

# Help Dialog

## Functions for displaying help in Xlisp-Stat

Luca Scrucca

Department of Statistics, Università degli Studi di Perugia, Italy

luca@stat.unipg.it

March 23, 2000

### 1 Introduction

This document describes two functions that allow to display the documentation associated with functions, variables, and prototypes defined in the *Xlisp-Stat* environment.

The standard distribution of *Xlisp-Stat* provides some basic functions:

- `help` provides a brief description of a function;
- `help*` gives help information about all functions with names containing its argument;
- `apropos` returns a listing of the symbols that contain its argument.

The file `helpdlg.lsp` contains two functions designed to provide an easier access to the documentation available. The next two sections give a brief explanation on how to use them. The code is available on the Web at the address <http://www.stat.umn.edu/arc/addons> or at <http://www.stat.unipg.it/~luca>. If you are using *Arc*, you may put this file in your `Extras` directory so it is automatically loaded at the startup. If you are using *Xlisp-Stat* you may want to add the line `(load "helpdlg")` to the file `startinit.lsp`.

### 2 `help*`

The first function is called `Help*` and it is a replacement for the standard `Help*` function. Instead of printing on the listener all the help available, the new function creates an help dialog that allows the user to choose which help to display.

When you call the function `(help* 'SYMBOL)` a dialog with a scroll bar is presented. This dialog contains those functions, variables, and prototypes, whose names have `SYMBOL` as substring. Then, you may select an item from the list and pushing the **Print** button the corresponding documentation is printed on the listener.

On the help dialog you may also double-click on the item, but in this case the documentation is shown in a text window. Any further help requested by double-clicking on an item will be shown on the same window. This option requires the file `displayw.lsp`, which can be downloaded from the same web site you obtained this document.

If the function is called without argument, it shows the **About XLISP-STAT** dialog.

For example, suppose that we want to see all the available documentation regarding any functions whose names include the string “array”. In the listener window, type:

```
> (help* 'array)
```

to obtain a dialog like that on Figure 1. From this dialog we can search through all the functions that have “array” as substring, and select the one we want to see the help. For example, we may want to see the help for the function `make-array`, then pushing the **Print** button (or double-clicking on the item) we get:

```
-----  
MAKE-ARRAY [function-doc]  
Args: (dimensions  
       &key initial-element (initial-contents nil)  
       (displaced-to nil))  
Creates an array of the specified DIMENSIONS. The default for INITIAL-  
ELEMENT depends on ELEMENT-TYPE.
```

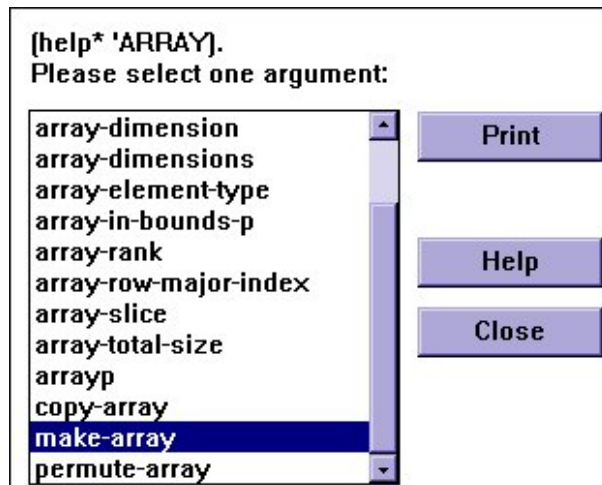


Figure 1: Help dialog for help\* on array

The list of items in the dialog contains not only the functions and variables that have a documentation, but also the prototypes or objects that contain the argument as substring. For example:

```
> (help* 'regr)
```

returns the dialogs in Figure 2. The items in the list are in this case two functions, and one prototype, namely `regression-model-PROTO`. If we select the last item and we click on the **Print** button, the corresponding documentation, if any, is printed as before. But, in the case of a prototype or object, double-clicking on the item will produce a different result, that is the function `display-help*` is called with the item as argument. This function is described in the next section.

Any dialog has also an **Help** button, which opens a window with an brief description on how to use help\*, and a **Close** button, which closes the help dialog.

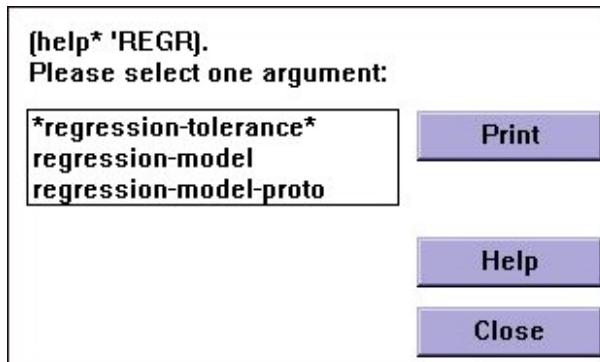


Figure 2: Help dialog for help\* on regr

### 3 display-help\*

The function `(display-help* 'SYMBOL)` prints the documentation associated with the methods own by the prototype called `SYMBOL`. A dialog with scrolling bar is presented, and it is possible to display the documentation for the selected items by pushing the **Print** button (or double-clicking on a item) as for the `help*` function described in the previous section.

For example, typing

```
> (display-help* 'dialog-proto)
```

will show a dialog with the methods own the `dialog-proto`, and a second list with the parents of the prototype.

Suppose that from the dialog in figure 2 we double-click on the item `regression-model-proto`. The dialog on figure 3 is displayed. In the left part of this dialog there is a list with all the methods own by the `regression-model-proto`, whereas on the right part there is a list with the parents of this prototype. Double-clicking or pushing the **Print** button on one of the items in the list of methods will display the corresponding documentation, whereas double-clicking on one of the items on the second list will call again the function `display-help*` with the selected prototype as argument. As for the `help*` function, double-clicking will display the help on a text window if the file `displayw.lsp` have been loaded.

For example, if we want an help on the method about the residual sum of squares, we can select the relative item and pushing the **Print** button we get:

```
-----
REGRESSION-MODEL-PROTO: RESIDUAL-SUM-OF-SQUARES
Message args: ()
Returns the residual sum of squares for the model.
```

#### References

- Cook R.D. and Weisberg S. (1999), *Applied Regression Including Computing and Graphics*. New York: Wiley.
- Tierney L. (1990), *Lisp-Stat: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*. New York: Wiley.

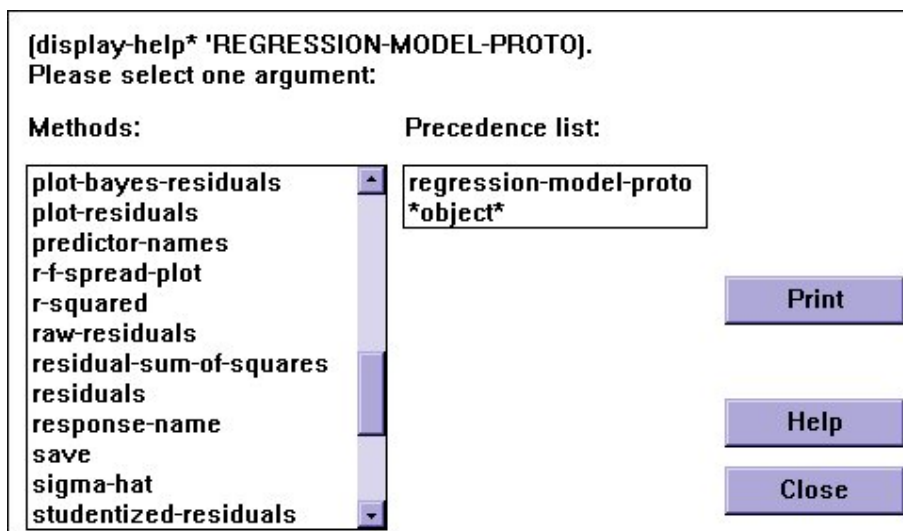


Figure 3: Help dialog for display-help\*