Package 'statlingua'

June 2, 2025

Type Package

Title Explain Statistical Output with Large Language Models

Version 0.1.0

Description Transform complex statistical output into straightforward, understandable, and context-aware natural language descriptions using Large Language Models (LLMs), making complex analyses more accessible to individuals with varying statistical expertise. It relies on the 'ellmer' package to interface with LLM providers including OpenAI <https://openai.com/>, Google AI Studio <https://aistudio.google.com/>, and Anthropic <https://www.anthropic.com/> (API keys are required and managed via 'ellmer').

Depends R (>= 4.1.0)

License GPL (>= 2)

URL https://github.com/bgreenwell/statlingua, https://bgreenwell.github.io/statlingua/

Encoding UTF-8

RoxygenNote 7.3.2

Suggests car, ellmer (>= 0.2.0), ISLR2, knitr, lme4, lmerTest, MASS, mgcv, nlme, R6, rmarkdown, survival, tibble, tinytest

VignetteBuilder knitr

Config/Needs/website rmarkdown

NeedsCompilation no

Author Brandon M. Greenwell [aut, cre] (ORCID: <https://orcid.org/0000-0002-8120-0084>)

Maintainer Brandon M. Greenwell <greenwell.brandon@gmail.com>

Repository CRAN

Date/Publication 2025-06-02 07:50:06 UTC

Contents

`explain` *Explain statistical output*

Description

Use an LLM to explain the output from various statistical objects using straightforward, understandable, and context-aware natural language descriptions.

Usage

```
explain(
  object,
  client,
  context = NULL,
  audience = c("novice", "student", "researcher", "manager", "domain_expert"),
  verbosity = c("moderate", "brief", "detailed"),
  style = c("markdown", "html", "json", "text", "latex"),
  ...
)

## Default S3 method:
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'htest'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'lm'
explain(
  object,
  client,
```

```
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'glm'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'polr'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'lme'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'lmerMod'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
```

```
  ...
)

## S3 method for class 'glmerMod'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'gam'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'survreg'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'coxph'
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)

## S3 method for class 'rpart'
```

```
explain(
  object,
  client,
  context = NULL,
  audience = "novice",
  verbosity = "moderate",
  style = "markdown",
  ...
)
```

Arguments

`object`	An appropriate statistical object. For example, `object` can be the output from calling t.test() or glm().
`client`	A Chat object (e.g., from calling chat_openai() or [chat_gemini()][ellmer::chat_gemini)]). [ellmer::chat_gemini)]: R:ellmer::chat_gemini)
`context`	Optional character string providing additional context, such as background on the research question and information about the data.
`audience`	Character string indicating the target audience: • `"novice"` - Assumes the user has a limited statistics background (default). • `"student"` - Assumes the user is learning statistics. • `"researcher"` - Assumes the user has a strong statistical background and is familiar with common methodologies. • `"manager"` - Assumes the user needs high-level insights for decision-making. • `"domain_expert"` - Assumes the user is an expert in their own field but not necessarily in statistics.
`verbosity`	Character string indicating the desired verbosity: • `"moderate"` - Offers a balanced explanation (default). • `"brief"` - Offers a high-level summary. • `"detailed"` - Offers a comprehensive interpretation.
`style`	Character string indicating the desired output style: • `"markdown"` (default) - Output formatted as plain Markdown. • `"html"` - Output formatted as an HTML fragment. • `"json"` - Output structured as a JSON string parseable into an R list. • `"text"` - Output as plain text. • `"latex"` - Output as a LaTeX fragment.
`...`	Additional optional arguments. (Currently ignored.)

Value

An object of class `"statlingua_explanation"`. Essentially a list with the following components:

- `text` - Character string representation of the LLM's response.
- `model_type` - Character string giving the model type (e.g., `"lm"` or `"coxph"`).
- `audience` - Character string specifying the level or intended audience for the explanations.
- `verbosity` - Character string specifying the level of verbosity or level of detail of the provided explanation.

Examples

```
## Not run:
# Polynomial regression
fm1 <- lm(dist ~ poly(speed, degree = 2), data = cars)
context <- "
The data give the speed of cars (mph) and the distances taken to stop (ft).
Note that the data were recorded in the 1920s!
"
# Use Google Gemini to explain the output; requires an API key; see
# ?ellmer::chat_google_gemini for details
client <- ellmer::chat_google_gemini(echo = "none")
ex <- explain(fm1, client = client, context = context)

# Poisson regression example from ?stats::glm
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
data.frame(treatment, outcome, counts) # showing data
fm2 <- glm(counts ~ outcome + treatment, family = poisson())

# Use Google Gemini to explain the output; requires an API key; see
# ?ellmer::chat_google_gemini for details
client <- ellmer::chat_google_gemini()
explain(fm2, client = client, audience = "student", verbosity = "detailed")

## End(Not run)
```

---

print.statlingua_explanation

Print LLM explanation

Description

Print a formatted version of an LLMs explanation using `cat()`.

Usage

```
## S3 method for class 'statlingua_explanation'
print(x, ...)
```

Arguments

x	A statlingua_explanation object.
...	Additional optional arguments to be passed to `print.default()`.

Value

Invisibly returns the printed statlingua_explanation object.

summarize *Summarize statistical output*

Description

Generate text-based summaries of statistical output that can be embedded into prompts for querying Large Language Models (LLMs). Intended primarily for internal use.

Usage

```
summarize(object, ...)

## Default S3 method:
summarize(object, ...)

## S3 method for class 'htest'
summarize(object, ...)

## S3 method for class 'lm'
summarize(object, ...)

## S3 method for class 'glm'
summarize(object, ...)

## S3 method for class 'polr'
summarize(object, ...)

## S3 method for class 'lme'
summarize(object, ...)

## S3 method for class 'lmerMod'
summarize(object, ...)

## S3 method for class 'glmerMod'
summarize(object, ...)

## S3 method for class 'gam'
summarize(object, ...)

## S3 method for class 'survreg'
summarize(object, ...)

## S3 method for class 'coxph'
summarize(object, ...)

## S3 method for class 'rpart'
summarize(object, ...)
```

Arguments

`object`	An object for which a summary is desired (e.g., a glm object).
`...`	Additional optional arguments. (Currently ignored.)

Value

A character string summarizing the statistical output.

See Also

summary().

Examples

```
tt <- t.test(1:10, y = c(7:20))
summarize(tt)  # prints output as a character string
cat(summarize(tt))  # more useful for reading
```

Index