Package 'NaileR'

August 1, 2024

Title Interpreting Latent Variables with AI

Version 1.1.0

Description A small package designed for interpreting continuous and categorical latent variables. You provide a data set with a latent variable you want to understand and some other explanatory variables. It provides a description of the latent variable based on the explanatory variables. It also provides a name to the latent variable.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.1

Imports dplyr, stringr, FactoMineR, glue, ollamar, magrittr

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Nel Hervé [aut],
Sébastien Lê [aut, cre] (<https://orcid.org/0000-0001-8814-6714>)

Maintainer Sébastien Lê <sebastien.le@institut-agro.fr>

Repository CRAN

Date/Publication 2024-08-01 08:40:06 UTC

Contents

`agri_studies` *Agribusiness studies survey*

Description

These data were collected after a Q-method-like survey on students' expectations of agribusiness studies. Participants had to rank how much they agreed with 38 statements about possible benefits from agribusiness studies; then, they were asked personal questions.

Usage

```
agri_studies
```

Format

A data frame with 53 rows (participants) and 42 columns (questions):

- columns 1-38: statements about agribusiness studies
- columns 39-42: personal information

Source

Juliette LE COLLONNIER and Lou ROBERT, students at l'Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(agri_studies)

res_mca_agri <- FactoMineR::MCA(agri_studies, quali.sup = 39:42,
level.ventil = 0.05, graph = FALSE)
agri_work <- res_mca_agri$ind$coord |> as.data.frame()
agri_work <- agri_work[,1] |> cbind(agri_studies)

intro_agri <- "These data were collected after a survey
on students' expectations of agribusiness studies.
```

```
Participants had to rank how much they agreed with 38 statements
about possible benefits from agribusiness studies;
then, they were asked personal questions."
intro_agri <- gsub('\n', ' ', intro_agri) |>
stringr::str_squish()

res_agri <- nail_condes(agri_work, num.var = 1,
introduction = intro_agri)
cat(res_agri$response)
```

`beard` *Beard descriptions*

Description

These data refer to 8 types of beards. Each beard was evaluated by 62 assessors (except beard 8 which only had 60 evaluations).

Usage

```
beard
```

Format

A data frame with 494 rows and 2 columns:

- the types of beards;
- the words used to describe them.

Source

Applied mathematics department, Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

data(beard)
beard[1:8,]
```

beard_cont *Beard descriptions*

Description

These data refer to 8 types of beards. Each beard was evaluated by 62 assessors (except beard 8 which only had 60 evaluations).

Usage

```
beard_cont
```

Format

A contingency table (data frame) with 8 rows and 337 columns:

- rows are the types of beards;
- columns are the words used at least once to describe them.

Source

Applied mathematics department, Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(beard_cont)

FactoMineR::descfreq(beard_cont)

intro_beard <- 'A survey was conducted about beards
and 8 types of beards were described.
In the data that follow, beards are named B1 to B8.'
intro_beard <- gsub('\n', ' ', intro_beard) |>
stringr::str_squish()

req_beard <- 'Please give a name to each beard
and summarize what makes this beard unique.'
req_beard <- gsub('\n', ' ', req_beard) |>
stringr::str_squish()

res_beard <- nail_descfreq(beard_cont,
introduction = intro_beard,
request = req_beard)
cat(res_beard$response)
```

beard_wide *Beard descriptions*

Description

These data refer to 8 types of beards. They come from a subset of the original "beard" dataset. Each beard was evaluated by 62 assessors (except beard 8 which only had 60 evaluations).

Usage

```
beard_wide
```

Format

A data frame with 8 rows and 24 columns:

- rows are the types of beards;
- columns are the assessors' opinions.

Source

Applied mathematics department, Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(beard_wide)

intro_beard <- "As a barber, you make
recommendations based on consumers comments.
Examples of consumers descriptions of beards
are as follows."
intro_beard <- gsub('\n', ' ', intro_beard) |>
stringr::str_squish()

res <- nail_sort(beard_wide[,1:5], name_size = 3,
stimulus_id = "beard", introduction = intro_beard,
measure = 'the description was')

res$dta_sort
cat(res$prompt_llm[[1]])
```

boss *Ideal boss survey*

Description

These data were collected after a Q-method-like survey on participants' perception of an "ideal boss". Participants had to rank how much they agreed with 30 statements about an ideal boss; then, they were asked personal questions.

Usage

```
boss
```

Format

A data frame with 73 rows (participants) and 39 columns (questions):

- columns 1-30: statements about the ideal boss
- columns 31-39: personal information

Source

Florian LECLERE and Marianne ANDRE, students at l'Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(FactoMineR)
library(NaileR)
data(boss)
res_mca_boss <- MCA(boss, quali.sup = 31:39,
ncp = 30, level.ventil = 0.05, graph = FALSE)
res_hcpc_boss <- HCPC(res_mca_boss, nb.clust = 4, graph = FALSE)
don_clust_boss <- res_hcpc_boss$data.clust

intro_boss <- 'A study on "the ideal boss" was led on 73 participants.
The study had 2 parts. In the first part,
participants were given statements about the ideal boss
(starting with "My ideal boss...").
They had to rate, on a scale from 1 to 5,
how much they agreed with the statements;
1 being "Strongly disagree", 3 being "neutral"
and 5 being "Strongly agree".
In the second part, they were asked for personal information:
work experience, age, etc.
Participants were then split into groups based on their answers.'
intro_boss <- gsub('\n', ' ', intro_boss) |>
```

```
stringr::str_squish()

req_boss <- "Please describe, for each group, their ideal boss.
Then, give each group a new name, based on your conclusions."
req_boss <- gsub('\n', ' ', req_boss) |>
stringr::str_squish()


res_boss <- nail_catdes(don_clust_boss, num.var = 40,
introduction = intro_boss, request = req_boss,
isolate.groups = FALSE, drop.negative = TRUE)
res_boss$response |> cat()
```

`dist_mat_llm` *LLM distance matrix*

Description

Compute a distance matrix between randomly-generated responses to an LLM prompt.

Usage

```
dist_mat_llm(ppt, n, per_miss = 0)
```

Arguments

`ppt`	an LLM prompt.
`n`	the number of responses to be generated.
`per_miss`	the proportion of missing values in the final matrix (between 0 and 1; 0 by default).

Details

The final percentage of missing values might differ from the per_miss parameter value; rather than a percentage of values being turned to NA, each value has a per_miss probability of being NA.

Value

A list containing:

- a list of the LLM results for each iteration;
- a distance matrix.

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

data(iris)

intro_iris <- "A study measured various parts of iris flowers
from 3 different species: setosa, versicolor and virginica.
I will give you the results from this study.
You will have to identify what sets these flowers apart."
intro_iris <- gsub('\n', ' ', intro_iris) |>
stringr::str_squish()

req_iris <- "Please explain what makes each species distinct.
Also, tell me which species has the biggest flowers,
and which species has the smallest."
req_iris <- gsub('\n', ' ', req_iris) |>
stringr::str_squish()

res_iris <- nail_catdes(iris, num.var = 5,
introduction = intro_iris, request = req_iris)

dist_mat_llm(res_iris$prompt, n = 5, per_miss = 0)
```

---

dist_ref_llm *LLM response consistency*

Description

Compute distances between an LLM response of interest and some other responses to the same prompt.

Usage

```
dist_ref_llm(ppt, ref, n)
```

Arguments

ppt	an LLM prompt.
ref	the reference response.
n	the number of new responses to be generated.

Value

A list containing:

- a list with the newly-generated prompts;
- a vector of distances to the reference response.

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

data(iris)

intro_iris <- "A study measured various parts of iris flowers
from 3 different species: setosa, versicolor and virginica.
I will give you the results from this study.
You will have to identify what sets these flowers apart."
intro_iris <- gsub('\n', ' ', intro_iris) |>
stringr::str_squish()

req_iris <- "Please explain what makes each species distinct.
Also, tell me which species has the biggest flowers,
and which species has the smallest."
req_iris <- gsub('\n', ' ', req_iris) |>
stringr::str_squish()

res_iris <- nail_catdes(iris, num.var = 5,
introduction = intro_iris, request = req_iris)

dist_ref_llm(res_iris$prompt, res_iris$response, n = 5)
```

---

glossophobia *Glossophobia survey*

Description

These data were collected after a Q-method-like survey on participants' feelings about speaking in public. Participants had to rank how much they agreed with 25 descriptions of speaking in public; then, they were asked personal questions.

Usage

```
glossophobia
```

Format

A data frame with 139 rows (participants) and 41 columns (questions):

- columns 1-25: descriptions of speaking in public
- columns 26-41: personal information

Source

Elina BIAU and Théo LEDAIN, students at l'Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(glossophobia)

res_mca_phobia <- FactoMineR::MCA(glossophobia, quali.sup = 26:41,
level.ventil = 0.05, graph = FALSE)
phobia_work <- res_mca_phobia$ind$coord |> as.data.frame()
phobia_work <- phobia_work[,1] |> cbind(glossophobia)

intro_phobia <- "These data were collected after a survey
on participants' feelings about speaking in public.
Participants had to rank how much they agreed with
25 descriptions of speaking in public;
then, they were asked personal questions."
intro_phobia <- gsub('\n', ' ', intro_phobia) |>
stringr::str_squish()

res_phobia <- nail_condes(phobia_work, num.var = 1,
introduction = intro_phobia)
cat(res_phobia$response)
```

---

local_food *Local food systems survey*

Description

These data were collected after a Q-method-like survey on sustainable food systems. Participants had to rank how acceptable they found 45 statements about a sustainable food system; then, they were asked if they agreed with 14 other statements.

Usage

```
local_food
```

Format

A data frame with 573 rows (participants) and 63 columns (questions):

- columns 1-45 statements about food systems
- columns 46-59 opinions
- columns 60-63 personal information

Source

Applied mathematics department, Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(FactoMineR)
library(NaileR)
data(local_food)

res_mca_food <- MCA(local_food, quali.sup = 46:63,
ncp = 100, level.ventil = 0.05, graph = FALSE)
res_hcpc_food <- HCPC(res_mca_food, nb.clust = 3, graph = FALSE)
don_clust_food <- res_hcpc_food$data.clust

intro_food <- 'A study on sustainable food systems
was led on several French participants.
This study had 2 parts. In the first part,
participants had to rate how acceptable
"a food system that..." (e.g, "a food system that
only uses renewable energy") was to them.
In the second part, they had to say
if they agreed or disagreed with some statements.'
intro_food <- gsub('\n', ' ', intro_food) |>
stringr::str_squish()

req_food <- 'I will give you the answers from one group.
Please explain who the individuals of this group are,
what their beliefs are.
Then, give this group a new name,
and explain why you chose this name.
Do not use 1st person ("I", "my"...) in your answer.'
req_food <- gsub('\n', ' ', req_food) |>
stringr::str_squish()

res_food <- nail_catdes(don_clust_food, num.var = 64,
introduction = intro_food,
request = req_food,
isolate.groups = TRUE, drop.negative = TRUE)
res_food[[1]]$response |> cat()
```

---

nail_catdes *Interpret a categorical latent variable*

Description

Generate an LLM response to analyze a categorical latent variable.

Usage

```
nail_catdes(
  dataset,
  num.var,
  introduction = "",
  request = NULL,
  model = "llama3",
  isolate.groups = FALSE,
  drop.negative = FALSE,
  proba = 0.05,
  row.w = NULL,
  generate = TRUE
)
```

Arguments

dataset	a data frame made up of at least one categorical variable and a set of quantitative variables and/or categorical variables.
num.var	the index of the variable to be characterized.
introduction	the introduction for the LLM prompt.
request	the request made to the LLM.
model	the model name ('llama3' by default).
isolate.groups	a boolean that indicates whether to give the LLM a single prompt, or one prompt per category. Recommended with long catdes results.
drop.negative	a boolean that indicates whether to drop negative v.test values for interpretation (keeping only positive v.tests). Recommended with long catdes results.
proba	the significance threshold considered to characterize the categories (by default 0.05).
row.w	a vector of integers corresponding to an optional row weights (by default, a vector of 1 for uniform row weights)
generate	a boolean that indicates whether to generate the LLM response. If FALSE, the function only returns the prompt.

Details

This function directly sends a prompt to an LLM. Therefore, to get a consistent answer, we highly recommend to customize the parameters introduction and request and add all relevant information on your data for the LLM. We also recommend renaming the columns with clear, unshortened and unambiguous names.

Additionally, if isolate.groups = TRUE, you will need an introduction and a request that take into account the fact that only one group is analyzed at a time.

Value

A data frame, or a list of data frames, containing the LLM's prompt and response (if generate = TRUE).

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

### Example 1: Fisher's iris ###
library(NaileR)
data(iris)

intro_iris <- "A study measured various parts of iris flowers
from 3 different species: setosa, versicolor and virginica.
I will give you the results from this study.
You will have to identify what sets these flowers apart."
intro_iris <- gsub('\n', ' ', intro_iris) |>
stringr::str_squish()

req_iris <- "Please explain what makes each species distinct.
Also, tell me which species has the biggest flowers,
and which species has the smallest."
req_iris <- gsub('\n', ' ', req_iris) |>
stringr::str_squish()

res_iris <- nail_catdes(iris, num.var = 5,
introduction = intro_iris, request = req_iris)
cat(res_iris$response)


### Example 2: food waste dataset ###

library(FactoMineR)

data(waste)
waste <- waste[-14]    # no variability on this question

set.seed(1)
res_mca_waste <- MCA(waste, quali.sup = c(1,2,50:76),
ncp = 35, level.ventil = 0.05, graph = FALSE)
plot.MCA(res_mca_waste, choix = "ind",
invisible = c("var", "quali.sup"), label = "none")
res_hcpc_waste <- HCPC(res_mca_waste, nb.clust = 3, graph = FALSE)
plot.HCPC(res_hcpc_waste, choice = "map", draw.tree = FALSE,
ind.names = FALSE)
don_clust_waste <- res_hcpc_waste$data.clust

intro_waste <- 'These data were collected
after a survey on food waste,
with participants describing their habits.'
intro_waste <- gsub('\n', ' ', intro_waste) |>
stringr::str_squish()

req_waste <- 'Please summarize the characteristics of each group.
Then, give each group a new name, based on your conclusions.
Finally, give each group a grade between 0 and 10,
```

```
based on how wasteful they are with food:
0 being "not at all", 10 being "absolutely".'
req_waste <- gsub('\n', ' ', req_waste) |>
stringr::str_squish()

res_waste <- nail_catdes(don_clust_waste,
num.var = ncol(don_clust_waste),
introduction = intro_waste, request = req_waste,
drop.negative = TRUE)

cat(res_waste$response)


### Example 3: local_food dataset ###

data(local_food)

set.seed(1)
res_mca_food <- MCA(local_food, quali.sup = 46:63,
ncp = 100, level.ventil = 0.05, graph = FALSE)
plot.MCA(res_mca_food, choix = "ind",
invisible = c("var", "quali.sup"), label = "none")
res_hcpc_food <- HCPC(res_mca_food, nb.clust = 3, graph = FALSE)
plot.HCPC(res_hcpc_food, choice = "map", draw.tree = FALSE,
ind.names = FALSE)
don_clust_food <- res_hcpc_food$data.clust

intro_food <- 'A study on sustainable food systems
was led on several French participants.
This study had 2 parts. In the first part,
participants had to rate how acceptable
"a food system that..." (e.g, "a food system that
only uses renewable energy") was to them.
In the second part, they had to say
if they agreed or disagreed with some statements.'
intro_food <- gsub('\n', ' ', intro_food) |>
stringr::str_squish()

req_food <- 'I will give you the answers from one group.
Please explain who the individuals of this group are,
what their beliefs are.
Then, give this group a new name,
and explain why you chose this name.
Do not use 1st person ("I", "my"...) in your answer.'
req_food <- gsub('\n', ' ', req_food) |>
stringr::str_squish()

res_food <- nail_catdes(don_clust_food, num.var = 64,
introduction = intro_food,
request = req_food,
isolate.groups = TRUE, drop.negative = TRUE)

res_food[[1]]$response |> cat()
```

```
res_food[[2]]$response |> cat()
res_food[[3]]$response |> cat()
```

`nail_condes` *Interpret a continuous latent variable*

Description

Generate an LLM response to analyze a continuous latent variable.

Usage

```
nail_condes(
  dataset,
  num.var,
  introduction = "",
  request = NULL,
  model = "llama3",
  quanti.threshold = 0,
 quanti.cat = c("Significantly above average", "Significantly below average", "Average"),
  weights = NULL,
  proba = 0.05,
  generate = TRUE
)
```

Arguments

`dataset`	a data frame made up of at least one quantitative variable and a set of quantitative variables and/or categorical variables.
`num.var`	the index of the variable to be characterized.
`introduction`	the introduction for the LLM prompt.
`request`	the request made to the LLM.
`model`	the model name ('llama3' by default).
`quanti.threshold`	the threshold above (resp. below) which a scaled variable is considered significantly above (resp.below) the average. Used when converting continuous variables to categorical ones.
`quanti.cat`	a vector of the 3 possible categories for continuous variables converted to categorical ones according to the threshold. Default is "above average", "below average" and "average".
`weights`	weights for the individuals (see `FactoMineR::condes()`).
`proba`	the significance threshold considered to characterize the category (by default 0.05).
`generate`	a boolean that indicates whether to generate the LLM response. If FALSE, the function only returns the prompt.

Details

This function directly sends a prompt to an LLM. Therefore, to get a consistent answer, we highly recommend to customize the parameters introduction and request and add all relevant information on your data for the LLM. We also recommend renaming the columns with clear, unshortened and unambiguous names.

Value

A data frame containing the LLM's prompt and response (if generate = TRUE).

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

### Example 1: decathlon dataset ###

library(FactoMineR)
data(decathlon)

names(decathlon) <- c('Time taken to complete the 100m',
'Distance reached for the long jump',
'Distance reached for the shot put',
'Height reached for the high jump',
'Time taken to complete the 400m',
'Time taken to complete the 110m hurdle',
'Distance reached for the discus',
'Height reached for the pole vault',
'Distance reached for the javeline',
'Time taken to complete the 1500 m',
'Rank/Counter-performance indicator',
'Points', 'Competition')

res_pca_deca <- FactoMineR::PCA(decathlon,
quanti.sup = 11:12, quali.sup = 13, graph = FALSE)
plot.PCA(res_pca_deca, choix = 'var')
deca_work <- res_pca_deca$ind$coord |> as.data.frame()
deca_work <- deca_work[,1] |> cbind(decathlon)

intro_deca <- "A study was led on athletes
participating in a decathlon event.
Their performance was assessed on each part of the decathlon,
and they were all placed on an unidimensional scale."
intro_deca <- gsub('\n', ' ', intro_deca) |>
stringr::str_squish()

res_deca <- nail_condes(deca_work, num.var = 1,
quanti.threshold = 1, quanti.cat = c('High', 'Low', 'Average'),
introduction = intro_deca)

cat(res_deca$response)
```

```
### Example 2: agri_studies dataset ###

data(agri_studies)

set.seed(1)
res_mca_agri <- FactoMineR::MCA(agri_studies, quali.sup = 39:42,
level.ventil = 0.05, graph = FALSE)
plot.MCA(res_mca_agri, choix = 'ind',
invisible = c('var', 'quali.sup'), label = 'none')

agri_work <- res_mca_agri$ind$coord |> as.data.frame()
agri_work <- agri_work[,1] |> cbind(agri_studies)

intro_agri <- "These data were collected after a survey
on students' expectations of agribusiness studies.
Participants had to rank how much they agreed with 38 statements
about possible benefits from agribusiness studies;
then, they were asked personal questions."
intro_agri <- gsub('\n', ' ', intro_agri) |>
stringr::str_squish()

res_agri <- nail_condes(agri_work, num.var = 1,
introduction = intro_agri)
cat(res_agri$response)


### Example 3: glossophobia dataset ###

data(glossophobia)

set.seed(1)
res_mca_phobia <- FactoMineR::MCA(glossophobia,
quali.sup = 26:41, level.ventil = 0.05, graph = FALSE)
plot.MCA(res_mca_phobia, choix = 'ind',
invisible = c('var', 'quali.sup'), label = 'none')

phobia_work <- res_mca_phobia$ind$coord |> as.data.frame()
phobia_work <- phobia_work[,1] |> cbind(glossophobia)

intro_phobia <- "These data were collected after a survey
on participants' feelings about speaking in public.
Participants had to rank how much they agreed with
25 descriptions of speaking in public;
then, they were asked personal questions."
intro_phobia <- gsub('\n', ' ', intro_phobia) |>
stringr::str_squish()

res_phobia <- nail_condes(phobia_work, num.var = 1,
introduction = intro_phobia)
cat(res_phobia$response)
```

```
### Example 4: beard_cont dataset ###

data(beard_cont)

set.seed(1)
res_ca_beard <- FactoMineR::CA(beard_cont, graph = FALSE)
plot.CA(res_ca_beard, invisible = 'col')

beard_work <- res_ca_beard$row$coord |> as.data.frame()
beard_work <- beard_work[,1] |> cbind(beard_cont)

intro_beard <- "These data refer to 8 types of beards.
Each beard was evaluated by 62 assessors."
intro_beard <- gsub('\n', ' ', intro_beard) |>
stringr::str_squish()

req_beard <- "Please explain what differentiates beards
on both sides of the scale.
Then, give the scale a name."
req_beard <- gsub('\n', ' ', req_beard) |>
stringr::str_squish()

res_beard <- nail_condes(beard_work, num.var = 1,
quanti.threshold = 0.5, quanti.cat = c('Very often', 'Never', 'Sometimes'),
introduction = intro_beard, request = req_beard,
generate = FALSE)

cat(res_beard$prompt)

ppt <- stringr::str_replace_all(res_beard$prompt, 'individuals', 'beards')
cat(ppt)

res_beard <- ollamar::generate(model = 'llama3', prompt = ppt, output = 'df')

cat(res_beard$response)
```

`nail_descfreq` *Interpret the rows of a contingency table*

Description

Describes the rows of a contingency table. For each row, this description is based on the columns of the contingency table that are significantly related to it.

Usage

```
nail_descfreq(
  dataset,
  introduction = "",
```

```
  request = NULL,
  model = "llama3",
  isolate.groups = FALSE,
  by.quali = NULL,
  proba = 0.05,
  generate = TRUE
)
```

Arguments

dataset	a data frame corresponding to a contingency table.
introduction	the introduction for the LLM prompt.
request	the request made to the LLM.
model	the model name ('llama3' by default).
isolate.groups	a boolean that indicates whether to give the LLM a single prompt, or one prompt per row. Recommended if the contingency table has a great number of rows.
by.quali	a factor used to merge the data from different rows of the contingency table; by default NULL and each row is characterized.
proba	the significance threshold considered to characterize the category (by default 0.05).
generate	a boolean that indicates whether to generate the LLM response. If FALSE, the function only returns the prompt.

Details

This function directly sends a prompt to an LLM. Therefore, to get a consistent answer, we highly recommend to customize the parameters introduction and request and add all relevant information on your data for the LLM.

Additionally, if isolate.groups = TRUE, you will need an introduction and a request that take into account the fact that only one group is analyzed at a time.

Value

A data frame, or a list of data frames, containing the LLM's prompt and response (if generate = TRUE).

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

### Example 1: beard dataset ###

data(beard_cont)

intro_beard_iso <- 'A survey was conducted about beards
and 8 types of beards were described.
I will give you the results for one type of beard.'
```

```
intro_beard_iso <- gsub('\n', ' ', intro_beard_iso) |>
stringr::str_squish()

req_beard_iso <- 'Please give a name to this beard
and summarize what makes this beard unique.'
req_beard_iso <- gsub('\n', ' ', req_beard_iso) |>
stringr::str_squish()

res_beard <- nail_descfreq(beard_cont,
introduction = intro_beard_iso,
request = req_beard_iso,
isolate.groups = TRUE, generate = FALSE)

res_beard$prompt[1]
res_beard$prompt[2]

intro_beard <- 'A survey was conducted about beards
and 8 types of beards were described.
In the data that follow, beards are named B1 to B8.'
intro_beard <- gsub('\n', ' ', intro_beard) |>
stringr::str_squish()

req_beard <- 'Please give a name to each beard
and summarize what makes this beard unique.'
req_beard <- gsub('\n', ' ', req_beard) |>
stringr::str_squish()

res_beard <- nail_descfreq(beard_cont,
introduction = intro_beard,
request = req_beard)
cat(res_beard$response)

text <- res_beard$response
titles <- stringr::str_extract_all(text, "\\*\\*B[0-9]+: [^\\*\\*]+\\*\\*")[[1]]

titles

# for the following code to work, the response must have the beards'
# new names with this format: **B1: The Nice beard**, etc.

titles <- stringr::str_replace_all(titles, "\\*\\*", "")  # remove asterisks
names <- stringr::str_extract(titles, ": .+")
names <- stringr::str_replace_all(names, ": ", "")  # remove the colon and space

rownames(beard_cont) <- names

library(FactoMineR)

res_ca_beard <- CA(beard_cont, graph = F)
plot.CA(res_ca_beard, invisible = "col")


### Example 2: children dataset ###
```

```
data(children)

children <- children[1:14, 1:5] |> t() |> as.data.frame()
rownames(children) <- c('No education', 'Elementary school',
'Middle school', 'High school', 'University')

intro_children <- 'The data used here is a contingency table
that summarizes the answers
given by different categories of people to the following question:
"according to you, what are the reasons that can make
a woman of a couple hesitate to have children?".
Each row corresponds to a level of education, and columns are reasons.'
intro_children <- gsub('\n', ' ', intro_children) |>
stringr::str_squish()

req_children <- "Please explain the main differences
between more educated and less educated couples,
when it comes to hesitating to have children."
req_children <- gsub('\n', ' ', req_children) |>
stringr::str_squish()

res_children <- nail_descfreq(children,
introduction = intro_children, request = req_children)

cat(res_children$response)
```

`nail_sort` *Sort textual data*

Description

Group textual data according to their similarity, in a context in which the assessors have commented on a set of stimuli.

Usage

```
nail_sort(
  dta_text,
  name_size = 3,
  stimulus_id = "stimulus",
  introduction = "",
  measure = "",
  nb_max = 6
)
```

Arguments

dta_text	a data frame where each row is a stimulus and each column is an assessor.
name_size	the maximum number of words in a group name created by the LLM.
stimulus_id	the nature of the stimulus. Customizing it is highly recommended.
introduction	the introduction to the LLM prompt.
measure	the type of measure used in the experiment.
nb_max	the maximum number of clusters the LLM can form per assessor.

Details

This function uses a while loop to ensure that the LLM gives the right number of groups. Therefore, customizing the stimulus ID, prompt introduction and measure is highly recommended; a clear prompt can help the LLM finish its task faster.

Value

A list consisting of:

- a list of prompts (one per assessor);
- a data frame with the group names.

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(beard_wide)

intro_beard <- "As a barber, you make
recommendations based on consumers comments.
Examples of consumers descriptions of beards
are as follows."
intro_beard <- gsub('\n', ' ', intro_beard) |>
stringr::str_squish()

res <- nail_sort(beard_wide[,1:5], name_size = 3,
stimulus_id = "beard", introduction = intro_beard,
measure = 'the description was')

res$dta_sort
cat(res$prompt_llm[[1]])
```

nutriscore *Nutri-score survey*

Description

These data were collected after a survey on the nutri-score. Participants were asked various questions about their views on the nutri-score, and about their eating habits.

Usage

```
nutriscore
```

Format

A data frame with 112 rows (participants) and 36 columns (questions).

Source

Anaëlle YANNIC and Jessie PICOT, students at l'Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
library(FactoMineR)

data(nutriscore)

res_mca_nutriscore <- MCA(nutriscore, quali.sup = 17:36,
ncp = 15, level.ventil = 0.05, graph = FALSE)

res_hcpc_nutriscore <- HCPC(res_mca_nutriscore, nb.clust = 3,
graph = FALSE)
don_clust_nutriscore <- res_hcpc_nutriscore$data.clust

intro_nutri <- 'These data were collected after a survey
on the nutri-score. Participants were asked
various questions about their views on the nutri-score,
and about their eating habits.
Participants were split into groups according to their answers.'
intro_nutri <- gsub('\n', ' ', intro_nutri) |>
stringr::str_squish()

req_nutri <- 'Please summarize the characteristics
of each group. Then, give each group a new name,
based on your conclusions.'
req_nutri <- gsub('\n', ' ', req_nutri)|>
stringr::str_squish()
```

```
res_nutriscore <- nail_catdes(don_clust_nutriscore, num.var = 37,
introduction = intro_nutri, request = req_nutri,
drop.negative = TRUE)

cat(res_nutriscore$response)
```

`quality` *Perception of food quality*

Description

These data were collected after a study on the perception of food quality. Participants were given 9 French logos; they had to rate, on a scale from 0 (not at all) to 10 (absolutely), how much a product bearing them aligned with their own perception of quality.

Usage

```
quality
```

Format

A data frame with 55 rows and 9 columns. Here is the list of logos:

- AB: organic;
- Label Rouge: superior quality (from the taste, process, packaging...);
- FairTrade: decent wages and working conditions for the producers;
- Bleu Blanc Coeur: diverse and balanced diet for the livestock;
- AOC: controlled designation of origin;
- Produit en Bretagne: processed in Brittany;
- Viandes de France: livestock bred, grown and slaughtered in France, with respectful living conditions;
- Nourri sans OGM: no GMOs in livestock food;
- Médailles Agro: a prize won at a yearly contest based on taste.

Source

Sébastien Lê, applied mathematics department, Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
data(quality)

colnames(quality) <- c("Agriculture biologique",
"Label Rouge",
"FairTrade",
"Bleu Blanc Coeur",
"Appelation d'origine contrôlée",
"Produit en Bretagne",
"Viandes de France",
"Nourri sans OGM",
"Médailles Agro")

res_pca_quality <- FactoMineR::PCA(quality, graph = FALSE)
quali_work <- res_pca_quality$ind$coord |> as.data.frame()
quali_work <- quali_work[,1] |> cbind(quality)

intro_quali <- "These data were collected after a study
on the perception of food quality.
Participants were given 9 French logos;
they had to rate, on a scale from 0 (not at all)
to 10 (absolutely), how much a product bearing them
aligned with their own perception of quality."
intro_quali <- gsub('\n', ' ', intro_quali) |>
stringr::str_squish()

res_quality <- nail_condes(quali_work, num.var = 1,
quanti.cat = c('Higher quality', 'Lower quality', 'Neutral'),
introduction = intro_quali, generate = FALSE)

ppt <- gsub('characteristics', 'opinions', res_quality$prompt)

res_quality <- ollamar::generate('llama3', ppt, output = 'df')

cat(res_quality$response)
```

---

sim_llm *LLM text similarity*

Description

Compute a similarity score, on a scale ranging from 0 (totally different) to 100 (the exact same), between two character strings.

Usage

```
sim_llm(textA, textB)
```

Arguments

textA, textB	two character strings.

Details

The similarity score is generated by an LLM. Therefore, the result might vary if the function is run several times.

Value

An integer between 0 and 100.

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

textA <- "Participant A was described as a nice, outgoing man, with a friendly attitude."
textB <- "Participant A was an extroverted and caring individual."

sim_llm(textA, textB)
```

---

waste *Food waste survey*

Description

These data were collected after a survey on food waste, with participants describing their habits.

Usage

```
waste
```

Format

A data frame with 180 rows (participants) and 77 columns (questions).

Source

Héloïse BILLES and Amélie RATEAU, students at l'Institut Agro Rennes-Angers

Examples

```
# Processing time is often longer than ten seconds
# because the function uses a large language model.

library(NaileR)
library(FactoMineR)
data(waste)
waste <- waste[-14]

res_mca_waste <- MCA(waste, quali.sup = c(1,2,50:76),
ncp = 35, level.ventil = 0.05, graph = FALSE)
res_hcpc_waste <- HCPC(res_mca_waste, nb.clust = 3, graph = FALSE)
don_clust_waste <- res_hcpc_waste$data.clust

intro_waste <- 'These data were collected
after a survey on food waste,
with participants describing their habits.'
intro_waste <- gsub('\n', ' ', intro_waste) |>
stringr::str_squish()

req_waste <- 'Please summarize the characteristics of each group.
Then, give each group a new name, based on your conclusions.
Finally, give each group a grade between 0 and 10,
based on how wasteful they are with food:
0 being "not at all", 10 being "absolutely".'
req_waste <- gsub('\n', ' ', req_waste) |>
stringr::str_squish()

res_waste <- nail_catdes(don_clust_waste,
num.var = ncol(don_clust_waste),
introduction = intro_waste, request = req_waste,
drop.negative = TRUE)
cat(res_waste$response)
```

Index