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 regressions, ordinal/logistic regression, and poisson regression. Includes visualization tools to evaluate the results.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

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.

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 network object containing data, priors, and a jags model file*

Description

Make a network 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,
  gamma.prior = NULL, dc.prior = NULL, c1.prior = NULL,
  rho.prior = NULL, hy.prior = NULL)
```

Arguments

`Y`	Outcome
`Treat`	Treatment indicator vector
`baseline`	baseline Treatment name
`ncat`	Number of categories. Used in ordinal models
`response`	Type of outcome. Can be normal, binomial, poisson or ordinal

`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