Package 'fddm'

September 9, 2022

Type Package

Title Fast Implementation of the Diffusion Decision Model

Version 0.5-2

Description Provides the probability density function (PDF), cumulative distribution function (CDF), and the partial derivatives of the PDF of the diffusion decision model (DDM; e.g., Ratcliff & McKoon, 2008, <doi:10.1162/neco.2008.12-06-420>) with across-trial variability in the drift rate. Because the PDF, its partial derivatives, and the CDF of the DDM both contain an infinite sum, they need to be approximated. 'fddm' implements all published approximations (Navarro & Fuss, 2009, <doi:10.1016/j.jmp.2009.02.003>; Gondan, Blurton, & Kesselmeier, 2014, <doi:10.1016/j.jmp.2014.05.002>; Blurton, Kesselmeier, & Gondan, 2017, <doi:10.1016/j.jmp.2016.11.003>; Hartmann & Klauer, 2021, <doi:10.1016/j.jmp.2021.102550>) plus new approximations. All approximations are implemented purely in 'C++' providing faster speed than existing packages.

License GPL (>= 2)

Imports Rcpp (>= 1.0.1)

LinkingTo Rcpp

Depends R (>= 3.5.0)

Suggests rtdists, RWiener, ggplot2, reshape2, testthat, knitr, rmarkdown, microbenchmark, ggnewscale, ggforce, WienR

URL `https://github.com/rtdists/fddm`

BugReports `https://github.com/rtdists/fddm/issues`

SystemRequirements C++11

RoxygenNote 7.2.0

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation yes

Author Kendal B. Foster [aut],
Henrik Singmann [ctb, cre] (<`https://orcid.org/0000-0002-4842-3657`>)

Maintainer Henrik Singmann <singmann@gmail.com>

Repository CRAN

Date/Publication 2022-09-09 19:02:54 UTC

R topics documented:

dfddm *Density of Ratcliff Diffusion Decision Model*

Description

Density function for the Ratcliff diffusion decision model (DDM) with following parameters: `a` (threshold separation), `v` (drift rate), `t0` (non-decision time/response time constant), `w` (relative starting point), `sv` (inter-trial variability of drift), and `sigma` (diffusion coefficient of underlying Wiener process).

Usage

```
dfddm(
  rt,
  response,
  a,
  v,
  t0,
  w = 0.5,
  sv = 0,
  sigma = 1,
  err_tol = 1e-06,
  log = FALSE,
  switch_mech = "eff_rt",
  switch_thresh = 0.8,
  n_terms_small = "SWSE",
  summation_small = "2017"
)
```

Arguments

`rt` A vector of response times (in seconds). If a response time is non-positve, then its density will evaluate to 0 if log = FALSE and $-\infty$ if log = TRUE.

`response` Binary response(s) that correspond(s) to either the "lower" or "upper" threshold. This model parameter can either be a singular value or a vector. The value(s) in 'response' can be of the following data types:

- integers or doubles (1 → "lower", 2 → "upper");
- factors (the first level gets mapped to "lower", and the second level gets mapped to "upper"; any additional levels are ignored).
- strings (only the first character is checked, "L" → "lower" or "U" → "upper", case insensitive);
- logicals (FALSE → "lower", TRUE → "upper");

`a` Threshold separation. Amount of information that is considered for a decision. Large values indicate a conservative decisional style. Allowed range: $0 < \texttt{a}$. Typical range: $0.5 < \texttt{a} < 2$.

`v` Drift rate. Average slope of the information accumulation process. The drift gives information about the speed and direction of the accumulation of information. Large (absolute) values of drift indicate a good performance. If received information supports the response linked to the upper threshold, then the sign will be positive; similarly a negative value indicates that the received information supports the response linked to the lower threshold. Allowed range: `v` is a real number. Typical range: $-5 < \texttt{v} < 5$.

`t0` Non-decision time or response time constant (in seconds). Lower bound for the duration of all non-decisional processes (encoding and response execution). If this value is greater than `rt`, then the resulting density is returned as if $\texttt{rt} \leq 0$. Allowed range: $0 \leq \texttt{t0}$. Typical range: $0.1 < \texttt{t0} < 0.5$.

`w` Relative starting point. Indicator of an a priori bias in decision making. When the relative starting point `w` deviates from 0.5, the amount of information necessary for a decision differs between response alternatives. Allowed range: $0 < \texttt{w} < 1$. Default value is 0.5 (i.e., no bias).

`sv` Inter-trial-variability of drift rate. Standard deviation of a normal distribution with mean `v` describing the distribution of actual drift rates from specific trials. Values different from 0 can predict slow errors. Allowed range: $0 \leq \texttt{sv}$. Typical range: $0 < \texttt{sv} < 2$. Default value is 0, which indicates no drift in the function call. See Details for more information.

`sigma` Diffusion coefficient of the underlying Wiener process. Allowed range: $0 < \texttt{sigma}$. Default value is 1. This parameter simply scales the parameters `a`, `v`, and `sv` as follows. See Details for more information.

- `a` → `a / sigma`
- `v` → `v / sigma`
- `sv` → `sv / sigma`

.

`err_tol` Allowed error tolerance of the density function. The density function contains an infinite sum that must be approximated; this parameter is the upper bound of the total error incurred from this approximation (in absolute value). See Details for more information. Default is $1e-6$.

`log` Logical; if `TRUE`, probabilities p are given as $log(p)$. Default is `FALSE`, which gives the density on the probability scale.

`switch_mech` Which switching mechanism to use in the choice of the "large-time" or "small-time" density function. Can be one of {`"eff_rt"`, `"terms_large"`, `"terms"`, `"small"`, `"large"`}. Note that the large-time approximation is unstable for small effective response times $((\texttt{rt}-\texttt{t0})\,/(\texttt{a}*\texttt{a}) < 0.009)$. See Details for more information. Default is `"eff_rt"`.

`switch_thresh` Threshold for determining whether the effective response time $((\texttt{rt}-\texttt{t0})\,/(\texttt{a}*\texttt{a}))$ is "large" or "small". This parameter is only considered if `switch_mech = "eff_rt"` or `switch_mech = "terms_large"`. If `switch_mech = "eff_rt"`, an effective response time greater than `switch_thresh` is considered "large", and the "large-time" variant of the density function is used; otherwise, the "small-time" variant of the density function is used. The default is 0.8. If `switch_mech = "terms_large"`, this parameter is treated as $ceiling($`switch_thresh`$)$; the smallest integer that is not less than `switch_thresh`. In this case, the default is $ceiling(0.8) = 1$. See the `switch_mech` section of Details for more information. Note that if `switch_thresh`≤ 0, then the effective response time is always treated as "large"; contrarily, if `switch_thresh` = $-\infty$ then the effective response time is always treated as "small". However, it is better to simply set `switch_mech = "large"` or `switch_mech = "small"` to always use the "large-time" or "small-time" variant, respectively.

`n_terms_small` Which method to use for calculating the "small-time" approximation to the density function. Only applicable if `switch_mech = "terms"` or `switch_mech = "small"`; all other values of `switch_mech` cause this parameter to be ignored. If `switch_mech = "small"`, the allowed values are `"SWSE"`, `"Gondan"`, and `"Navarro"`. The default value is `"SWSE"`. If `switch_mech = "terms"`, the allowed values are `"Gondan"` and `"Navarro"`. Note that if the user inputs `switch_mech = "terms"`, then the user must also explicitly input either `n_terms_small = "Gondan"` or `n_terms_small = "Navarro"`. See Details for more information.

`summation_small`

Which style of summation to use for the small-time approximation to the infinite sum. Can be one of {`"2017"`, `"2014"`}. Only applicable if `switch_mech` is one of {`"eff_rt"`, `"terms_large"`, `"terms"`, `"small"`}. See Details for more information. Default is `"2017"`.

Details

All of the model inputs and parameters (`rt`, `response`, `a`, `v`, `t0`, `w`, `sv`, `sigma`) can be input as a single value or as a vector of values. If input as a vector of values, then the standard `R` recycling rules apply to ensure all inputs are of the same length.

The default settings of `switch_mech = "eff_rt"`, `switch_thresh = "0.8"`, `n_terms_small = "SWSE"`, `summation_small = "2017"` produce the fastest and most accurate results, as shown in our associated paper.

`sv` - Both the "small-time" and "large-time" variants of the density function have two further variants: one with a constant drift rate `v` (i.e., `sv` $= 0$), and one with a variable drift rate `v` (i.e., `sv` > 0). The details of the differences between these two density functions can be found in our associated paper. To use the density function with a constant drift rate, leave the parameter `sv` to its default value of `sv = 0`, as this will indicate no drift to the function. To use the density function with a variable drift rate, set the parameter `sv` to some non-negative value, i.e., `sv` > 0.

sigma - The default value of this parameter is 1 because it only scales the parameters a, v, and sv, as shown above. However, other formulations of the DDM may set sigma = 0.1 (see Ratcliff (1978), the fourth reference), so care must be taken when comparing the results of different formulations.

err_tol - The density function is composed of an infinite sum (that must be approximated) and a multiplicative term outside the infinite sum, m. The total error of the approximation is the error incurred from truncating the infinite sum multiplied by m. Thus, to ensure that the total error is bounded by the user-provided error tolerance, err_tol, the approximation to the infinite sum uses a modified error tolerance (err_tol$/m$). If the error tolerance is small and m is large, the modified error tolerance can underflow to 0. To protect against this, we check that the modified error tolerance is at least $1e - 300$ (near the smallest value that is representable by a floating point number with double precision). If the modified error tolerance is smaller than this threshold, then we silently change it to $1e-300$. This case should only be encountered if extreme values of error tolerance are used (i.e., on the scale of $1e - 300$).

switch_mech - The density function for the DDM has traditionally been written in two forms: a "large-time" variant, and a "small-time" variant (Navarro and Fuss, 2009). These two forms are more efficient at calculating the density for large and small response times, respectively. The parameter switch_mech determines how dfddm decides which of these two variants is used. switch_mech = "small" uses only the "small-time" variant, and switch_mech = "large" uses only the "large-time" variant. The "large-time" variant is unstable for small effective response times ((rt$-$t0) /(a$*$a) < 0.009) and may produce inaccurate densities; thus, we do not recommend using the switch_mech = "large" option if the inputs may contain such small effective response times. To circumvent this accuracy issue and resolve the differing efficiencies of the "large-time" and "small-time" variants, there are three switching mechanisms that can be used to determine which of the two variants is more efficient. First, switch_mech = "terms" is the traditional approach and pre-calculates the number of terms required for the "large-time" and "small-time" sums, and then uses whichever variant requires fewer terms. This is the mechanism used in the Navarro and Fuss (2009) and Gondan, Blurton, and Kesselmeier (2014) papers. Second, switch_mech = "terms_large" pre-calculates the number of terms only for the "large-time" variant, and compares that to the constant value of $ceil$(switch_thresh) (default value of $ceil(0.8) = 1$) to determine if the "large-time" variant is sufficiently efficient. Third, switch_mech = "eff_rt" determines whether a given effective response time ((rt$-$t0) /(a$*$a)) is considered "large" or "small" by comparing it to the value of the parameter switch_thresh (default value of 0.8); it then uses the corresponding variant. Both switch_mech = "terms_large" and switch_mech = "eff_rt" only use the SWSE "small-time" approximation in the case that the "small-time" variant is more efficient. switch_mech = "eff_rt" is the most efficient method to determine which variant of the density function should be used.

switch_thresh - This parameter determines what effective response times (rt/(a $*$a)) are "large" and "small". The paper_analysis folder in the fddm GitHub repository contains plots showing the relative efficiencies of a range of values for the switch_thresh parameter when used with switch_mech = "eff_rt" and also switch_mech = "terms_large". Note that this parameter changed name and purpose with the release of fddm version 0.5-0.

n_terms_small - The "small-time" variant has three different methods for how to truncate the infinite sum in the density function. These different methods are discussed extensively in our associated paper, but the key distinction is that n_terms_small = "SWSE" uses a new method of truncating the infinite sum. The n_terms_small = "SWSE" method is currently recommended (when possible) because it is the fastest and most stable algorithm when used with switch_mech = "eff_rt".

summation_small - The "large-time" variant of the density function does not have any further variants, but the "small-time" variant has more options with respect to evaluating the infinite sum.

There are two equivalent styles of summation, summation_small = "2017" and summation_small = "2014", of which the "2017" version evaluates slightly faster and thus earns our recommendation. These different styles of summation are discussed in our associated paper.

Value

A vector containing the densities of the DDM with precision err_tol whose length matches that of the longest input parameter (usually rt).

References

Navarro, D. J., & Fuss, I. G. (2009). Fast and accurate calculations for first-passage times in Wiener diffusion models. Journal of Mathematical Psychology, 53(4), 222-230.

Gondan, M., Blurton, S. P., & Kesselmeier, M. (2014). Even faster and even more accurate first-passage time densities and distributions for the Wiener diffusion model. Journal of Mathematical Psychology, 60, 20-22.

Blurton, S. P., Kesselmeier, M., & Gondan, M. (2017). The first-passage time distribution for the diffusion model with variable drift. Journal of Mathematical Psychology, 76, 7-12.

Ratcliff, R. (1978). A theory of memory retrieval. Psychological review, 85(2), 59.

Examples

```
# minimal example
dfddm(rt = 1.2, response = "lower", a = 1, v = -1, t0 = 0.3)

# example with all function parameters set to default or a practical value
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = 1, v = -1, t0 = 0.2, w = 0.5, sv = 0, sigma = 1,
      err_tol = 1e-6, log = FALSE, switch_mech = "eff_rt", switch_thresh = 0.8,
      n_terms_small = "SWSE", summation_small = "2017")

# example of mismatched input lengths
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = c(1, 3), v = c(-2, 2, 2, -2, 2, -2),
      t0 = 0.3, w = c(0.4, 0.5, 0.6), sv = 0.9,
      err_tol = 1e-10, log = FALSE, switch_mech = "large",
      summation_small = "2017")

# example with Wiener diffusion coefficient (sigma) not equal to 1
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = 1, v = -1, t0 = 0.3, w = 0.5, sv = 0, sigma = 0.1,
      err_tol = 1e-10, log = TRUE, switch_mech = "terms_large",
      switch_thresh = 1, summation_small = "2017")


### examples of different response inputs

# integer
```

```
resp_int <- as.integer(c(1, 2, 2, 1, 2, 1))
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_int,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, switch_mech = "eff_rt",
      summation_small = "2017")

# double
resp_dbl <- as.double(c(1, 2, 2, 1, 2, 1))
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_dbl,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, switch_mech = "eff_rt",
      summation_small = "2017")

# factor (first level is mapped to "lower")
days <- c("Monday", "Friday", "Friday", "Monday", "Friday", "Monday")
resp_fac <- factor(days, levels = c("Monday", "Friday"))
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_fac,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, switch_mech = "eff_rt",
      summation_small = "2017")

# string
resp_str <- c("lower", "upper", "upper", "lower", "upper", "lower")
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_str,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, switch_mech = "eff_rt",
      summation_small = "2017")

# logical
resp_log <- c(FALSE, TRUE, TRUE, FALSE, TRUE, FALSE)
dfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_log,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, switch_mech = "eff_rt",
      summation_small = "2017")
```

med_dec *Medicial decision data*

Description

Part of the accuracy and response time data presented in Trueblood et al. (2017) investigating medical decision making among medical professionals (pathologists) and novices (i.e., undergraduate students). The task of participants was to judge whether pictures of blood cells show cancerous cells (i.e., blast cells) or non-cancerous cells (i.e., non-blast cells). The current data set contains 200 decisions per participant (the "accuracy" condition from Trueblood et al.).

Usage

```
data(med_dec)
```

Format

A data frame with 11000 rows and 9 variables:

id identification number of the participant

group expertise of participant; "experienced", "inexperienced", or "novice". The first two levels refer to different type of medical professional (i.e., experts).

block block number

trial index of trial for each participant

classification true classification of the pictured cell; i.e. the correct response

difficulty adjudged difficulty of the task for the particular image

response response given by the participant; either "blast" or "non-blast"

rt the response time associated with the response, in seconds

stimulus the image file used for the specific trial

Details

At the beginning of the experiment, both novices and medical experts completed a training to familiarize themselves with blast cells. After that, each participant performed the main task in which they judged whether or not presented images were blast cells or non-blast cells. Among them, some of the cells were judged as easy and some as difficult trials by an additional group of experts. The current data set only contains the data from the "accuracy" condition (i.e., Trueblood et al. considered additional conditions that are not part of the current data set).

The relevant part of the method section for the accuracy condition from the original paper is as follows:

"The main task consisted of six blocks with 100 trials in each block. The main task was the same as the practice block, where participants were asked to identify single images. However, participants did not receive trial-by-trial feedback about their choices. They received feedback about their performance at the end of each block. The 100 trials in each block were composed of equal numbers of easy blast images, hard blast images, easy non-blast images, and hard non-blast images, fully randomized.

There were three manipulations across blocks: accuracy, speed, and bias. In the accuracy blocks, participants were instructed to respond as accurately as possible and were given 5 s to respond. [...] If they responded after the deadline, they received the message "Too Slow!" The 5-s [...] response windows for the accuracy [...] [condition was] based on the response time data from the three expert raters. The 0.975 quantile of the expert raters' response times was 4.96 s; thus, we set the accuracy response window to 5 s.

The order of the first three blocks was randomized but with the constraint that there was one block for each type of manipulation (i.e., accuracy, speed, and bias). The order of the last three blocks was identical to the order of the first three blocks."

Note that this dataset contains some negative response times that indicate a missing response (i.e., the response value for that trial is 'NA'). Take care in removing these values before using this dataset. See our Validity vignette for an example of use in an optimization setting.

Source

Trueblood, J.S., Holmes, W.R., Seegmiller, A.C. et al. The impact of speed and bias on the cognitive processes of experts and novices in medical image decision-making. Cogn. Research 3, 28 (2018). https://doi.org/10.1186/s41235-018-0119-2

Examples

```
data("med_dec", package = "fddm")
str(med_dec)

## number of participants per expertise condition:
aggregate(id ~ group, med_dec, function(x) length(unique(x)))

## number of trials per participant
aggregate(rt ~ group + id, med_dec, length)
```

pfddm *Distribution of Ratcliff Diffusion Decision Model*

Description

Distribution function for the Ratcliff diffusion decision model (DDM) with following parameters: `a` (threshold separation), `v` (drift rate), `t0` (non-decision time/response time constant), `w` (relative starting point), `sv` (inter-trial variability of drift), and `sigma` (diffusion coefficient of underlying Wiener process).

Usage

```
pfddm(
  rt,
  response,
  a,
  v,
  t0,
  w = 0.5,
  sv = 0,
  sigma = 1,
  err_tol = 1e-06,
  log = FALSE,
  method = "Mills"
)
```

Arguments

`rt` A vector of response times (in seconds). If a response time is non-positve, then its density will evaluate to 0 if log = FALSE and $-\infty$ if log = TRUE.

`response` Binary response(s) that correspond(s) to either the "lower" or "upper" threshold. This model parameter can either be a singular value or a vector. The value(s) in 'response' can be of the following data types:

- integers or doubles (1 → "lower", 2 → "upper");
- factors (the first level gets mapped to "lower", and the second level gets mapped to "upper"; any additional levels are ignored).
- strings (only the first character is checked, "L" → "lower" or "U" → "upper", case insensitive);
- logicals (FALSE → "lower", TRUE → "upper");

`a` Threshold separation. Amount of information that is considered for a decision. Large values indicate a conservative decisional style. Allowed range: $0 <$ `a`. Typical range: $0.5 <$ `a` < 2.

`v` Drift rate. Average slope of the information accumulation process. The drift gives information about the speed and direction of the accumulation of information. Large (absolute) values of drift indicate a good performance. If received information supports the response linked to the upper threshold, then the sign will be positive; similarly a negative value indicates that the received information supports the response linked to the lower threshold. Allowed range: `v` is a real number. Typical range: $-5 <$ `v` < 5.

`t0` Non-decision time or response time constant (in seconds). Lower bound for the duration of all non-decisional processes (encoding and response execution). If this value is greater than `rt`, then the resulting density is returned as if `rt` ≤ 0. Allowed range: $0 \leq$ `t0`. Typical range: $0.1 <$ `t0` < 0.5.

`w` Relative starting point. Indicator of an a priori bias in decision making. When the relative starting point `w` deviates from 0.5, the amount of information necessary for a decision differs between response alternatives. Allowed range: $0 <$ `w` < 1. Default value is 0.5 (i.e., no bias).

`sv` Inter-trial-variability of drift rate. Standard deviation of a normal distribution with mean `v` describing the distribution of actual drift rates from specific trials. Values different from 0 can predict slow errors. Allowed range: $0 \leq$ `sv`. Typical range: $0 <$ `sv` < 2. Default value is 0, which indicates no drift in the function call. See Details for more information.

`sigma` Diffusion coefficient of the underlying Wiener process. Allowed range: $0 <$ `sigma`. Default value is 1. This parameter simply scales the parameters `a`, `v`, and `sv` as follows. See Details for more information.

- `a` → `a` / `sigma`
- `v` → `v` / `sigma`
- `sv` → `sv` / `sigma`

.

`err_tol` Allowed error tolerance of the density function. Since the density function contains an infinite sum, this parameter defines the precision of the approximation to that infinite sum. If the provided error tolerance is less than $1e-300$, it is set to $1e-300$. Default is $1e-6$.

`log` Logical; if `TRUE`, probabilities p are given as $log(p)$. Default is `FALSE`.

`method` Which method of calculating the CDF to use. Can be one of {`"Mills"`, `"NCDF"`}, case-insensitive. Default is `"Mills"`.

Details

All of the model inputs and parameters (rt, response, a, v, t0, w, sv, sigma) can be input as a single value or as a vector of values. If input as a vector of values, then the standard R input wrapping will occur.

sigma - The default value of this parameter is 1 because it only scales the parameters a, v, and sv, as shown above. However, other formulations of the DDM may set sigma = 0.1 (see Ratcliff (1978), the fourth reference), so care must be taken when comparing the results of different formulations.

Value

A vector containing the distribution of the DDM with precision err_tol whose length matches that of the longest input parameter (usually rt).

References

Navarro, D. J., & Fuss, I. G. (2009). Fast and accurate calculations for first-passage times in Wiener diffusion models. Journal of Mathematical Psychology, 53(4), 222-230.

Gondan, M., Blurton, S. P., & Kesselmeier, M. (2014). Even faster and even more accurate first-passage time densities and distributions for the Wiener diffusion model. Journal of Mathematical Psychology, 60, 20-22.

Blurton, S. P., Kesselmeier, M., & Gondan, M. (2017). The first-passage time distribution for the diffusion model with variable drift. Journal of Mathematical Psychology, 76, 7-12.

Ratcliff, R. (1978). A theory of memory retrieval. Psychological review, 85(2), 59.

Examples

```
# minimal example
pfddm(rt = 1.2, response = "lower", a = 1, v = -1, t0 = 0.3)

# example with all function parameters set to default or a practical value
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = 1, v = -1, t0 = 0.2, w = 0.5, sv = 0, sigma = 1,
      err_tol = 1e-6, log = FALSE, method = "Mills")

# example of mismatched input lengths
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = c(1, 3), v = c(-2, 2, 2, -2, 2, -2),
      t0 = 0.3, w = c(0.4, 0.5, 0.6), sv = 0.9,
      err_tol = 1e-10, log = FALSE, method = "NCDF")

# example with Wiener diffusion coefficient (sigma) not equal to 1
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3),
      response = c("lower", "upper", "upper", "lower", "upper", "lower"),
      a = 1, v = -1, t0 = 0.3, w = 0.5, sv = 0, sigma = 0.1,
      err_tol = 1e-10, log = TRUE, method = "Mills")
```

```
### examples of different response inputs

# integer
resp_int <- as.integer(c(1, 2, 2, 1, 2, 1))
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_int,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, method = "Mills")

# double
resp_dbl <- as.double(c(1, 2, 2, 1, 2, 1))
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_dbl,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, method = "Mills")

# factor (first level is mapped to "lower")
days <- c("Monday", "Friday", "Friday", "Monday", "Friday", "Monday")
resp_fac <- factor(days, levels = c("Monday", "Friday"))
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_fac,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, method = "Mills")

# string
resp_str <- c("lower", "upper", "upper", "lower", "upper", "lower")
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_str,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, method = "Mills")

# logical
resp_log <- c(FALSE, TRUE, TRUE, FALSE, TRUE, FALSE)
pfddm(rt = c(1.2, 0.9, 1.1, 1.4, 0.8, 1.3), response = resp_log,
      a = 1, v = -2, t0 = 0.3, w = 0.5, sv = 0.1,
      err_tol = 1e-10, log = FALSE, method = "Mills")
```

Index