

Package: helpai (via r-universe)

May 28, 2026

Type Package

Version 0.1.1

Title 'SciViews::R' - Simplified Help Pages in R Thanks to AI

Description R help page written by AI engines (LLMs).

Maintainer Philippe Grosjean <phgrosjean@sciviews.org>

Depends R (>= 4.2.0)

Imports htr2 (>= 1.0.1), roxygen2 (>= 7.3.1), utils (>= 4.2.0)

Suggests covr (>= 3.5.0), knitr (>= 1.42), rmarkdown (>= 2.21),
spelling (>= 2.2.1), testthat (>= 3.0.0)

License MIT + file LICENSE

URL <https://github.com/SciViews/helpai>,
<https://www.sciviews.org/helpai/>,
<https://sciviews.r-universe.dev/helpai>

BugReports <https://github.com/SciViews/helpai/issues>

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Encoding UTF-8

Language en-US

ByteCompile yes

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev libxml2-dev libssl-dev

Repository <https://sciviews.r-universe.dev>

Date/Publication 2025-08-31 20:52:21 UTC

RemoteUrl <https://github.com/SciViews/helpai>

RemoteRef HEAD

RemoteSha 972b6fbe873a5503156ddb2cb48415e9808598dd

Contents

helpai-package	2
ai_ask	3
chatbot1	5
chatbot2	6
chatbot3	6
Index	7

helpai-package	<i>'SciViews::R' - Simplified Help Pages in R Thanks to AI</i>
----------------	--

Description

This package builds custom R help pages, based on questions you ask to a LLM (by default, it is Ollama running locally).

Important functions

- `ai_ask()` the main function for queries,
- `ai_explain_term()` to get a definition and brief explanation on a (statistical) term.
- `ai_explain_function()` produces a short help page for an R function with explanation of its main arguments and a short example.
- `ai_explain_code()` details how a small piece of R code works.
- `ai_explain_error()` explain what an R error message means.

Author(s)

Maintainer: Philippe Grosjean <phgrosjean@sciviews.org> ([ORCID](#))

See Also

Useful links:

- <https://github.com/SciViews/helpai>
- <https://www.sciviews.org/helpai/>
- <https://sciviews.r-universe.dev/helpai>
- Report bugs at <https://github.com/SciViews/helpai/issues>

ai_ask

Ask a question to a chatbot

Description

The question is sent to the chatbot server, and the answer is returned. The default chatbot server is ollama running locally, and the default model is codestral:latest. It was also tested with mistral:7b-instruct-v0.2-q6_K for an even smaller model.

Usage

```
ai_ask(
    question,
    context = NULL,
    max_tokens = getOption("SciViews.chatbot.max_tokens",
        Sys.getenv("SCIVIEWS_CHATBOT_MAX_TOKENS", 1000L)),
    lang = getOption("SciViews_lang", "en"),
    url = getOption("SciViews.chatbot.url", Sys.getenv("SCIVIEWS_CHATBOT_URL",
        "http://localhost:11434/api/chat")),
    model = getOption("SciViews.chatbot.model", Sys.getenv("SCIVIEWS_CHATBOT_MODEL",
        "codestral:latest")),
    api_key = Sys.getenv("CONNECT_API_KEY", ""),
    verbose = FALSE
)

ai_explain_term(term, lang = getOption("SciViews_lang", "en"), ...)

ai_explain_function(
    fun,
    package = NULL,
    lang = getOption("SciViews_lang", "en"),
    ...
)

ai_explain_code(code, lang = getOption("SciViews_lang", "en"), ...)

ai_explain_error(
    code = NULL,
    error = NULL,
    lang = getOption("SciViews_lang", "en"),
    ...
)
```

Arguments

`question` A character string with the question to ask.

context	An R object used as context (usually a data frame). This is not used yet, but it should be implemented in the future.
max_tokens	The maximum number of tokens to return in the answer. By default, it is 1000.
lang	The language to use for the answer. Default is "en". You can also use "fr" for instance.
url	The URL of the chatbot server. Default is <code>http://localhost:11434/api/chat</code>
model	The LLM (large language model) to use. Default is <code>codestral</code> . Make sure you complies to its license (see https://mistral.ai/news/mistral-ai-non-production-license-mnpl/), or switch to another model that better suits your requirements.
api_key	The API key to use for connecting to the chatbot server (optional, see your server administrator).
verbose	Should more information be printed? FALSE by default.
term	The term to describe.
...	Further arguments passed to <code>ai_ask()</code> .
fun	The R function to explain.
package	The R package that provides the function.
code	A small chunk of R code to explain.
error	The error message that R returns.

Value

The answer is returned invisibly. The function is used for its side-effect of displaying the chatbot help page with the question, answer and examples

Examples

```
## Not run:
# Basic questions
ai_ask("Who are you?")
ai_ask("What is a chatbot?")
ai_ask("Qui es-tu ?")
ai_ask("Qu'est-ce que R ?")
ai_ask("Qu'est-ce que RStudio ?")
ai_ask("What is GitHub?")
ai_ask("Qu'est-ce que le R Markdown ?")
ai_ask("What is data science?")

# Inappropriate questions
ai_ask("Qu'est ce qu'un Acanthurus sp ?")
ai_ask("Raconte-moi une bonne blague.")
ai_ask("Va te faire voir !")

# Now, more complex questions
ai_ask("Comment filtrer un data frame en R?")
ai_ask("Write R code to filter a data frame.")
ai_ask("Que fait AIC()? Donne un exemple.")
ai_ask("Qu'est ce que l'hétéroscédasticite et comment la détecter dans une ANOVA à un facteur ?")
```

```

ai_ask("How to determine which model is better using an ANOVA for nested linear models?")

# Explain terms
ai_explain_term("True positive")
ai_explain_term("percentile", lang = "fr")
ai_explain_term("git push")
ai_explain_term("Quarto", lang = "fr")
ai_explain_term("boite à moustaches") # Language mismatch
ai_explain_term("boites à moustaches parallèles", lang = "fr")

# Explain R functions
ai_explain_function("mean")
ai_explain_function("fmean", lang = "fr")
ai_explain_function("collapse::fmean", lang = "fr")
ai_explain_function("glm", package = "stats", lang = "fr")
ai_explain_function("replace_na", "tidyr", lang = "fr") # collapse::replace_na() used instead!
try(ai_explain_function("nonexistingfunction")) # Error
try(ai_explain_function("apropos", package = "stats")) # Wrong package
try(ai_explain_function("apropos", package = "unknownpkg")) # Unknown package

# Explain R code
ai_explain_code("y <- c(1, 5, 7, NA, -Inf, 8)")
ai_explain_code(r"-[
mtcars |>
  filter(cyl == 4) |>
  summarise(mean_hp = mean(hp), median_disp = median(displ))]-")
ai_explain_code(r"-[
mtcars %>.%
  sfilter(., cyl == 4) %>.%
  ssummarise(., mean_hp = fmean(hp), median_disp = fmedian(displ))
]-")
ai_explain_code(r"-[
  chart(data = trees, Volume ~ Girth) +
  geom_point() +
  geom_smooth()
]-")
ai_explain_error(error = "longer object length is not a multiple of shorter object length")
ai_explain_error(error = "Error: object 'mydata' not found")
ai_explain_error(error = "Error in lenght(1:10) : could not find function \"lenght\"", lang = "fr")
ai_explain_error(code = r"-[y <- c(1, 5, 7, NA, -Inf, 8, )]-", lang = "fr")
ai_explain_error(code = r"-[trees %>.% filter(Girth > 10)]-", lang = "fr")
ai_explain_error(code = r"-[
urchin <- read("urchin", package = "data.io")
]-", error = r"-[
Error in read("urchin", package = "data.io") :
  dataset 'urchin' not found in package 'data.io'
]-", lang = "fr")

## End(Not run)

```

Description

A chatbot does not always provide reliable results. Take this with a grain of salt!

Question: No question

Answer

No answer

Examples

No example

chatbot2

SciViews chatbot

Description

A chatbot does not always provide reliable results. Take this with a grain of salt!

Question: No question

Answer

No answer

Examples

No example

chatbot3

SciViews chatbot

Description

A chatbot does not always provide reliable results. Take this with a grain of salt!

Question: No question

Answer

No answer

Examples

No example

Index

ai_ask, 3
ai_ask(), 2, 4
ai_explain_code (ai_ask), 3
ai_explain_code(), 2
ai_explain_error (ai_ask), 3
ai_explain_error(), 2
ai_explain_function (ai_ask), 3
ai_explain_function(), 2
ai_explain_term (ai_ask), 3
ai_explain_term(), 2

chatbot1, 5
chatbot2, 6
chatbot3, 6

helpai (helpai-package), 2
helpai-package, 2