ',class(if.test))
  message('')
}


main <- function(){
  parser <- PARSEME()
  parser$helpme()

  hello()
  info()

  invisible()
```

```
}


main()
```

References

[1] Xiaobei Zhao. *Xmisc: Xiaobei's miscellaneous classes and functions*, 2014. R package version 0.2.1.