Package 'xplain'

July 30, 2020

Type Package

Title Providing Interactive Interpretations and Explanations of Statistical Results

Version 0.2.2

Maintainer Joachim Zuckarelli <joachim@zuckarelli.de>

Description Allows to provide live interpretations and explanations of statistical functions in R. These interpretations and explanations are shown when the explained function is called by the user. They can interact with the values of the explained function's actual results to offer relevant, meaningful insights. The 'xplain' interpretations and explanations are based on an easy-to-use XML format that allows to include R code to interact with the returns of the explained function.

Suggests car, methods

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

BugReports https://github.com/jsugarelli/xplain/issues

URL https://github.com/jsugarelli/xplain/,
https://www.zuckarelli.de/xplain/index.html

Repository CRAN

LazyData true

Imports XML, readr, RCurl, httr

RoxygenNote 7.1.1

NeedsCompilation no

Author Joachim Zuckarelli [aut, cre] (<https://orcid.org/0000-0002-9280-3016>)

Date/Publication 2020-07-30 11:10:02 UTC

R topics documented:

`xplain-package` *xplain Package*

Description

This package allows to provide live interpretations and explanations of statistical functions in R. These interpretations and explanations are shown when the explained function is called by the user. They can interact with the values of the explained function's actual results to offer relevant, meaningful insights. The xplain interpretations and explanations are based on an easy-to-use XML format that allows to include R code to interact with the returns of the explained function.

Ressources on the internet

Web tutorial on how to work with xplain: `https://www.zuckarelli.de/xplain/index.html`
xplain cheat sheet: `https://www.zuckarelli.de/xplain/xplain_cheatsheet.pdf`
xplain on GitHub: `https://www.github.com/jsugarelli/xplain`

Comments and suggestions

Your comments and suggestions are highly appreciated.

Author(s)

Author and maintainer: Joachim Zuckarelli, <joachim@zuckarelli.de>.

See Also

`xplain`, `xplain.overview`, `xplain.getcall`

xplain *Showing interpretation information for the results of a function*

Description

Interprets/explains the results of a function call. Main function of the **xplain** package.

Usage

```
xplain(call, xml="", lang = "", level = -1, filename="", sep="\n", title.char="-",
    before=TRUE, addfun="", addfun.args="", addfun.title="")
```

Arguments

call	Function call (as string) to be explained/interpreted.
xml	Path to the xplain XML file containing the interpretation/explanation information (optional). Can be either a local path or an URL. See below for more details on how xplain() tries to find xplain XML files if no file is specified with xml.
lang	ISO country code of the language of the interpretations/explanations that shall be shown (optional). If none is specified, xplain() tries to determine the language of the user's current work environment. If that is not possible then English is taken as defualt value (same effect as lang="EN").
level	Integer number indicating the complexity level of the interpretations/explanations that shall be shown (optional). level is cumulative: All interpretations/explanations with a level number up to the number provided are shown. Default: -1, i.e. all interpretations/explanations are shown.
filename	File to write the xplain() output to (optional). If no file name is provided the xplain() output is shown in the console.
sep	Separator used to separate the outputs from consecutive XML text elements (<text>...</text>) (optional). Default: "\n".
title.char	Character used for underlining titles (optional). Default: "-".
before	Indicates if the results of the call of the explained function shall be shown before the xplain interpretations/explanations, or after (optional). Default: TRUE, i.e. function output is shown before the interpretations/explanations.
addfun	Vector of names of additional functions that shall be called (e.g. summary()), without brackets (optional). It is assumed that these functions take the return object of the explained function as their first argument. Further arguments can be specified with addfun.args. Results of the additional functions are shown right after the output of the explained function.
addfun.args	Vector of arguments (apart from the return object of the explained function) for the additional functions (optional). Example: addfun.args = "trim = 0,na.rm = FALSE". Argument must be of the same length as addfun; addfun.args must be "" (empty string) if the respective additional function does not take any additional arguments.

addfun.title Vector of titles that will be shown as headers to the outputs of the addional functions (optional). Argument must be of the same length as addfun; addfun.args must be "" (empty string) if the respective the output of the additional function shall have no title.

Details

xplain interprets/explains the results of a function call (argument call) by using the information provided in the xplain XML file specified by the xml argument.

1. xplain XML files: xplain XML files follow a simple structure (here an example for the lm() function from the stats package):

<xml>
____<xplain>
_____<package name = "stats">
_____<function name = "lm">
_____<title>...</title>
_____<text>...</text>
_____<result name = "coefficients">
_____<title>...<title>
_____<text>...</text>
_____</result>
_____</function>
_____</package>
____</xplain>
</xml>

<title> elements contain plain text and can be used to structure the output of xplain(). They are underlined by the character given in the title.char argument.

<text> elements contain the actual explanations and interpretations. They may consist of plain text as well as R code. R code must be enclosed by special opening and closing tags like this: !%% Here comes the R code %%!. The placeholder @ can be used to access the explained function's return object, e.g. !%% summary(@) %%!.

With a <result> block you can interpret specific elements of the explained function's return object. The element of the current <result> block can be accessed with the ## placeholder from within an R code section delimited by !%% and %%!. Example: <text> The mean is: !%% mean(##) %%!</text>.

If you use certain R code expressions multiple times and want to save typing effort and reduce error-proness, you can work with a <define> block. Like a <text> block, <define> can encompass both, plain text and R code, e.g.
<define name="my.summary">The summary is: !%% summary(@) %%! </define>. After having defined an expression this way, you can call it from within an R code section by using its name and placing it between the special placeholder tags !** and **!, like <text>Let us have a lookt at the summary: !** my.summary **! </text>.

Sometimes you will want to apply a <text> block not only to one element of the explained function's return object. Consider, for example, the case in which the return object contains a vector and

you want to run through each element of that vector. In this case, you can use a <text> element with the foreach attribute, e.g. <text foreach="rows">. The attribute's value defines how xplain iterates over the object. Possible values are "rows", "columns", "rows,columns", "columns,rows" and "items" for list items. Within R code included in your <text> block you can then refer to the index of the current object with the $ placeholder, e.g. !%% The current element is: @$coefficients[$,1] %%!. If two different indices are in play (e.g. when foreach="rows,columns") then you can work with two index placeholders $, e.g. coefficients[$,$].
Because xplain() needs to know which object to iterate over, you can use <text> with the foreach attribute only form within a <result> block.

xplain() can access XML files both locally and from the internet. xplain XML fiels are not case-sensitive.

2. XML Attributes: <package>, <function>, <result> and <define> blocks always need a name attribute.

<title> and <text> blocks can have lang and level attributes, for the language and the complexity level of the explanations, respectively. lang is an ISO country code and level an integer number (for details, see the explanantion of the corresponding arguments of xplain() above). The values of these two attributes are inherited from higher-level XML elements, e.g. from <package> or <function>. Attributes defined at lower levels (e.g. in an individual <text> element) overrule these inherited attributes.

3. Search paths from xplain XML files: If no path is provided with the xml argument of xplain() or the provided file does not exist then xplain() searches for a suitable XML file in various locations:

1. in the path of the package containing the function from which xplain() was called for a file of the name "package_of_the_calling_function.xml";
2. in the same path for a file with the name "package_of_the_explained_function.xml" (the function given in the call argument);
3. in the path of the package containing the explained function for a file with the name "package_of_the_explained_function.xml";
4. in the current working directory for a file with the name "package_of_the_explained_function.xml"; and
5. in the current working directory for a file with the name "explained_function.xml".

4. More information on xplain XML files: For more details on the structure of xplain XML files, please consult the web tutorial on http://www.zuckarelli.de/xplain/index.html.

Value

xplain() returns the return value of the explained function call (argument call) as if the function were called without xplain(). The interpretation/explanation information is either shown on the screen or written to a file (depending on the filename argument).

More material on the internet

Web tutorial on how to work with xplain: http://www.zuckarelli.de/xplain/index.html
xplain cheat sheet: https://www.zuckarelli.de/xplain/xplain_cheatsheet.pdf

Author(s)

Joachim Zuckarelli, <joachim@zuckarelli.de>

See Also

xplain-package, xplain.overview, xplain.getcall

Examples

```
library(car)

xml.path <- system.file("", "example_lm.xml", package = "xplain")

xplain(call="lm(education ~ young + income + urban, data=Anscombe)",
  xml=xml.path)
```

---

xplain.getcall *Generating function calls for xplain wrapper functions*

Description

This function returns a string containing the call to the explained function to be used in an xplain wrapper function.

Usage

```
xplain.getcall(fun)
```

Arguments

fun	Name of the explained function as string

Details

xplain.getcall() can be called from an xplain wrapper function. It returns a string containing the call to the explained function with all arguments provided to the wrapper function. With xplain.getcall() it is very easy to write xplain wrapper functions.

Wrapper functions are an elegant way to provide access to xplain interpretation/explanation information. The wrapper function takes the sames arguments as the explained function and then calls xplain(). xplain.getcall() identifies the arguments provided to the wrapper function and returns the complete call of the explained function as a string which can then be handed over to xplain() as argument call.

The Examples section illustrates how to build a wrapper function using xplain.getcall().

Value

The full call of the explained function as a string, including the arguments provided by the user to the wrapper function.

More material on the internet

Web tutorial on how to work with xplain: https://www.zuckarelli.de/xplain/index.html
xplain cheat sheet: https://www.zuckarelli.de/xplain/xplain_cheatsheet.pdf

Author(s)

Joachim Zuckarelli, <joachim@zuckarelli.de>

See Also

xplain-package, xplain, xplain.overview

Examples

```
library(car)

xml.path <- system.file("", "example_lm.xml", package = "xplain")

# Example of a wrapper function for lm()
lm.xplain <- function(formula, data, subset, weights, na.action,
                      method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
                      singular.ok = TRUE, contrasts = NULL, offset, ...) {
  call<-xplain.getcall("lm")
  xplain(call, xml=xml.path)
}

lm.xplain(education ~ young + income + urban, data=Anscombe)
```

---

xplain.overview *Summarizing the content of xplain XML files*

Description

xplain.overview summarizes the content of an xplain XML file.

Usage

```
xplain.overview(xml, show.text=FALSE, preserve.seq=FALSE)
```

Arguments

`xml`	Path to the xplain XML file. Can be either a local path or an URL.
`show.text`	Indicates if the full interpretation/explanation texts shall be included in the summary (optional). Default: `FALSE`.
`preserve.seq`	Indicates if the overview results for the interpretation/explanation texts shall be shown in the same sequence as they appear in the XML file (optional). If `FALSE`, the results are sorted, e.g. by package, function, language and complexity level. Default: `FALSE`.

Value

A data frame summarizing the XML file. Each row corresponds to a `<title>` or `<text>` element in the xplain XML file.
The column structure is as follows:

- `Package`: The package to which the explained function belongs.
- `Function`: The explained function.
- `Type`: Indicates whether the element is a `<title>` or a `<text>` element.
- `Language`: Language of the element (also considering inheritance from higher-level XML elements).
- `Level`: Complexity level of the element (also considering inheritance from higher-level XML elements).
- `Result object`: Element of the explained function's return object to which the `<title>` or `<text>` element relates (if any).
- `Iteration`: Type of iteration (if any). Value of the `foreach` attribute of the `<text>` element.
- `Has R code`: Indicates if the `<text>` element includes R code.
- `Uses return obj.`: Indicates if the `<text>` element refers to the explained function's return object.
- `Text`: The text of the respective `<text>` or `<title>` element (including R code). This column is only included if `show.text=TRUE`.

To learn more about the structure of xplain XML files, go to the `xplain` help page or consult the web tutorial.

More material on the internet

Web tutorial on how to work with xplain: `https://www.zuckarelli.de/xplain/index.html`
xplain cheat sheet: `https://www.zuckarelli.de/xplain/xplain_cheatsheet.pdf`

Author(s)

Joachim Zuckarelli, `<joachim@zuckarelli.de>`

See Also

`xplain-package`, `xplain`, `xplain.getcall`

Examples

```
xml.path <- system.file("", "example_lm.xml", package = "xplain")
xplain.overview(xml.path)
```

Index