

# Package: rvisual (via r-universe)

June 2, 2026

**Title** Visual Data Analysis 'RStudio' 'Addin' with AI Copilot

**Version** 0.1.0

**Description** An 'RStudio' 'addin' providing a visual, point-and-click interface for data manipulation and analysis, designed for users transitioning from 'SPSS' to R. Every visual action generates clean, reproducible R code using 'tidyverse' conventions ('dplyr', 'tidyr'). Includes a context-aware AI copilot supporting multiple 'LLM' providers ('OpenAI', 'Anthropic' Claude, 'Google Gemini') with built-in proxy support and privacy controls.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**Imports** shiny (>= 1.7.0), rstudioapi (>= 0.13), bslib (>= 0.5.0), DT (>= 0.28), dplyr (>= 1.1.0), readxl (>= 1.4.0), jsonlite (>= 1.8.0), httr2 (>= 0.2.3), glue (>= 1.6.0), rlang (>= 1.1.0), DBI (>= 1.1.0), tools, stats, utils

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, haven, readr, writexl, RPostgres, RMySQL, ROracle, RSQLite, odbc

**Config/testthat/edition** 3

**URL** <https://github.com/facszero/rvisual>

**BugReports** <https://github.com/facszero/rvisual/issues>

**Config/pak/sysreqs** cmake make libuv1-dev libssl-dev zlib1g-dev

**Repository** <https://facszero.r-universe.dev>

**Date/Publication** 2026-06-02 18:10:55 UTC

**RemoteUrl** <https://github.com/facszero/rvisual>

**RemoteRef** main

**RemoteSha** 91825cc38756615f00c75d46f5b2fd88f3f00a81

## Contents

ai_send	2
apply_operations	3
classify_visual_type	3
config_load	4
config_path	4
config_save	4
discover_datasets	5
generate_code	5
get_metadata	6
history_log	6
launch_rvisual	7
load_file	7
mod_ai_ui	8
mod_builder_ui	8
mod_code_ui	9
mod_config_ui	9
mod_explorer_ui	10
mod_history_ui_inline	10
op_arrange	11
op_filter	11
op_group_summarise	12
op_join	12
op_mutate	13
op_recode	13
op_rename	13
op_select	14
rstudio_active_script_path	14
rstudio_insert_code	15
rstudio_run_in_console	15
rvisual_server	16
rvisual_ui	16
<b>Index</b>	<b>17</b>

---

 ai\_send

*Capa de Integración con IA*


---

### Description

Usa curl directamente para mayor compatibilidad con proxies corporativos.

### Usage

```
ai_send(cfg, system_prompt, user_prompt)
```

**Arguments**

- cfg            Lista de configuración del proveedor IA: provider, model, api\_key.
- system\_prompt   String con el prompt de sistema (contexto del dataset activo).
- user\_prompt    String con el mensaje del usuario.

---

apply\_operations      *Aplicar una lista de operaciones a un data.frame*

---

**Description**

Aplicar una lista de operaciones a un data.frame

**Usage**

```
apply_operations(df, ops)
```

**Arguments**

- df            Data.frame base
- ops           Lista de operaciones (output de op\_\* functions)

**Value**

Data.frame transformado

---

classify\_visual\_type    *Clasificar tipo visual de una columna (para mostrar al usuario)*

---

**Description**

Clasificar tipo visual de una columna (para mostrar al usuario)

**Usage**

```
classify_visual_type(col)
```

**Arguments**

- col            Vector de una columna del data.frame a clasificar.

---

config_load	<i>Cargar configuración guardada</i>
-------------	--------------------------------------

---

**Description**

Cargar configuración guardada

**Usage**

```
config_load()
```

**Value**

Lista de configuración o lista vacía

---

config_path	<i>Configuración persistente de RVisual</i>
-------------	---

---

**Description**

Guarda/lee configuración en un archivo JSON local usando el directorio de datos del usuario (rappdirs-style).

**Usage**

```
config_path()
```

---

config_save	<i>Guardar configuración</i>
-------------	------------------------------

---

**Description**

Guardar configuración

**Usage**

```
config_save(cfg)
```

**Arguments**

cfg	Lista de configuración
-----	------------------------

---

discover_datasets	<i>Descubrimiento de datasets, metadatos y carga de archivos Listar data.frames disponibles en el entorno global</i>
-------------------	--

---

**Description**

Descubrimiento de datasets, metadatos y carga de archivos Listar data.frames disponibles en el entorno global

**Usage**

```
discover_datasets()
```

**Value**

Lista de listas con name, nrow, ncol por cada dataset

---

generate_code	<i>Generador de Código R</i>
---------------	------------------------------

---

**Description**

Convierte un stack de operaciones (lista de objetos op\_\*) en código R limpio, legible y reproducible usando tidyverse (dplyr/tidyr).

**Usage**

```
generate_code(df_name, ops, assign_to = NULL)
```

**Arguments**

df_name	Nombre del objeto R que contiene el dataset
ops	Lista de operaciones (output de op_* functions)
assign_to	Si no es NULL, el resultado se asigna a este nombre

**Details**

Estrategia de generación:

- Una operación por línea del pipe
- Pipe nativo (|>) con R >= 4.1
- Convenciones tidyverse: snake\_case, nombres explícitos
- Comentarios explicativos cuando la operación es compleja
- Código legible para usuarios en transición desde SPSS Generar código R desde nombre de dataset y stack de operaciones

**Value**

String con código R formateado

---

get_metadata	<i>Extraer metadatos estructurados de un data.frame</i>
--------------	---

---

**Description**

Extraer metadatos estructurados de un data.frame

**Usage**

```
get_metadata(df, name = "dataset")
```

**Arguments**

df	Data.frame
name	Nombre del objeto (para display)

**Value**

Lista con nrow, ncol, columns (data.frame con info de cada columna)

---

history_log	<i>Registrar evento en el historial de sesión</i>
-------------	---

---

**Description**

Registrar evento en el historial de sesión

**Usage**

```
history_log(history, type, detail = list())
```

**Arguments**

history	reactiveVal que contiene la lista de eventos
type	String identificador del tipo de evento
detail	Lista con detalles del evento

---

launch_rvisual	<i>Lanzar RVisual</i>
----------------	-----------------------

---

**Description**

Punto de entrada del addin de RStudio.

**Usage**

```
launch_rvisual(browser = FALSE)
```

**Arguments**

browser	Lógico. Si TRUE abre en el browser del sistema en lugar del panel Viewer de RStudio. Default FALSE.
---------	---

---

load_file	<i>Cargar un archivo de datos según su extensión</i>
-----------	--

---

**Description**

Cargar un archivo de datos según su extensión

**Usage**

```
load_file(path, extension)
```

**Arguments**

path	Ruta al archivo
extension	Extensión sin punto: "csv", "xlsx", "rds", "sav"

**Value**

Data.frame o tibble

---

`mod_ai_ui`*Módulo: Asistente IA Contextual*

---

**Description**

Panel de chat con agente IA multi-proveedor. El agente conoce:

- El nombre y esquema del dataset activo
- Los tipos de columnas
- El stack de operaciones actual
- Opcionalmente, una muestra limitada de datos

**Usage**`mod_ai_ui(id)`**Arguments**

`id` ID del módulo Shiny (string). Usado internamente por `NS()` para crear el namespace del módulo.

**Details**

Modalidad segura: el agente propone código y el usuario decide si ejecutarlo.

---

`mod_builder_ui`*Módulo: Constructor Visual de Operaciones*

---

**Description**

Módulo: Constructor Visual de Operaciones

**Usage**`mod_builder_ui(id)`**Arguments**

`id` ID del módulo Shiny (string). Usado internamente por `NS()` para crear el namespace del módulo.

---

`mod_code_ui`*Módulo: Panel de Código R Generado*

---

**Description**

Muestra en tiempo real el código R producido por el constructor visual. Permite:

- Copiar al portapapeles
- Insertar en el script activo de RStudio
- Ejecutar directamente
- Guardar como archivo .R

**Usage**`mod_code_ui(id)`**Arguments**

<code>id</code>	ID del módulo Shiny (string). Usado internamente por <code>NS()</code> para crear el namespace del módulo.
-----------------	--

---

`mod_config_ui`*Módulo: Configuración*

---

**Description**

Pantalla de configuración del addin. Permite gestionar proveedores IA, API keys, modelos y preferencias.

**Usage**`mod_config_ui(id)`**Arguments**

<code>id</code>	ID del módulo Shiny (string). Usado internamente por <code>NS()</code> para crear el namespace del módulo.
-----------------	--

---

mod\_explorer\_ui      *Módulo: Explorador de datos*

---

### Description

Vista tabular interactiva del dataset activo. Incluye:

- Tabla paginada con DT
- Resumen de columnas con tipos detectados
- Estadísticas rápidas por variable
- Búsqueda de variables

### Usage

```
mod_explorer_ui(id)
```

### Arguments

id	ID del módulo Shiny (string). Usado por NS() para crear el namespace del módulo.
----	--

---

mod\_history\_ui\_inline      *Renderizar historial como UI inline (para modal)*

---

### Description

Renderizar historial como UI inline (para modal)

### Usage

```
mod_history_ui_inline(history_list)
```

### Arguments

history_list	Lista de eventos del historial de sesión, generada por history_log().
--------------	---

---

op_arrange	<i>Crear operación: arrange</i>
------------	---------------------------------

---

**Description**

Crear operación: arrange

**Usage**

```
op_arrange(col, desc = FALSE)
```

**Arguments**

col	Nombre de columna
desc	Lógico: TRUE = descendente

---

op_filter	<i>Crear operación: filter</i>
-----------	--------------------------------

---

**Description**

Crear operación: filter

**Usage**

```
op_filter(col, op, value)
```

**Arguments**

col	Nombre de columna
op	Operador: "=", "!=", ">", "<", ">=", "<="
value	Valor de comparación

---

op\_group\_summarise      *Crear operación: group\_summarise*

---

**Description**

Crear operación: group\_summarise

**Usage**

```
op_group_summarise(group_cols, summary_fns)
```

**Arguments**

group\_cols      Character vector de columnas de agrupación  
summary\_fns      Lista nombrada: nombre\_nueva\_col = list(fn, col\_origen)

---

op\_join                      *Crear operación: join*

---

**Description**

Crear operación: join

**Usage**

```
op_join(right_df_name, by_cols, join_type = "left")
```

**Arguments**

right\_df\_name      Nombre del segundo dataset  
by\_cols              Vector de columnas clave (mismas en ambos)  
join\_type            "left", "right", "inner", "full"

---

op_mutate	<i>Crear operación: mutate (nueva variable)</i>
-----------	---

---

**Description**

Crear operación: mutate (nueva variable)

**Usage**

```
op_mutate(new_col, expression)
```

**Arguments**

new_col	Nombre de la nueva columna
expression	Expresión como string (será validada antes de generar código)

---

---

op_recode	<i>Crear operación: recode</i>
-----------	--------------------------------

---

**Description**

Crear operación: recode

**Usage**

```
op_recode(col, mapping)
```

**Arguments**

col	Columna a recodificar
mapping	Lista nombrada: valor_original = valor_nuevo

---

---

op_rename	<i>Crear operación: rename</i>
-----------	--------------------------------

---

**Description**

Crear operación: rename

**Usage**

```
op_rename(old_name, new_name)
```

**Arguments**

old_name	Nombre actual de la columna
new_name	Nuevo nombre

---

`op_select`*Motor de Operaciones Visuales — Modelo Interno*

---

**Description**

Define la representación estructurada intermedia de las operaciones que el usuario construye visualmente.

**Usage**

```
op_select(cols)
```

**Arguments**

`cols` Character vector de nombres de columnas a seleccionar

**Details**

En lugar de concatenar texto, cada operación es un objeto R con campos tipados que permiten:

- Generar código R limpio y reproducible
- Reconstruir la UI desde el estado
- Soportar historial y deshacer
- Exportar/importar flujos de trabajo

Estructura de una operación (lista R):

**type** Tipo de operación: select, filter, arrange, group\_summarise, mutate, rename, recode, join

**label** Descripción legible para mostrar en UI

**params** Lista nombrada con parámetros específicos del tipo

**created\_at** Timestamp de creación

Crear operación: select

---

`rstudio_active_script_path`*Obtener el path del script activo en RStudio*

---

**Description**

Obtener el path del script activo en RStudio

**Usage**

```
rstudio_active_script_path()
```

---

rstudio\_insert\_code    *RStudio Bridge — integración con el IDE*

---

**Description**

Funciones para interactuar con RStudio desde el addin: insertar código, leer entorno, navegar entre archivos. Insertar código en el script activo de RStudio

**Usage**

```
rstudio_insert_code(code)
```

**Arguments**

code                    String con el código R a insertar

---

rstudio\_run\_in\_console  
*Ejecutar código en la consola de RStudio*

---

**Description**

Ejecutar código en la consola de RStudio

**Usage**

```
rstudio_run_in_console(code)
```

**Arguments**

code                    String con el código R a ejecutar en la consola de RStudio.

---

rvisual_server	<i>Server principal de RVisual</i>
----------------	------------------------------------

---

**Description**

Orquesta todos los módulos Shiny. El estado compartido entre módulos se gestiona a través de reactivos pasados como argumentos — no mediante variables globales.

**Usage**

```
rvisual_server(input, output, session)
```

**Arguments**

```
input, output, session
```

Parámetros estándar de Shiny server

---

rvisual_ui	<i>UI principal de RVisual</i>
------------	--------------------------------

---

**Description**

Define la interfaz Shiny completa del addin. Estructura en 6 paneles navegables via sidebar.

**Usage**

```
rvisual_ui()
```

**Value**

Un objeto shiny.tag con la UI completa

# Index

ai\_send, 2  
apply\_operations, 3  
  
classify\_visual\_type, 3  
config\_load, 4  
config\_path, 4  
config\_save, 4  
  
discover\_datasets, 5  
  
generate\_code, 5  
get\_metadata, 6  
  
history\_log, 6  
  
launch\_rvisual, 7  
load\_file, 7  
  
mod\_ai\_ui, 8  
mod\_builder\_ui, 8  
mod\_code\_ui, 9  
mod\_config\_ui, 9  
mod\_explorer\_ui, 10  
mod\_history\_ui\_inline, 10  
  
op\_arrange, 11  
op\_filter, 11  
op\_group\_summarise, 12  
op\_join, 12  
op\_mutate, 13  
op\_recode, 13  
op\_rename, 13  
op\_select, 14  
  
rstudio\_active\_script\_path, 14  
rstudio\_insert\_code, 15  
rstudio\_run\_in\_console, 15  
rvisual\_server, 16  
rvisual\_ui, 16