

# Package: MetaStan (via r-universe)

October 26, 2024

**Version** 1.0.0

**Title** Bayesian Meta-Analysis via 'Stan'

**Description** Performs Bayesian meta-analysis, meta-regression and model-based meta-analysis using 'Stan'. Includes binomial-normal hierarchical models and option to use weakly informative priors for the heterogeneity parameter and the treatment effect parameter which are described in Guenhan, Roever, and Friede (2020) <[doi:10.1002/jrsm.1370](https://doi.org/10.1002/jrsm.1370)>.

**License** GPL (>=3)

**Encoding** UTF-8

**Date** 2021-12-23

**LazyData** true

**ByteCompile** true

**Depends** R (>= 3.5.0)

**Imports** Rcpp (>= 0.12.17), methods, rstan (>= 2.18.1), rstantools (>= 1.5.0), loo (>= 2.0), forestplot (>= 1.6), metafor (>= 2.0-0), HDInterval, coda

**Suggests** testthat, knitr, rmarkdown, ggplot2, shinystan, vdiff, bayesmeta, metadat

**LinkingTo** StanHeaders (>= 2.18.0), rstan (>= 2.18.1), BH (>= 1.66.0-1), Rcpp (>= 0.12.17), RcppEigen (>= 0.3.3.4.0), RcppParallel (>= 5.0.1),

**SystemRequirements** GNU make

**NeedsCompilation** yes

**RoxygenNote** 7.2.3

**URL** <https://github.com/gunhanb/MetaStan>

**BugReports** <https://github.com/gunhanb/MetaStan/issues>

**VignetteBuilder** knitr

**Repository** <https://gunhanb.r-universe.dev>

**RemoteUrl** <https://github.com/gunhanb/metastan>

**RemoteRef** HEAD

**RemoteSha** 2fe3438fd1826207cf7d8cac80db7a0cbe631406

## Contents

MetaStan-package . . . . .	2
compare_MBMA . . . . .	3
convert_data_arm . . . . .	4
create_MetaStan_dat . . . . .	4
dat.Berkey1995 . . . . .	6
dat.Boucher2016 . . . . .	6
dat.Boucher2016.pairwise . . . . .	7
dat.Crins2014 . . . . .	8
dat.Eletriptan . . . . .	9
forest_plot . . . . .	10
MBMA_stan . . . . .	11
metastan_data . . . . .	13
meta_stan . . . . .	15
plot_MBMA_stan . . . . .	18
print_MBMA_stan . . . . .	19
print_meta_stan . . . . .	20
<b>Index</b>	<b>21</b>

---

MetaStan-package	<i>The 'MetaStan' package.</i>
------------------	--------------------------------

---

## Description

Fitting Bayesian meta-analysis models via Rstan.

## Details

To fit meta-analysis models using frequentist methods, there are many R packages available including 'metafor'. On the other hand, Bayesian estimation methods such as Markov chain Monte Carlo (MCMC) are very attractive for meta-analysis, especially because they can be used to fit more complicated models. These include binomial-normal hierarchical models and beta-binomial models which are based on the exact distributional assumptions unlike (commonly used) normal-normal hierarchical model. Another advantage of Bayesian methods to be able to use informative prior distributions for example to regularize heterogeneity estimates in case of low number of studies. Thus, we developed 'MetaStan' which uses Stan (a modern MCMC engine) to fit several pairwise meta-analysis models including binomial-normal hierarchical model and beta-binomial model. This package is also the accompanying package of Guenhan et al (2020). Another important functionality of the package is the model-based meta-analysis models.

## Author(s)

Burak Kuersad Guenhan <burak.gunhan@med.uni-goettingen.de>

## References

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.17.3. <http://mc-stan.org>

Günhan, B and Röver, C and Friede, T (2020). Random-effects meta-analysis of few studies involving rare events. Research Synthesis Methods. doi = 10.1002/jrsm.1370.

---

compare\_MBMA

*Compare MBMA fits using LOO-IC*

---

## Description

Takes a vector of MBMA\_stan fits and give the model comparison results based on LOO-IC criteria. This is useful to compare different dose-response models. The function depends on loo\_compare function from loo package.

## Usage

```
compare_MBMA(model_list, digits = 2, ...)
```

## Arguments

model_list	A vector of MBMA_stan object.
digits	An integer indicating the number of decimal places.
...	Further arguments passed to or from other methods.

## References

Vehtari, A, A Gelman, and J Gabry (Sept. 2017). "Practical Bayesian model evaluation using leave one-out cross-validation and WAIC." In: Statistics and Computing 27.5, pp. 1413–1432.

## See Also

loo::loo\_compare

---

convert_data_arm	<i>Convert contrast-based dataset to arm-based dataset (deprecated)</i>
------------------	---

---

### Description

convert\_data\_arm creates a dataframe to fit a meta-analysis model using meta\_Stan function.

### Usage

```
convert_data_arm(nt, nc, pt, pc, pub, data = NULL)
```

### Arguments

nt	Number of subjects in treatment arm
nc	Number of subjects in control arm
pt	Number of events in treatment arm
pc	Number of events in treatment arm
pub	The corresponding publication
data	Optional data frame containing the variables given to the arguments above.

### Value

A dataframe object

### Examples

```
## Create arm-based dataset
data('dat.Crins2014', package = "MetaStan")
dat_converted <- convert_data_arm(exp.total, cont.total,
                                  exp.AR.events, cont.AR.events,
                                  publication, data = dat.Crins2014)
```

---

create_MetaStan_dat	<i>Prepare model-based meta-analysis dataset for Stan.</i>
---------------------	--

---

### Description

create\_MetaStan\_dat converts datasets in the one-study-per-row format to one-arm-per-row format,

**Usage**

```
create_MetaStan_dat(  
  dat = NULL,  
  armVars = c(dose = "d", responders = "r", sampleSize = "n"),  
  nArmsVar = "nd"  
)
```

**Arguments**

dat	Data in one-study-per-row format.
armVars	Vector of per-arm variables The name of each component will be the column name in the resulting dataset.
nArmsVar	Variable holding the number of arms for each study.

**Details**

The resulting data.frame can be used as data argument in MBMA\_stan.

**Value**

A data frame with the generated columns.

**Author(s)**

Burak Kuersad Guenhan, <burak.gunhan@med.uni-goettingen.de> and Gert van Valkenhoef

**See Also**

gemtc::mtc.data.studyrow and nmaINLA::create\_INLA\_dat

**Examples**

```
## Not run:  
data('dat.Eletriptan')  
## Create the dataset suitable for MBMA_stan  
EletriptanDat <- create_MetaStan_dat(dat = dat.Eletriptan,  
                                     armVars = c(dose = "d",  
                                                  responders = "r",  
                                                  sampleSize = "n"),  
                                     nArmsVar = 'nd')  
  
## Check that the data are correct  
print(EletriptanDat)  
  
## End(Not run)
```

---

 dat.Berkey1995

*Trials investigating effectiveness of the BCG vaccine against TB*


---

**Description**

A dataset containing the results from 13 trials examining the efficacy of Bacillus Calmette-Guerin (BCG) vaccine against tuberculosis (TB).

**Usage**

dat.Berkey1995

**Format**

A data frame with following columns

**Trial** Trial number

**r1** number of TB events in treatment arm

**n1** number of subjects in treatment arm

**r2** number of TB events in control arm

**n2** number of subjects in control arm

**Latitude** absolute latitude of the study location

**publication** author and date

**Source**

Berkey, C.S., Hoaglin, D.C., Mosteller, F. and Colditz, G.A., 1995. A random-effects regression model for meta-analysis. *Statistics in medicine*, 14(4), pp.395-411

---

 dat.Boucher2016

*PARESTHESIA RATES WITH TOPIRAMATE IN MIGRAINE PRO-PHYLAXIS TRIALS*


---

**Description**

Numbers of patients and events (paresthesia rates) in experimental and control groups of 6 studies. It is in one-study-per-row format.

**Usage**

dat.Boucher2016

**Format**

A data frame with following columns

- d1** dose (mg) in the first arm (placebo)
- r1** number of events in the first arm (placebo)
- n1** number of patients in the first arm (placebo)
- d2** dose (mg) in the second arm
- r2** number of events in the second arm
- n2** number of patients in the second arm
- d3** dose (mg) in the third arm
- r3** number of events in the third arm
- n3** number of patients in the third arm
- d4** dose (mg) in the fourth arm
- r4** number of events in the fourth arm
- n4** number of patients in the fourth arm
- nd** number of treatment arms

**Source**

Boucher M, and Bennets M. The Many Flavors of Model-Based Meta-Analysis: Part I - Introduction and Landmark Data. *CPT Pharmacometrics Syst. Pharmacol.* (2016) 5, 54-64; doi:10.1002/psp4.12041

---

dat.Boucher2016.pairwise

*PARESTHESIA RATES WITH TOPIRAMATE IN MIGRAINE PRO-PHYLAXIS TRIALS*

---

**Description**

Numbers of patients and events (paresthesia rates) in experimental and control groups of 6 studies. It is in one-study-per-row format. Only the arm with 200 mg is included.

**Usage**

dat.Boucher2016.pairwise

**Format**

A data frame with following columns

- study** Study ID
- duration** Duration of the study
- r1** number of events in the first arm (placebo)
- n1** number of patients in the first arm (placebo)
- r2** number of events in the second arm
- n2** number of patients in the second arm

**Source**

Boucher M, and Bennets M. The Many Flavors of Model-Based Meta-Analysis: Part I - Introduction and Landmark Data. *CPT Pharmacometrics Syst. Pharmacol.* (2016) 5, 54-64; doi:10.1002/psp4.12041

---

dat.Crins2014

*Pediatric liver transplant example data*

---

**Description**

Numbers of cases and events (PTLDs or deaths) in experimental and control groups of six studies.

**Usage**

dat.Crins2014

**Format**

A data frame with following columns

**publication** publication identifier (first author and publication year)

**year** publication year

**randomized** randomization status (y/n)

**control.type** type of control group ("concurrent" or "historical")

**comparison** type of comparison ("IL-2RA only", "delayed CNI", or "no/low steroids")

**followup** t follow-up time in months

**exp.AR.events** number of AR events in experimental group

**exp.PTLD.events** number of PTLD events in experimental group

**exp.deaths** number of deaths in experimental group

**exp.total** number of patients in experimental group

**exp.SRR.events** number of SRR events in experimental group

**cont.AR.events** number of AR events in control group

**cont.SRR.events** number of SRR events in control group

**cont.PTLD.events** number of PTLD events in control group

**cont.deaths** number of deaths in control group

**cont.total** number of patients in control group

**r2** number of AR events in experimental group

**n1** number of patients in control group

**n2** number of patients in experimental group

**r1** number of AR events in control group

**Source**

N.D. Crins, C. Roever, A.D. Goralczyk, T. Friede. Interleukin-2 receptor antagonists for pediatric liver transplant recipients: A systematic review and meta-analysis of controlled studies. *Pediatric Transplantation*, 18(8):839-850, 2014.



---

dat.Eletriptan	<i>Migraine pain relief example (Eletriptan)</i>
----------------	--

---

**Description**

Numbers of patients and events (headache free at 2 hours) in experimental and control groups of 12 studies. It is in one-study-per-row format.

**Usage**

```
dat.Eletriptan
```

**Format**

A data frame with following columns

**ID** trial ID  
**d1** dose (mg) in the first arm (placebo)  
**r1** number of events in the first arm (placebo)  
**n1** number of patients in the first arm (placebo)  
**d2** dose (mg) in the second arm  
**r2** number of events in the second arm  
**n2** number of patients in the second arm  
**d3** dose (mg) in the third arm  
**r3** number of events in the third arm  
**n3** number of patients in the third arm  
**d4** dose (mg) in the fourth arm  
**r4** number of events in the fourth arm  
**n4** number of patients in the fourth arm  
**nd** number of treatment arms

**Source**

Thorlund, K., Mills, E., Wu, P., Ramos, E., Chatterjee, A., Druyts, E., and Goadsby, P. (2014). Comparative efficacy of triptans for the abortive treatment of migraine: A multiple treatment comparison meta-analysis. *Cephalalgia*, 34(4):258-267.

---

forest_plot	<i>Plot a forest plot</i>
-------------	---------------------------

---

**Description**

Takes a meta\_stan object which is obtained by function meta\_stan and plot a forestplot, showing individual estimates along with their 95 percent credible intervals, resulting effect estimate and prediction interval.

**Usage**

```
forest_plot(  
  x = NULL,  
  labels = NULL,  
  digits = 2,  
  boxsize = 0.3,  
  heterogeneity = TRUE,  
  col,  
  ...  
)
```

**Arguments**

x	A meta_stan object.
labels	Optional vector with labels for the studies (publication author/year).
digits	A numerical value specifying the number of significant digits to be shown. Default is 2.
boxsize	A numerical value specifying the box size. Default is 0.3.
heterogeneity	A logical value to include heterogeneity estimates (DEFAULT = TRUE)
col	A function specifying the colors. See forestplot::fpColors for details.
...	Further arguments passed to or from other methods.

**Value**

The return value is invisible NULL.

**Author(s)**

Christian Roever and Burak Kuersad Guenhan

**Source**

This function is based foresplot function from foresplot R package.

**See Also**

foresplot::foresplot

**Examples**

```
## Not run:
data('dat.Crins2014', package = "MetaStan")
dat_long <- create_MetaStan_dat(dat = dat.Crins2014,
                               armVars = c(responders = "r", sampleSize = "n"))
bnhm.Crins <- meta_stan(data = dat_long, likelihood = "binomial",
                      mu_prior = c(0, 10), theta_prior = c(0, 100),
                      tau_prior = 0.5)
forest_plot(bnhm.Crins, xlab = "log-OR", labels = dat.Crins2014$publication)

## End(Not run)
```

---

 MBMA\_stan

*Fitting a model-based meta-analysis model using Stan*


---

**Description**

‘MBMA\_stan’ fits a model-based meta-analysis model using Stan.

**Usage**

```
MBMA_stan(
  data = NULL,
  likelihood = NULL,
  dose_response = "emax",
  mu_prior = c(0, 10),
  Emax_prior = c(0, 100),
  alpha_prior = c(0, 100),
  tau_prior = 0.5,
  tau_prior_dist = "half-normal",
  ED50_prior = c(-2.5, 1.8),
  ED50_prior_dist = "functional",
  gamma_prior = c(1, 2),
  Pred_doses,
  re = TRUE,
  ncp = TRUE,
  chains = 4,
  iter = 2000,
  warmup = 1000,
  adapt_delta = 0.95,
  ...
)
```

**Arguments**

`data` An object of ‘create\_MBMA\_dat’.

likelihood	A string specifying the likelihood of distributions defining the statistical model. Options include "normal", "binomial", and "Poisson".
dose_response	A string specifying the function defining the dose-response model. Options include "linear", "log-linear", "emax", and "sigmoidal".
mu_prior	A numerical vector specifying the parameter of the normal prior density for baseline risks, first value is parameter for mean, second is for variance. Default is c(0, 10).
Emax_prior	A numerical vector specifying the parameter of the normal prior density for Emax parameter, first value is parameter for mean, second is for standard deviation. Default is c(0, 10). Needed for emax and sigmoidal models.
alpha_prior	A numerical vector specifying the parameter of the normal prior density for the alpha parameter, first value is parameter for mean, second is for variance. Default is c(0, 10). Needed for linear and linear log-dose models.
tau_prior	A numerical value specifying the standard dev. of the prior density for heterogeneity stdev. Default is 0.5.
tau_prior_dist	A string specifying the prior density for the heterogeneity standard deviation, option is 'half-normal' for half-normal prior, 'uniform' for uniform prior, 'half-cauchy' for half-cauchy prior.
ED50_prior	A numerical vector specifying the parameter of the normal prior density for ED50 parameter, first value is parameter for mean, second is for standard deviation. Default is c(0, 10). Needed for emax and sigmoidal models.
ED50_prior_dist	A string specifying the prior density for the ED50 parameter, 'functional' is for a functional uniform prior, 'half-normal' for uniform prior, 'half-cauchy' for half-cauchy prior.
gamma_prior	A numerical vector specifying the parameter of the normal prior density for gamma parameter, first value is parameter for mean, second is for standard deviation. Default is c(1, 2). Needed for sigmoidal model.
Pred_doses	A numerical vector specifying the doses which prediction will be made.
re	A string specifying whether random-effects are included to the model. When 'FALSE', the model corresponds to a fixed-effects model. The default is 'TRUE'.
ncp	A string specifying whether to use a non-centered parametrization. The default is 'TRUE'.
chains	A positive integer specifying the number of Markov chains. The default is 4.
iter	A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.
warmup	A positive integer specifying the number of warmup (aka burnin) iterations per chain. The default is 1000.
adapt_delta	A numerical value specifying the target average proposal acceptance probability for adaptation. See Stan manual for details. Default is 0.95. In general you should not need to change adapt_delta unless you see a warning message about divergent transitions, in which case you can increase adapt_delta from the default to a value closer to 1 (e.g. from 0.95 to 0.99, or from 0.99 to 0.999, etc).
...	Further arguments passed to or from other methods.

**Value**

an object of class 'stanfit' returned by 'rstan::sampling'

**References**

Boucher M, et al. The many flavors of model-based meta-analysis: Part I-Introduction and landmark data. *CPT: Pharmacometrics and Systems Pharmacology*. 2016;5:54-64.

Guenhan BK, Roeber C, Friede T. MetaStan: An R package for meta-analysis and model-based meta-analysis using Stan. In preparation.

Mawdsley D, et al. Model-based network meta-analysis: A framework for evidence synthesis of clinical trial data. *CPT: Pharmacometrics and Systems Pharmacology*. 2016;5:393-401.

Zhang J, et al. (2014). Network meta-analysis of randomized clinical trials: Reporting the proper summaries. *Clinical Trials*. 11(2), 246–262.

Dias S, et al. Absolute or relative effects? Arm-based synthesis of trial data. *Research Synthesis Methods*. 2016;7:23–28.

**Examples**

```
## Not run:
## Load the dataset
data('dat.Eletriptan', package = "MetaStan")
datMBMA = create_MetaStan_dat(dat = dat.Eletriptan,
                             armVars = c(dose = "d",
                                           responders = "r",
                                           sampleSize = "n"),
                             nArmsVar = "nd")

MBMA.Emax <- MBMA_stan(data = datMBMA,
                      likelihood = "binomial",
                      dose_response = "emax",
                      Pred_doses = seq(0, 80, length.out = 11),
                      mu_prior = c(0, 100),
                      Emax_prior = c(0, 100),
                      tau_prior_dist = "half-normal",
                      tau_prior = 0.5)
plot(MBMA.Emax) + ggplot2::xlab("Doses (mg)") + ggplot2::ylab("response probabilities")

## End(Not run)
```

---

metastan\_data

*Prepare meta-analysis dataset for meta\_stan function*


---

**Description**

metastan\_data creates datasets suitable for meta\_stan function

**Usage**

```
metastan_data(
  id,
  treatment,
  x,
  type = c("normal", "binomial", "poisson"),
  y,
  se,
  v,
  count,
  total,
  exposure,
  labels,
  data,
  sort = TRUE,
  checkForConflicts = TRUE
)
```

**Arguments**

id	a vector of study IDs (labels)
treatment	(optional) vector indicating treatment groups
x	(optional) covariable/regressor vector or matrix
type	type of outcome
y	estimates (for normal outcomes)
se	associated standard errors (normal outcomes)
v	variances / squared standard errors (normal outcomes)
count	event count (for binomial or Poisson outcomes)
total	sample size (for binomial outcomes)
exposure	exposure (for Poisson outcomes)
labels	(optional) vector of row labels
data	a data frame from which (some of) above variables may be taken
sort	if TRUE (the default), output rows will be sorted (keeping entries corresponding to the same study ("id") next to each other, and sorting by "treatment" and "x")

**Details**

The resulting data.frame can be used as data argument in meta\_stan.

NB: arguments "id", "treatment", "x", "y", "se", "v", "count", "total", "exposure", "labels" may be taken from local environment OR from "data" (data frame) argument.

**Value**

The returned object is a "list" of class "metastan\_data" containing the following elements: \* "id" : study identifier (a vector of type factor) \* "treatment" : (optional) a treatment identifier (a vector of type factor, or NULL) \* "x" : (optional) covariable(s), (a numeric matrix, or NULL) \* "type" : a flag identifying the outcome type ("normal", "binomial", or "poisson") \* "outcome" : the actual outcome data (a two-column numeric matrix) \* "call" : the original function call

**Examples**

```
# 3 studies, binomial endpoint, one covariable (named "dose"):
msd <- metastan_data(id=c("Smith", "Smith", "Taylor", "Taylor",
                          "Jones", "Jones", "Taylor"),
                    treatment=c("placebo", "verum", "placebo", "verum",
                                "verum", "placebo", "verum"),
                    type="binomial",
                    count=c(3, 2, 7, 5, 4, 5, 6),
                    total=c(10, 10, 25, 27, 15, 16, 23),
                    x=cbind("dose"=c(0, 50, 0, 50, 30, 0, 20)))

msd
print(msd, n=7)      # (show all 7 lines of data)
print.default(msd)  # default view of returned object
```

---

 meta\_stan

*Fitting a meta-analysis model using Stan*


---

**Description**

'meta\_stan' fits a meta-analysis model using Stan.

**Usage**

```
meta_stan(
  data = NULL,
  likelihood = NULL,
  mu_prior = c(0, 10),
  theta_prior = NULL,
  tau_prior = 0.5,
  tau_prior_dist = "half-normal",
  beta_prior = c(0, 100),
  delta = NULL,
  param = "Smith",
  re = TRUE,
  ncp = TRUE,
  interval.type = "shortest",
  mreg = FALSE,
  cov = NULL,
  chains = 4,
  iter = 2000,
```

```

    warmup = 1000,
    adapt_delta = 0.95,
    ...
)

```

## Arguments

data	Data frame created by ‘create_MetaStan_dat’
likelihood	A string specifying the likelihood function defining the statistical model. Options include ‘normal’, ‘binomial’, and ‘Poisson’.
mu_prior	A numerical vector specifying the parameter of the normal prior density for baseline risks, first value is parameter for mean, second is for variance. Default is c(0, 10).
theta_prior	A numerical vector specifying the parameter of the normal prior density for treatment effect estimate, first value is parameter for mean, second is for variance. Default is NULL.
tau_prior	A numerical value specifying the standard dev. of the prior density for heterogeneity stdev. Default is 0.5.
tau_prior_dist	A string specifying the prior density for the heterogeneity standard deviation, option is ‘half-normal’ for half-normal prior, ‘uniform’ for uniform prior, ‘half-cauchy’ for half-cauchy prior.
beta_prior	A numerical vector specifying the parameter of the normal prior density for beta coefficients in a meta-regression model, first value is parameter for mean, second is for variance. Default is c(0, 100).
delta	A numerical value specifying the upper bound of the a priori interval for treatment effect on odds ratio scale ( <i>Guenhan et al (2020)</i> ). This is used to calculate a normal weakly informative prior. for theta. Thus when this argument is specified, ‘theta’ should be left empty. Default is NULL.
param	Parameterization used. The default is the ‘Smith’ model suggested by Smith et al (1995). The alternative is ‘Higgins’ is the common meta-analysis model (Simmonds and Higgins, 2014).
re	A string specifying whether random-effects are included to the model. When ‘FALSE’, the model corresponds to a fixed-effects model. The default is ‘TRUE’.
ncp	A string specifying whether to use a non-centered parametrization. The default is ‘TRUE’.
interval.type	A string specifying the type of interval estimate. Options include shortest credible interval ‘shortest’ (default) and qui-tailed credible interval ‘central’.
mreg	A string specifying whether to fit a meta-regression model. The default is ‘FALSE’.
cov	A numeric vector or matrix specifying trial-level covariates (in each row). This is needed when ‘mreg = TRUE’.
chains	A positive integer specifying the number of Markov chains. The default is 4.
iter	A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.





```
mreg = TRUE,  
cov = dat.Berkey1995$Latitude)  
  
print(meta.reg.stan)  
  
## End(Not run)
```

---

plot.MBMA_stan	<i>Plot a dose-response plot</i>
----------------	----------------------------------

---

### Description

Takes a MBMA\_stan object which is obtained by function MBMA\_stan and plot a dose-response plot, showing observed event probabilities and the estimated dose-response function with pointwise 95

### Usage

```
## S3 method for class 'MBMA_stan'  
plot(x = MBMA.stan, ...)
```

### Arguments

x	A MBMA_stan object.
...	Further arguments passed to ggplot.

### Value

The return value is invisible NULL.

### Author(s)

Christian Roever and Burak Kuersad Guenhan

### Source

This function uses ggplot function from ggplot2 R package.

### See Also

ggplot2::ggplot

**Examples**

```
## Not run:
data('dat.Eletriptan', package = "MetaStan")
datMBMA = create_MetaStan_dat(dat = dat.Eletriptan,
                              armVars = c(dose = "d",
                                           responders = "r",
                                           sampleSize = "n"),
                              nArmsVar = "nd")

MBMA.Emax <- MBMA_stan(data = datMBMA,
                      likelihood = "binomial",
                      dose_response = "emax",
                      Pred_doses = seq(0, 80, length.out = 11),
                      mu_prior = c(0, 100),
                      Emax_prior = c(0, 100),
                      tau_prior_dist = "half-normal",
                      tau_prior = 0.5)
plot(MBMA.Emax) + ggplot2::xlab("Doses (mg)") + ggplot2::ylab("response probabilities")

## End(Not run)
```

---

```
print.MBMA_stan      Print MBMA object
```

---

**Description**

Takes an MBMA\_stan object which is obtained by function MBMA\_stan and print the model and data information such as model type used in the model.

**Usage**

```
## S3 method for class 'MBMA_stan'
print(x, digits = 2, ...)
```

**Arguments**

x	A MBMA_stan object.
digits	An integer indicating the number of decimal places.
...	Further arguments passed to or from other methods.

**Value**

The return value is invisible NULL

---

<code>print.meta_stan</code>	<i>Print meta_stan object</i>
------------------------------	-------------------------------

---

**Description**

Takes an `meta_stan` object which is obtained by function `meta_stan` and print the model and data information such as model type used in the model.

**Usage**

```
## S3 method for class 'meta_stan'  
print(x, digits = 2, ...)
```

**Arguments**

<code>x</code>	A <code>meta_stan</code> object.
<code>digits</code>	An integer indicating the number of decimal places.
<code>...</code>	Further arguments passed to or from other methods.

**Value**

The return value is invisible NULL

# Index

## \* datasets

- dat.Berkey1995, [6](#)
- dat.Boucher2016, [6](#)
- dat.Boucher2016.pairwise, [7](#)
- dat.Crins2014, [8](#)
- dat.Eletriptan, [9](#)

- compare\_MBMA, [3](#)
- convert\_data\_arm, [4](#)
- create\_MetaStan\_dat, [4](#)

- dat.Berkey1995, [6](#)
- dat.Boucher2016, [6](#)
- dat.Boucher2016.pairwise, [7](#)
- dat.Crins2014, [8](#)
- dat.Eletriptan, [9](#)

- forest\_plot, [10](#)

- MBMA\_stan, [11](#)
- meta\_stan, [15](#)
- MetaStan (MetaStan-package), [2](#)
- MetaStan-package, [2](#)
- metastan\_data, [13](#)

- plot\_MBMA\_stan, [18](#)
- print\_MBMA\_stan, [19](#)
- print\_meta\_stan, [20](#)