xplain Cheat Sheet

Important Links

xplain package on CRAN	https://cran.r-project.org/web/packages/xplain/index.html
xplain web tutorial	http://www.zuckarelli.de/xplain/index.html
xplain cheat sheet	http://www.zuckarelli.de/xplain/xplain_cheatsheet.pdf
xplain on GitHub	https://www.github.com/jsugarelli/xplain

Purpose & Application

- xplain allows to **write interpretation/explanation texts** for statistical functions in the form of XML files.
- The user of the functions can read these explanations **while working on his/her specific problems**.
- xplain explanations **can react to the user's results** and provide meaningful insights related to the user's problem.
- For this, the xplain **XML files can contain R code** and can **work with the return object** of the user's function call.

```
> xplain("lm(education ~ young + income + urban)")
> Your R^2 is 0.11 which is quite low. There is a serious
risk your model is misspecified. You should reconsider the
selection of variables included in your model.
```

xplain XML files

<xplain>

1 Any valid xplain XML must be enclosed in an `<xplain>` block. Multiple `<xplain>` blocks per XML file are possible.

<package>

2 A `<package>` block combines all functions from the same package.

<function>

3 Within a `<function>` block, explanations/interpretations for the function as such or for specific elements of the return object can be provided.

<result>

4 Packages explanations/ interpretations related to one element of the function's return object.

```
<xml>
 1 <xplain>
     2 <package name = "stats">
         3 <function name = "lm">
             4 <title>This is about lm</title>
             5 <text>…</text>
             6 <result name = "coefficients">
                 4 <title>…<title>
                 5 <text>…</text>
               </result>
           </function>
       </package>
   </xplain>
</xml>
```

Not case-sensitive

<title>

5 Structures explanations with headers.

<text>

6 The actual explanations/interpretations. Can include R code with references to the function's return object.

Including R code

R code can be easily integrated into `<text></text>` elements:

`<text>` `!%%` `R code` `%%!` `</text>`

R code delimiter tags

Access the explained function's (`<function name="…">`) return object:

- Access the full return object with `@`. Example: `summary(@)`.
- Access the current `<result name="…">` item of the return object with `##`. Example: `mean(##)`.

Using placeholders

`<define name=` `"placeholder"` `> !%%` `R code` `%%!` `</define>`

`</text>` `Text…` `!**` `"placeholder"` `**!` `Text…` `</text>`

Placeholder name delimiter tags

Example:

```
<define name="s">!%% summary(@) %%!</define>
<text>And here is the summary !**s**! for your model</text>
```

Iterating through (items of) the return object

- To apply a `<text>` element to a whole matrix, data frame, vector or list, use the **`foreach`** attribute.
- Value of `foreach` defines what is iterated over and (for 2D structures) in which sequence; `items` is for lists.
- `$` is a placeholder for the index of the current element.
- **Example** (shows all 1st column elements of the coefficient matrix):

```
<text foreach="rows">!%% @$coefficients[$,1] %%!</text>
```

`foreach =`
`"rows"`
`"columns"`
`"rows, columns"`
`"columns, rows"`
`"items"`

Main attributes: Overview

name	Name of the element (package, function, result).
lang	Language (ISO code) of the explanation (e.g. `"EN"`).
level	Complexity level; integer number; cumulative, i.e. `level=1` explanations will also be presented when `level=2` or `level=3` are called.

Attributes: Inheritance and necessity

- Elements **inherit attributes from higher-level** elements; e.g., if only one language, definition on `<xplain>` level suffices. Lower-level attributes overrule higher-level.
- **`name`** attribute required for `<package>`, `<function>` and `<result>` elements.
- All levels shown, if no **`level`** is given to `xplain()`.

Calling xplain()

1 Direct call of `xplain()`

call	Call of the explained function as string
xml	Path of the XML file providing the explanations
lang	Language of the explanations to be shown (default means English)
level	Complexity level of the explanations (cumulative! Default means "all")

2 Wrapper function with `xplain.getcall()`

Example: 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="http://www.zuckarelli.de/example_lm.xml")
}
```

Joachim Zuckarelli 2018 jsugarelli joachim@zuckarelli.de