

# Package: RegDDM (via r-universe)

December 10, 2024

**Title** Generalized Linear Regression with DDM

**Version** 1.1

**Description** Drift-Diffusion Model (DDM) has been widely used to model binary decision-making tasks, and many research studies the relationship between DDM parameters and other characteristics of the subject. This package uses 'RStan' to perform generalized linear regression analysis over DDM parameters via a single Bayesian Hierarchical model. Compared to estimating DDM parameters followed by a separate regression model, 'RegDDM' reduces bias and improves statistical power.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**URL** <https://github.com/biorabbit/RegDDM>

**BugReports** <https://github.com/biorabbit/RegDDM/issues>

**Imports** rstan, stringr, dplyr, tidyr, purrr, rtdists, rlang, stats,

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**LazyData** true

**Config/pak/sysreqs** make libgs10-dev libicu-dev

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

**RemoteUrl** <https://github.com/biorabbit/regddm>

**RemoteRef** HEAD

**RemoteSha** 477a1fd61476d0b3c3b54bc1807b8ac7fa6465c2

## Contents

generate_sim_data	2
get_stan_fit	3
print.regddmfit	4
regddm	4
regddmfit	6
regddm_tutorial	7
summary.regddmfit	7

<b>Index</b>	<b>9</b>
--------------	----------

---

generate_sim_data	<i>Generate simulated binary decision data using DDM</i>
-------------------	--

---

## Description

This function generates a simulated dataset under different configurations. It can be used to test the performance and functionality of **RegDDM**. The outcome variable is  $y$ , which is influenced by different variables.

## Usage

```
generate_sim_data(
  N = 30,
  n_each = 100,
  n_xvar = 2,
  beta_0 = 0,
  beta_c1 = 0,
  beta_c2 = 0,
  beta_v_0 = 0,
  beta_v_x1 = 0,
  beta_v_x2 = 0,
  sigma_y = 1,
  sigma_v = 0,
  y_family = "gaussian"
)
```

## Arguments

N	Number of subjects.
n_each	Number of trials per subject
n_xvar	Number of trial-level variables influencing drift rate
beta_0	Intercept
beta_c1	Slope of c1
beta_c2	Slope of c2

beta_v_0	Slope of v_0
beta_v_x1	Slope of v_x1
beta_v_x2	Slope of v_x2
sigma_y	Standard deviation of error term of y, Only used when y_family is "gaussian"
sigma_v	Contaminant level for drift rate v.
y_family	Family of distribution of y. Can be either "gaussian", "bernoulli" or "poisson"

**Value**

A named list with four elements. data1\_true and data2\_true are true values of DDM parameters of each subject and trial. data1 and data2 removed those hidden variables.

**Examples**

```
sim_data = generate_sim_data()
sim_data$data1
sim_data$data2
```

---

get\_stan\_fit

*Get the rstan fit of regddmfit objects*


---

**Description**

Get the [stanfit](#) object of the [regddmfit](#) object to perform further analysis and diagnosis.

**Usage**

```
get_stan_fit(fit)
```

**Arguments**

fit                    A regddmfit object to summary

**Value**

A stanfit object.

---

`print.regddmfit`      *Printed summary of **RegDDM** fit object*

---

### Description

Summarize the posterior distributions of estimated regression coefficients and print necessary information.

### Usage

```
## S3 method for class 'regddmfit'
print(x, digits = 3, ...)
```

### Arguments

<code>x</code>	An <code>regddmfit</code> object to print
<code>digits</code>	digits of the output results. Default value is 3.
<code>...</code>	Unused...

### Value

No values are returned.

### See Also

[summary.regddmfit](#) Table summaries of the `regddmfit` object.

---

`regddm`      *Bayesian hierachical generalized linear regression using Drift-Diffusion Model*

---

### Description

`regddm` makes it easy to fit a single Bayesian hierarchical drift-diffusion model (DDM) that estimates the DDM parameters of each subject and uses the estimated parameters as variables in a generalized linear regression.

**Usage**

```
regddm(
  data1,
  data2,
  model = list(),
  family = "gaussian",
  init = "default",
  prior = TRUE,
  stan_filename = "",
  gen_model = TRUE,
  fit_model = TRUE,
  warmup = 500,
  iter = 1000,
  chains = 4,
  cores = 4,
  ...
)
```

**Arguments**

<code>data1</code>	Subject-level dataframe with column such as age and gender. It must contain an <code>id</code> column unique for each subject.
<code>data2</code>	Trial-level dataframe. It must contain three columns: <code>id</code> , <code>response</code> and <code>rt</code> . It can also contain additional trial-level variables such as experiment condition.
<code>model</code>	A list containing 0-5 formulas, specifying the dependence structure between variables.
<code>family</code>	Family of distribution of <code>y</code> . Can be <code>gaussian</code> , <code>bernoulli</code> or <code>poisson</code> .
<code>init</code>	Either <code>default</code> or other values supported by <code>stan</code> function of <b>RStan</b>
<code>prior</code>	A logistic value, specifying whether or not to use default prior for DDM parameters. By default, <code>prior = TRUE</code> .
<code>stan_filename</code>	A string specifying the automatically generated stan file name. By default, an empty string <code>''</code> is provided. A temporary file will be created and deleted after the model is fit.
<code>gen_model</code>	A logistic value indicating weather or not to generate the model. If not, RegDDM will not generate the code but use the existing stan code from <code>stan_filename</code> instead.
<code>fit_model</code>	A logistic value indicating weather or not to fit the model. If not, RegDDM will only generate the code and return an unfitted <code>regddmfit</code> object.
<code>warmup</code>	Number of warm-up iterations. Default is 500.
<code>iter</code>	Number of iterations, which must be greater than <code>warmup</code> . Default value is 1000.
<code>chains</code>	Number of chains to run for diagnosis. Default value is 4.
<code>cores</code>	Number of cores to run the chains. It is best to make <code>cores = chains</code> . Default value is 4.
<code>...</code>	Other parameters sent to <code>stan</code> function of <b>RStan</b> .

**Value**

A `regddmfit` object.

**References**

To be added

**Examples**

```
# Note: each example takes about 20 minutes to run. During this period, you
# may not be able to open/save files or see the progress. To prevent this,
# it is recommended to copy, paste and run the example code in the console.
```

```
## Not run:
# Example analysis over the simulated tutorial dataset.
data(regddm_tutorial)
model = list(v ~ x1, y ~ v_0 + v_x1 + c1)
fit1 = regddm(
  regddm_tutorial$data1,
  regddm_tutorial$data2,
  model,
  stan_filename = ""
)
print(fit1)

# Alternatively, subjects' DDM parameters can be used as the outcome.
model = list(v ~ x1, v_x1 ~ y + c1)
fit2 = regddm(
  regddm_tutorial$data1,
  regddm_tutorial$data2,
  model,
  stan_filename = ""
)
print(fit2)

## End(Not run)
```

---

regddmfit

*Class of models fitted by **RegDDM***

---

**Description**

`regddmfit` is an S3 object storing the fitted models of **RegDDM**. It contains information used to fit the model and the resulting `stanfit` and can be summarized and printed using `summary` and `print`.

**Usage**

```
regddmfit(data1, data2, model, family, stan_fit)
```

**Arguments**

data1	Subject-level data frame.
data2	Trial-level data frame.
model	A list containing 0-5 formulas, specifying the dependence structure between variables.
family	Family of distribution of the outcome.
stan_fit	Fitted stan model.

**Details**

Use `methods(class = "regddmfit")` for a list of available methods.

---

regddm_tutorial	<i>Simulated example dataset</i>
-----------------	----------------------------------

---

**Description**

This is the same simulated dataset used in the tutorial part of the RegDDM paper.

$$v = x_1 \times v_{x_1}$$

$$y \sim N(1 + 1 \times v_{x_1} + 1 \times c_1, 1)$$

**Usage**

```
regddm_tutorial
```

**Format**

An object of class `list` of length 2.

---

summary.regddmfit	<i>Summary of <b>RegDDM</b> fit object</i>
-------------------	--

---

**Description**

Summarize the posterior distributions of estimated parameters and group them into four categories.

**Usage**

```
## S3 method for class 'regddmfit'
summary(object, ...)
```

**Arguments**

object            A [regddmfit](#) object to summary  
...                parameters passed to [summary,stanfit-method](#)

**Value**

The summary method returns a named list of `glm_coefficients`, `subject_ddm_param`, `group_param`, and `missing_value`. Each element is a [tibble](#) data frame of posterior summary statistics of the regression coefficient, DDM parameter of each subject, group mean and standard deviation of DDM parameters and covariates, plus the estimated missing values.

**See Also**

[print.regddmfit](#) a printed summary of the `regddmfit` object.



# Index

## \* datasets

- regddm\_tutorial, 7
  
- generate\_sim\_data, 2
- get\_stan\_fit, 3
  
- print, 6
- print.regddmfit, 4, 8
  
- regddm, 4
- regddm\_tutorial, 7
- regddmfit, 3–6, 6, 8
  
- stan, 5
- stanfit, 3, 6
- summary, 6
- summary.regddmfit, 4, 7
  
- tibble, 8