Package 'SocialFacts'

July 24, 2026

Title Average Attributable Fraction and Other Relevant Social Sciences Indicators

Version 1.3.0

Description Compute Average Attributable Fraction (AAF) and produce a GT table with Odds.ratio, Average Marginal Effects and AAF, with confidence interval and p.value. Also compute other metrics such as Yule's Q and Cramer's V. For more details, see Fergusion and al. (2024) <doi:10.1007/s10654-024-01129-1>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports averisk, broom.helpers, dplyr, graphPAF (>= 2.0.1), gt, psych, gtsummary, rcompanion, scales, stats, stringr, weights, questionr, katex, marginaleffects

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/Geminy3/SocialFacts, https://geminy3.github.io/SocialFacts/

BugReports https://github.com/Geminy3/SocialFacts/issues

NeedsCompilation no

Author Alioscha Massein [aut, cre, cph] (ORCID: <https://orcid.org/0009-0005-5791-1045>), Brice Lefèvre [aut] (ORCID: <https://orcid.org/0000-0002-2482-7104>)

Maintainer Alioscha Massein <alioscha.massein@cnrs.fr>

Repository CRAN

Date/Publication 2026-07-24 10:30:16 UTC

Contents

`cramer_tab` *get_cramer*

Description

Get Cramer or YuleQ's tab

Usage

```
cramer_tab(var = NULL, metric = "cramer", data = NULL, weight = NULL)
```

Arguments

`var`	A list of variable name you want to compute "metric" on
`metric`	Choose between 'Cramer' or 'YuleQ' parameter to select one of this metrics
`data`	The data table you want to compute 'metric' parameter on
`weight`	The weight vector if there is one in the dataframe

Value

A list of gt information

Examples

```
data <- graphPAF::Hordaland_data
cramer_tab(c("y", "urban.rural"), metric = 'cramer', data = data)
```

get_AAF *get_AAF*

Description

Get the AAF

Usage

```
get_AAF(
  model = NULL,
  nvar = NULL,
  name_var = NULL,
  vars_dep = NULL,
  data = NULL,
  nbootstrap = 50,
  nperm = 100
)
```

Arguments

model	The model you want to compute AAF on
nvar	You can choose this integer parameter if you prefer to add explanatory variables in the order they appear in the model. You don't have to use name_var parameter.
name_var	A list of explanatory variables you want to compute AAF on
vars_dep	A string with the name of the variable you want to explain
data	The data you used to fit the model (with the variable in name_var and vars_dep)
nbootstrap	An integer with the n bootsrap replication you want. By default, 50.
nperm	Number of permutation between modalities and variable to compute the AAF

Value

A sf table with the result of the AAF computation

Examples

```
set.seed(9876)
data <- data.frame(
   "var_dep" = as.factor(sample(c(rep_len(c(0, 1), length.out = 1000)), size = 100)),
   "var_1" = as.factor(sample(c(rep_len(c(1, 2, 3), length.out = 1000)), size = 100)),
   "var_2" = as.factor(sample(c(rep_len(c(3, 1, 2), length.out = 100)), size = 100))
   )
glmmodel <- glm(var_dep ~ var_1 + var_2,
                data = data, family = binomial("logit"))
```

```
res <- get_AAF(model = glmmodel, nvar = 3, vars_dep = "var_dep", data = data,
               nbootstrap = 5)
res <- get_AAF(model = glmmodel,
               name_var = c("var_1", "var_2"),
              vars_dep = "var_dep", data = data, nbootstrap = 5)
```

`get_list` *get_list*

Description

Get a list with the names of the variables pass as parameter

Usage

```
get_list(name_var = NULL)
```

Arguments

`name_var`	A list of variable you want to pass a name for a list

Value

a list of zero named after the variable in name_var parameter

Examples

```
vec <- get_list(name_var = c("var1", "var2"))
```

`get_yuleq` *Get Yule's Q*

Description

Get Yule's Q

Usage

```
get_yuleq(
  vars_dep = NULL,
  name_var = c(),
  data = NULL,
  source = "",
  alpha = 0.01,
  weight = NULL
)
```

Arguments

vars_dep	A string with name of the dependant variable to use as one variable for the 2x2 table
name_var	A vector of explanatory variable you want to compute Yule's Q on
data	A dataframe with the variable named before
source	A string with the name of the data source
alpha	A integer with the alpha value for fisher.test and IC
weight	A vector with weigths corresponding to the dataframe individuals weights

Value

A gt table with the Yule's computed for the variable in name_var

Examples

```
data <- graphPAF::Hordaland_data
get_yuleq(vars_dep = "y", name_var = c("y",  "urban.rural"),
                    data = data, source = "dataSource - Year",
                    alpha = 0.01)
```

---

result_tab *result_tab*

Description

Get the result table of OR, AME and AAF

Usage

```
result_tab(
  model = NULL,
  var_ref = NULL,
  var_names = NULL,
  res_AAF = NULL,
  source = "",
  data = data.frame()
)
```

Arguments

model	The model used to perform statistical analysis
var_ref	The variable to explain in the statistical model
var_names	A list with the name of the variable you want to show on the contingency tables
res_AAF	the result of the AAF computation with `get_AAF()` function (done outside of this function because of the length of the computation)
source	Data source information to add to the plots
data	The data used for the statistical analysis

Value

A gt table

Examples

```
set.seed(9876)
data <- data.frame(
   "var_dep" = as.factor(sample(c(rep_len(c(0, 1), length.out = 1000)), size = 100)),
   "var_1" = as.factor(sample(c(rep_len(c(1, 2, 3), length.out = 1000)), size = 100)),
   "var_2" = as.factor(sample(c(rep_len(c(3, 1, 2), length.out = 100)), size = 100))
   )
glmmodel <- glm(var_dep ~ var_1 + var_2, data = data, family = binomial("logit"))


res <- get_AAF(model = glmmodel, nvar = 3, vars_dep = "var_dep", data = data,
               nbootstrap = 2)
res_tab <- result_tab(model = glmmodel, var_ref = "var_dep",
                      var_names = c("var_1", "var_2"),
                      res_AAF = res$res, source = "TESTDATA", data = data)
```

Index