

Package ‘ravel’

August 3, 2026

Title AI Copilot for R Analysis Workflows in 'RStudio'

Version 0.1.4

Description An AI copilot for R users in 'RStudio' and Posit workflows with active-editor, workspace, object, console, plot, and git-aware context. Provides statistical helpers for interpreting `lm()` and `glm()` models, stages code and file actions before execution, drafts reproducible 'Quarto' content, and connects to official provider APIs or CLIs for 'OpenAI', 'GitHub Copilot', 'Gemini', and 'Anthropic'.

License MIT + file LICENSE

URL <https://github.com/msaule/ravel>, <https://msaule.github.io/ravel/>

BugReports <https://github.com/msaule/ravel/issues>

Language en-US

Encoding UTF-8

Depends R (>= 4.2.0)

Imports cli, digest, httr2, jsonlite, miniUI, rstudioapi, shiny, stats, tibble, tools, utils

Suggests keyring, knitr, lintr, pkgdown, rcmdcheck, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website pkgdown

Config/Needs/release devtools, rcmdcheck, roxygen2

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Markuss Saule [aut, cre, cph]

Maintainer Markuss Saule <markusstomas@gmail.com>

Repository CRAN

Date/Publication 2026-08-03 02:40:02 UTC

Contents

ravel_apply_action	2
ravel_approve_action	3
ravel_auth_status	4
ravel_chat_addin	4
ravel_chat_turn	5
ravel_collect_context	5
ravel_doctor	6
ravel_draft_quarto_section	7
ravel_get_setting	7
ravel_interpret_model	8
ravel_launch_login	8
ravel_list_project_files	9
ravel_list_providers	9
ravel_login	10
ravel_logout	10
ravel_mcp_tool	11
ravel_mcp_tools	11
ravel_new_action	12
ravel_open_provider_page	12
ravel_preview_code	13
ravel_provider_capabilities	14
ravel_read_history	14
ravel_reject_action	15
ravel_run_code	15
ravel_settings_addin	16
ravel_setup_addin	16
ravel_set_api_key	17
ravel_set_bearer_token	17
ravel_set_setting	18
ravel_stage_file_write	18
ravel_suggest_diagnostics	19
ravel_summarize_model	19
ravel_summarize_object	20
ravel_verify_provider	20

Index	22
--------------	-----------

ravel_apply_action	<i>Execute a staged action</i>
--------------------	--------------------------------

Description

Execute a staged action

Usage

```
ravel_apply_action(
  action,
  approve = FALSE,
  envir = NULL,
  allow_outside_project = FALSE
)
```

Arguments

action	A ravel_action.
approve	Whether to override and treat the action as approved.
envir	Evaluation environment for code execution. When NULL, Ravel uses a dedicated session-scoped environment that inherits from <code>globalenv()</code> for reads without writing into <code>.GlobalEnv</code> by default.
allow_outside_project	Whether approved file actions may write outside the detected project root. The default is FALSE.

Value

A structured result list.

ravel_approve_action	<i>Approve a staged action</i>
----------------------	--------------------------------

Description

Approve a staged action

Usage

```
ravel_approve_action(action)
```

Arguments

action	A ravel_action.
--------	-----------------

Value

The updated action.

ravel_auth_status	<i>Report provider auth status</i>
-------------------	------------------------------------

Description

Report provider auth status

Usage

```
ravel_auth_status(provider = c("openai", "copilot", "gemini", "anthropic"))
```

Arguments

provider	Provider name.
----------	----------------

Value

A named list describing auth configuration.

ravel_chat_addin	<i>Launch the Ravel chat addin</i>
------------------	------------------------------------

Description

Launch the Ravel chat addin

Usage

```
ravel_chat_addin()
```

Value

Invisibly launches a Shiny gadget.

ravel_chat_turn	<i>Run one chat turn through a provider</i>
-----------------	---

Description

Run one chat turn through a provider

Usage

```
ravel_chat_turn(  
  prompt,  
  provider = NULL,  
  model = NULL,  
  context = NULL,  
  history = NULL  
)
```

Arguments

prompt	User prompt.
provider	Provider name.
model	Optional model override.
context	Optional precomputed context.
history	Optional existing chat history.

Value

A list with message, actions, and raw.

ravel_collect_context	<i>Collect context for a Ravel chat turn</i>
-----------------------	--

Description

Collect context for a Ravel chat turn

Usage

```
ravel_collect_context(  
  include_selection = TRUE,  
  include_file = TRUE,  
  include_objects = TRUE,  
  include_console = TRUE,  
  include_plot = TRUE,
```

```

    include_session = TRUE,
    include_project = TRUE,
    include_git = TRUE,
    include_activity = TRUE,
    envir = NULL,
    max_objects = 10L
  )

```

Arguments

include_selection	Include the active selection.
include_file	Include the active file contents.
include_objects	Include summaries of loaded objects.
include_console	Include recent Ravel-managed console output.
include_plot	Include current plot metadata when available.
include_session	Include session information.
include_project	Include project root and file listing.
include_git	Include git status and diff summaries when available.
include_activity	Include recent Ravel action state.
envir	Environment used for object summaries. When NULL, Ravel reads from the current global workspace without modifying it.
max_objects	Maximum number of objects to summarize.

Value

A named list.

ravel_doctor	<i>Inspect local Ravel readiness</i>
--------------	--------------------------------------

Description

Inspect local Ravel readiness

Usage

```
ravel_doctor()
```

Value

A tibble of system and provider checks with suggested fixes.

`ravel_draft_quarto_section`*Draft a Quarto section from analysis context*

Description

Draft a Quarto section from analysis context

Usage

```
ravel_draft_quarto_section(  
  section = c("results", "methods", "diagnostics"),  
  model = NULL,  
  context = NULL,  
  include_chunk = TRUE  
)
```

Arguments

<code>section</code>	Section type.
<code>model</code>	Optional fitted model object.
<code>context</code>	Optional collected context.
<code>include_chunk</code>	Whether to include a placeholder code chunk.

Value

A character string containing Quarto markdown.

`ravel_get_setting` *Get a Ravel setting*

Description

Get a Ravel setting

Usage

```
ravel_get_setting(key, default = NULL)
```

Arguments

<code>key</code>	Setting name.
<code>default</code>	Default value when the setting is missing.

Value

The setting value.

ravel_interpret_model *Interpret a model in plain English*

Description

Interpret a model in plain English

Usage

```
ravel_interpret_model(model)
```

Arguments

model A fitted model object.

Value

A character string.

ravel_launch_login *Launch an official provider login flow*

Description

Launch an official provider login flow

Usage

```
ravel_launch_login(  
  provider = c("openai", "copilot", "gemini", "anthropic"),  
  mode = NULL  
)
```

Arguments

provider Provider name.
mode Optional auth mode override.

Value

A login plan list, invisibly.

ravel_list_project_files

List project files for context gathering

Description

List project files for context gathering

Usage

```
ravel_list_project_files(root = NULL, limit = 200L)
```

Arguments

root	Optional project root.
limit	Maximum number of file paths to return.

Value

A character vector of relative paths.

ravel_list_providers *List configured providers*

Description

List configured providers

Usage

```
ravel_list_providers()
```

Value

A tibble describing the available provider adapters.

ravel_login	<i>Start a provider login flow</i>
-------------	------------------------------------

Description

Start a provider login flow

Usage

```
ravel_login(  
  provider = c("openai", "copilot", "gemini", "anthropic"),  
  mode = NULL  
)
```

Arguments

- provider Provider name.
- mode Optional auth mode override.

Value

A list describing the supported login action.

ravel_logout	<i>Logout a provider from Ravel-managed credentials</i>
--------------	---

Description

Logout a provider from Ravel-managed credentials

Usage

```
ravel_logout(provider = c("openai", "copilot", "gemini", "anthropic"))
```

Arguments

- provider Provider name.

Value

Invisibly TRUE.

ravel_mcp_tool	<i>Declare a remote MCP tool for OpenAI Responses API calls</i>
----------------	---

Description

Declare a remote MCP tool for OpenAI Responses API calls

Usage

```
ravel_mcp_tool(
  server_label,
  server_url,
  allowed_tools = NULL,
  require_approval = c("always", "never"),
  server_description = NULL
)
```

Arguments

server_label	Short label for the MCP server.
server_url	HTTPS URL for the remote MCP server.
allowed_tools	Optional character vector limiting exposed MCP tools.
require_approval	Approval policy sent to providers that support it. Ravel defaults to "always" so every remote call is shown for review before data is shared with the MCP server. "never" is an explicit opt-out intended only for a server and data flow the user has already reviewed.
server_description	Optional human-readable description shown to the model when it selects tools from this server.

Value

A provider-ready MCP tool definition.

ravel_mcp_tools	<i>Normalize MCP tool definitions for provider requests</i>
-----------------	---

Description

Normalize MCP tool definitions for provider requests

Usage

```
ravel_mcp_tools(tools = list())
```

Arguments

tools A list of MCP tool definitions created by `ravel_mcp_tool()`.

Value

A list suitable for provider payloads.

ravel_new_action	<i>Create a new staged Ravel action</i>
------------------	---

Description

Create a new staged Ravel action

Usage

```
ravel_new_action(type, label, payload, provider = NULL, model = NULL)
```

Arguments

type	Action type.
label	Human-readable label.
payload	Action payload.
provider	Optional provider name.
model	Optional model name.

Value

A `ravel_action` object.

ravel_open_provider_page	<i>Open an official provider documentation or key-management page</i>
--------------------------	---

Description

Open an official provider documentation or key-management page

Usage

```
ravel_open_provider_page(  
  provider = c("openai", "copilot", "gemini", "anthropic"),  
  page = c("docs", "api_keys")  
)
```

Arguments

provider	Provider name.
page	Which page to open.

Value

The opened URL, invisibly.

ravel_preview_code	<i>Preview generated code as a staged action</i>
--------------------	--

Description

Preview generated code as a staged action

Usage

```
ravel_preview_code(
  code,
  label = "Run generated R code",
  provider = NULL,
  model = NULL
)
```

Arguments

code	R code text.
label	Label for the staged action.
provider	Optional provider name.
model	Optional model name.

Value

A ravel_action object.

ravel_provider_capabilities
Report provider capabilities

Description

Report provider capabilities

Usage

```
ravel_provider_capabilities(  
  provider = c("openai", "copilot", "gemini", "anthropic")  
)
```

Arguments

provider Provider name.

Value

A named list.

ravel_read_history *Read recent Ravel history entries*

Description

Read recent Ravel history entries

Usage

```
ravel_read_history(limit = 100L)
```

Arguments

limit Maximum number of history entries to return.

Details

History stays in session memory by default. To mirror it to disk explicitly, configure `options(ravel.user_dirs = list(data = "<path>"))`.

Value

A tibble.

ravel_reject_action	<i>Reject a staged action</i>
---------------------	-------------------------------

Description

Reject a staged action

Usage

```
ravel_reject_action(action)
```

Arguments

action	A <code>ravel_action</code> .
--------	-------------------------------

Value

The updated action.

ravel_run_code	<i>Run R code with explicit approval semantics</i>
----------------	--

Description

Run R code with explicit approval semantics

Usage

```
ravel_run_code(
  action,
  approve = FALSE,
  envir = NULL,
  allow_outside_project = FALSE
)
```

Arguments

action	Either a <code>ravel_action</code> or a character string of R code.
approve	Whether to approve and run immediately.
envir	Evaluation environment. When <code>NULL</code> , code runs in Ravel's dedicated session-scoped execution environment instead of <code>.GlobalEnv</code> .
allow_outside_project	Whether approved file actions may write outside the detected project root. Ignored for character R code.

Value

A structured result list.

ravel_settings_addin	<i>Launch the Ravel settings addin</i>
----------------------	--

Description

Launch the Ravel settings addin

Usage

```
ravel_settings_addin()
```

Value

Invisibly launches a Shiny gadget.

ravel_setup_addin	<i>Launch the Ravel setup assistant</i>
-------------------	---

Description

Launch the Ravel setup assistant

Usage

```
ravel_setup_addin()
```

Value

Invisibly launches a Shiny gadget.

ravel_set_api_key	<i>Store an API key for a provider</i>
-------------------	--

Description

Store an API key for a provider

Usage

```
ravel_set_api_key(  
  provider = c("openai", "gemini", "anthropic"),  
  key,  
  persist = TRUE  
)
```

Arguments

provider	Provider name.
key	API key value.
persist	Whether to try to persist the secret using keyring.

Value

The stored key, invisibly.

ravel_set_bearer_token	<i>Store an externally managed OAuth access token for a provider</i>
------------------------	--

Description

Store an externally managed OAuth access token for a provider

Usage

```
ravel_set_bearer_token(provider = c("gemini"), token, persist = TRUE)
```

Arguments

provider	Provider name.
token	OAuth access token value. Ravel does not initiate or refresh OAuth tokens; use an official Google flow or Application Default Credentials tooling to obtain and refresh the token.
persist	Whether to try to persist the token using keyring.

Value

The stored token, invisibly.

ravel_set_setting	<i>Set a Ravel setting</i>
-------------------	----------------------------

Description

Set a Ravel setting

Usage

```
ravel_set_setting(key, value)
```

Arguments

key	Setting name.
value	Setting value.

Details

Settings are stored in session memory by default. To mirror non-sensitive settings to disk explicitly, configure `options(ravel.user_dirs = list( config = "<path>" ))` before calling this function.

Value

The written settings list, invisibly.

ravel_stage_file_write	<i>Stage a file write action</i>
------------------------	----------------------------------

Description

Stage a file write action

Usage

```
ravel_stage_file_write(  
  path,  
  text,  
  append = FALSE,  
  provider = NULL,  
  model = NULL  
)
```

Arguments

path	Target file path.
text	Text to write.
append	Whether to append instead of overwrite.
provider	Optional provider name.
model	Optional model name.

Value

A ravel_action object.

ravel_suggest_diagnostics
Suggest diagnostics for a model

Description

Suggest diagnostics for a model

Usage

```
ravel_suggest_diagnostics(model)
```

Arguments

model	A fitted model object.
-------	------------------------

Value

A character vector of suggested checks.

ravel_summarize_model *Summarize a model object*

Description

Summarize a model object

Usage

```
ravel_summarize_model(model, name = NULL)
```

Arguments

model	A fitted model object.
name	Optional object name.

Value

A named list with model metadata and coefficient summaries.

```
ravel_summarize_object
```

Summarize an R object for provider context

Description

Summarize an R object for provider context

Usage

```
ravel_summarize_object(x, name = NULL)
```

Arguments

x	An R object.
name	Optional object name.

Value

A named list.

```
ravel_verify_provider
```

Verify a provider with a tiny live prompt

Description

Verify a provider with a tiny live prompt

Usage

```
ravel_verify_provider(
  provider = c("openai", "copilot", "gemini", "anthropic"),
  model = NULL
)
```

Arguments

<code>provider</code>	Provider name.
<code>model</code>	Optional model override.

Value

A named list with `ok`, `content`, `provider`, and `model`.

Index

ravel_apply_action, [2](#)
ravel_approve_action, [3](#)
ravel_auth_status, [4](#)
ravel_chat_addin, [4](#)
ravel_chat_turn, [5](#)
ravel_collect_context, [5](#)
ravel_doctor, [6](#)
ravel_draft_quarto_section, [7](#)
ravel_get_setting, [7](#)
ravel_interpret_model, [8](#)
ravel_launch_login, [8](#)
ravel_list_project_files, [9](#)
ravel_list_providers, [9](#)
ravel_login, [10](#)
ravel_logout, [10](#)
ravel_mcp_tool, [11](#)
ravel_mcp_tools, [11](#)
ravel_new_action, [12](#)
ravel_open_provider_page, [12](#)
ravel_preview_code, [13](#)
ravel_provider_capabilities, [14](#)
ravel_read_history, [14](#)
ravel_reject_action, [15](#)
ravel_run_code, [15](#)
ravel_set_api_key, [17](#)
ravel_set_bearer_token, [17](#)
ravel_set_setting, [18](#)
ravel_settings_addin, [16](#)
ravel_setup_addin, [16](#)
ravel_stage_file_write, [18](#)
ravel_suggest_diagnostics, [19](#)
ravel_summarize_model, [19](#)
ravel_summarize_object, [20](#)
ravel_verify_provider, [20](#)