Package 'nof1'

August 13, 2018

Type Package

Title Single Subject (N-Of-1) Designs to Answer Patient-Identified Research Questions

Version 0.5.0

Depends R (>= 2.10)

Imports rjags (>= 4-6), splines, combinat, MASS, jsonlite, ggplot2, scales, coda (>= 0.13)

Description A package for running N of 1 study trials. Runs Bayesian linear regression, ordinal/logistic regression, and poisson regression. Includes different plots to visualize the results.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

nof1-package . . . 2
find_raw_mean2 . . . 2
frequency_plot . . . 2
kernel_plot . . . 3
nof1.binomial.simulation . . . 3
nof1.data . . . 3
nof1.inits . . . 5
nof1.normal.simulation . . . 5
nof1.ordinal.simulation . . . 5
nof1.ordinal.simulation2 . . . 6
nof1.poisson.simulation . . . 6
nof1.run . . . 6
odds_ratio_plot . . . 7
probability_barplot . . . 7
raw_table . . . 7
read_input_data . . . 8
stacked_percent_barplot . . . 8
summarize_nof1 . . . 8
time_series_plot . . . 9
time_series_plot2 . . . 9
wrap . . . 9
wrap2 . . . 10

Index **11**

nof1-package *mcnet: A package for N of 1 study analysis using Bayesian methods*

Description

A package for running N of 1 study trials

Details

An N of 1 trial is a clinical trial in which a single patient is the entire trial, a single case study. The main purpose of this package was to serve as an analysis tool for one of the PCORI grants we were working with. It is designed for N of 1 trials and can fit bayesian versions of linear regression, logistic/ordinal regression, and poisson regression. Package includes number of different plotting tools for visualization.

find_raw_mean2 *Summarizes the result from the model into json format*

Description

Summarizes the result from the model into json format

Usage

```
find_raw_mean2(Y, Treat, baseline, response)
```

frequency_plot *Frequency plot for raw data*

Description

Frequency plot for raw data

Usage

```
frequency_plot(nof1, xlab = NULL, title = NULL)
```

Arguments

nof1	nof1 object created using nof1.data

kernel_plot *Kernel density of the posterior distribution for odds ratio*

Description

Kernel density of the posterior distribution for odds ratio

Usage

```
kernel_plot(result, xlim_value = c(0, 10), title = NULL)
```

Arguments

result	nof1 result object created using nof1.run

nof1.binomial.simulation *Binomial simulation*

Description

Binomial simulation

Usage

```
nof1.binomial.simulation(Base.size = 14, Treat.size = 56, alpha = 0.5,
  beta_A = -0.1, beta_B = -0.05)
```

nof1.data *Make a nof1 object containing data, priors, and a jags model file*

Description

Make a nof1 object containing data, priors, and a jags model file

Usage

```
nof1.data(Y, Treat, baseline = "baseline", ncat = NULL, response = NULL,
  Time = NULL, knots = NULL, alpha.prior = NULL, beta.prior = NULL,
  dc.prior = NULL, c1.prior = NULL, rho.prior = NULL, hy.prior = NULL)
```

Arguments

Y	Outcome of the study. This should be a vector with length of total number of observations.
Treat	Treatment indicator vector with same length as the outcome.
baseline	baseline Treatment name. This serves as a baseline/placebo when comparing different treatments.
ncat	Number of categories. Used in ordinal models.
response	Type of outcome. Can be normal, binomial, poisson or ordinal.
Time	parameter used for modelling splines. Still under development.
knots	parameter used for modelling splines. Still under development.
alpha.prior	Prior for the intercept of the model.
beta.prior	Prior for the treatment coefficient.
dc.prior	Prior for the length between cutpoints. Used only for ordinal logistic models.
c1.prior	Prior for the first cutpoint. Used only for ordinal logistic models.
rho.prior	Prior for the correlated error model. Still under development.
hy.prior	Prior for the heterogeneity parameter. Supports uniform, gamma, and half normal for normal and binomial response and wishart for multinomial response. It should be a list of length 3, where first element should be the distribution (one of dunif, dgamma, dhnorm, dwish) and the next two are the parameters associated with the distribution. For example, list("dunif", 0, 5) give uniform prior with lower bound 0 and upper bound 5 for the heterogeneity parameter. For wishart distribution, the last two parameter would be the scale matrix and the degrees of freedom.
gamma.prior	Prior for modelling splines. Still under development.

Value

Creates list of variables that are used to run the model using `nof1.run`

Y	Outcome
Treat	Treatment
baseline	Baseline variable
ncat	Number of categories for ordinal response
nobs	Total number of observations in a study
Treat.name	Treatment name besides baseline treatment
response	The type of response variable
priors	Priors that the code will be using
code	Rjags model file code that is generated using information provided by the user. To view model file inside R, use `cat(nof1$code)`.

Examples

```
###Blocker data example
laughter
Y <- laughter$Y
Treat <- laughter$Treat
nof1 <- nof1.data(Y, Treat, ncat = 11, baseline = "Usual Routine", response = "ordinal")
str(nof1)
cat(nof1$code)
```

nof1.inits *Generate initial values*

Description

Generate initial values

Usage

```
nof1.inits(nof1, n.chains)
```

Arguments

n.chains	number of chains you want

nof1.normal.simulation *Normal simulation*

Description

Normal simulation

Usage

```
nof1.normal.simulation(Base.size = 2, Treat.size = 8, prec = 0.5,
  alpha = 50, beta_A = -3, beta_B = -1)
```

nof1.ordinal.simulation *Ordinal simulation*

Description

Ordinal simulation

Usage

```
nof1.ordinal.simulation(Base.size = 100, Treat.size = 100, alpha = 0,
  beta_A = -0.1, beta_B = -0.3, cut = c(0.5, 1, 1.5, 2), ncat = 5)
```

nof1.ordinal.simulation2 *Ordinal simulation for WNYC*

Description

Ordinal simulation for WNYC

Usage

```
nof1.ordinal.simulation2(alpha = 0, beta_B = 1, cut = c(-2, -1.5, -1,
  -0.5, 0, 0.5, 1, 1.5, 2, 2.1, 2.3), ncat = 11)
```

nof1.poisson.simulation *Poisson simulation*

Description

Poisson simulation

Usage

```
nof1.poisson.simulation(Base.size = 14, Treat.size = 56, alpha = 1,
  beta_A = -0.1, beta_B = -0.05)
```

nof1.run *Run nof1 model*

Description

Run nof1 model

Usage

```
nof1.run(nof1, inits = NULL, n.chains = 3, max.run = 1e+05,
  setsize = 10000, n.run = 50000, conv.limit = 1.05,
  extra.pars.save = NULL)
```

Arguments

nof1	nof1 object created using nof1.data

odds_ratio_plot *Odds ratio plot for the raw data*

Description

Odds ratio plot for the raw data

Usage

```
odds_ratio_plot(result.list, result.name = NULL, level = 0.95,
  title = NULL)
```

Arguments

result.list	list of nof1 results created using nof1.run
level	confidence interval level (default is 0.95)
name	of the outcomes. If left unspecified, it numbers each result in order of how it is stored in result.list

probability_barplot *Plot showing probability certain treatment is better than the other one*

Description

Plot showing probability certain treatment is better than the other one

Usage

```
probability_barplot(result.list, result.name = NULL)
```

Arguments

result.list	list of nof1 results created using nof1.run
result.name	name of the outcomes. If left unspecified, it numbers each result in order of how it is stored in result.list

raw_table *Summary data table for nof1*

Description

Summary data table for nof1

Usage

```
raw_table(nof1)
```

Arguments

nof1	nof1 object created using nof1.data

read_input_data *Read json data in as an R object*

Description

Read json data in as an R object

Usage

```
read_input_data(data, metadata)
```

Arguments

data	input data. see sample input.json in the github repo
metadata	metadata. see sample input.json in the github repo

stacked_percent_barplot *Stacked_percent_barplot for raw data (for ordinal or binomial data)*

Description

Stacked_percent_barplot for raw data (for ordinal or binomial data)

Usage

```
stacked_percent_barplot(nof1, title = NULL)
```

Arguments

nof1	nof1 object created using nof1.data

summarize_nof1 *Summarizes the result from the model into json format*

Description

Summarizes the result from the model into json format

Usage

```
summarize_nof1(nof1, result)
```

time_series_plot *Time series plot for the raw data*

Description

Draw time series plot

Usage

```
time_series_plot(nof1, time = NULL, timestamp = NULL,
  timestamp.format = "%m/%d/%Y %H:%M")
```

Arguments

nof1	nof1 object created using nof1.data
time	can manually specify time variable
timestamp	or instead provide timestamp information for the all the outcomes
timestamp.format	format of timestamp used. See default format.

time_series_plot2 *time series plot across different interventions*

Description

time series plot across different interventions

Usage

```
time_series_plot2(nof1, time = NULL, timestamp = NULL,
  timestamp.format = "%m/%d/%Y %H:%M", Outcome.name = "")
```

Arguments

nof1	nof1 object created using nof1.data

wrap *Wrapper function that runs the n-of-1 model*

Description

Wrapper function that runs the n-of-1 model

Usage

```
wrap(data, metadata)
```

Arguments

json.file	input json data

wrap2 *Wrapper function for afib study that runs the n-of-1 model*

Description

Wrapper function for afib study that runs the n-of-1 model

Usage

```
wrap2(data, metadata)
```

Arguments

json.file	input json data

Index

find_raw_mean2, 2
frequency_plot, 2

kernel_plot, 3

nof1-package, 2
nof1.binomial.simulation, 3
nof1.data, 3
nof1.inits, 5
nof1.normal.simulation, 5
nof1.ordinal.simulation, 5
nof1.ordinal.simulation2, 6
nof1.poisson.simulation, 6
nof1.run, *4*, 6

odds_ratio_plot, 7

probability_barplot, 7

raw_table, 7
read_input_data, 8

stacked_percent_barplot, 8
summarize_nof1, 8

time_series_plot, 9
time_series_plot2, 9

wrap, 9
wrap2, 10