

# Package ‘IGoRRR’

July 21, 2025

**Type** Package

**Title** A Shiny Interface for Simple Data Management

**Version** 0.3.4

**Author** Jean-Luc Lipatz [aut, cre], Bernard Gestin [ctb]

**Maintainer** Jean-Luc Lipatz <jllipatz@protonmail.com>

**Description** Launches a shiny application generating code to  
view tables in several ways,  
import/export tables,  
modify tables,  
make some basic graphics.

'IGoR' is a graphic user interface designed to help beginners  
using simple functions around table management and exploration.

Inspired by 'Rcmdr', 'IGoR' is a code generator that, with simple inputs under a Shiny application,  
provides R code mainly built around the 'tidyverse' or some packages in the direct line of the Mo-  
saic project:

the 'rio' and 'ggformula' packages.

The generated code doesn't depend on IGoR and can be manually modi-  
fied by the user or copied elsewhere.

**License** GPL (>= 2)

**Encoding** UTF-8

**ByteCompile** yes

**Depends** R (>= 4.0.0)

**Imports** shiny, shinydashboard, shinyWidgets, shinyFiles, htmltools,  
rhandsontable, sortable, purrr, stringr, glue, lubridate, zoo,  
tibble, dplyr (>= 1.0.0), tidyr, fuzzyjoin, rio, jsonlite,  
arrow, fst, feather, haven, readxl, readODS, Hmisc, skimr,  
tables, ggformula, mapsf, clipr, magrittr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-03-10 11:30:02 UTC

Contents

go .....	2
<b>Index</b>	<b>4</b>

---

go	<i>IGoR</i>
----	-------------

---

Description

Launches IGoR.

Usage

```
go(envir=.GlobalEnv, examples=TRUE)
```

Arguments

- |          |   |
|----------|---|
| envir    | An environment where to remember tables that will be built by the user with IGoR. Only tables in this environment will be usable by IGoR. |
| examples | If TRUE, load three small tables usefull for testing IGoR. If same named tables exist they will be replaced.                              |

Details

**IGoRRR** (within the application: IGoR for I Go R), generates R code using a limited set of packages, mostly from the 'tidyverse' or inspired by the 'Mosaic' project. The generated code is systematically displayed and, in most cases, not activated until the user aks for. Statisticals tables and graphics are the exceptions. The code may be copied elsewhere to help building a program that will not need IGoR to run. Copied code give some indication of needed packages when it is possible.

There are two small exceptions to the rule, two trivial shortcuts used to produce a readable code. They are both related to weighted statitics. Their (small) R code is in the file `Utilities.R`.

```
wtd.sum  
wtd.percent
```

Startup

Before running IGoR, the packages used to generate code are requested to be added to loaded ones. These packages will still be available when exiting IGoR, in order to make any copy of the generated code still able to run.

Configuration

Configuration files reside in the `text` subdirectory of the package.

The file `config.json` contains the description of drives used when searching a file or a place to write a file. They should be changed to places that are meaningfull on the target system. Among them only `home` is mandatory but it may reference another place than the working directory.

The file `config.json` contains the structure of menus displayed by IGoR. They can be reordered differently and some items may be removed but there is now provision, today, to add some. The

menus names cannot be changed as they reference function names in package or entries in the file FR.json where the actual titles are extracted.

The file FR.json contains every text that is displayed in the user's language. It contains the title of every widget displayed by Shiny and also every text that is used to help the user in its use of IGoR. Current file is in french language and may be completed by files in some other languages in the future, although the code today doesn't give any choice.

**Value**

None. Returns NULL invisibly.

**Examples**

```
## Not run:
go()          # will retrieve and store tables in .GlobalEnv

## End(Not run)
## Not run:
e <- new.env() # will store tables in a specific environment
go(e)          # make some tables then leave
go(e, FALSE)   # go back to IGoR with previously made tables without refreshing example tables

## End(Not run)
```

# Index

go, [2](#)