

# Package: flankr (via r-universe)

May 20, 2026

**Title** Implementing Computational Models of Attentional Selectivity

**Version** 1.2.0.9999

**Description** A set of methods to simulate from and fit computational models of attentional selectivity. The package implements the dual-stage two-phase (DSTP) model of Hübner et al. (2010) <[doi:10.1037/a0019471](https://doi.org/10.1037/a0019471)>, and the shrinking spotlight (SSP) model of White et al. (2011) <[doi:10.1016/j.cogpsych.2011.08.001](https://doi.org/10.1016/j.cogpsych.2011.08.001)>.

**Depends** R (>= 4.0)

**Imports** Rcpp

**LinkingTo** Rcpp

**Suggests** knitr, rmarkdown

**License** GPL-3

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**URL** <https://github.com/JimGrange/flankr>

**BugReports** <https://github.com/JimGrange/flankr/issues>

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

**Date/Publication** 2025-09-19 05:41:27 UTC

**RemoteUrl** <https://github.com/jimgrange/flankr>

**RemoteRef** HEAD

**RemoteSha** d64813d687f8536d87d00bacc5a2aaa98d04da52

## Contents

caf . . . . .	2
cdf . . . . .	3
exampleData . . . . .	5
fitDSTP . . . . .	6

fitDSTP_fixed . . . . .	8
fitMultipleDSTP . . . . .	10
fitMultipleDSTP_fixed . . . . .	12
fitMultipleSSP . . . . .	14
fitMultipleSSP_fixed . . . . .	17
fitSSP . . . . .	19
fitSSP_fixed . . . . .	21
plotFitDSTP . . . . .	23
plotFitSSP . . . . .	25
simulateDSTP . . . . .	27
simulateSSP . . . . .	28

## Index 29

---

caf	<i>Find conditional accuracy function (CAF) values for a single condition</i>
-----	---

---

## Description

caf takes a data frame for a single experimental condition and returns a vector of requested conditional accuracy function (CAF) values.

## Usage

```
caf(data, quantiles = c(0.25, 0.5, 0.75), multipleSubjects = TRUE)
```

## Arguments

data	A data frame containing the data to be passed to the function. At the very least, the data frame must contain columns named "accuracy" logging the accuracy (1 for correct, 0 for error) and "rt" containing the response time data. If the user wishes to find the average CAFs across multiple subjects, then another column must be included ("subject") with numbers identifying unique subjects. See ?exampleData for a data frame formatted correctly.
quantiles	The quantile values to be found by the function. By default, the function finds the accuracy for the .25, .5, and .75 quantiles.
multipleSubjects	Inform the function whether the data frame contains data from multiple subjects. If set to TRUE, the function returns the average CAF values across all subjects. If set to FALSE, the function assumes all data being passed is just from one subject.

## Details

The function only deals with one experimental condition. There is another function (cafAll) which will return CAFs for all experimental conditions. If there are more than one subject in the data frame being passed to this function, the function first finds the CAF values for each subject, and then takes the average for each quantile. This average is then returned to the user.

**Value**

Returns a matrix with response time (row 1) and accuracy (row 2) for each quantile of CAF requested.

**Examples**

```
### example of multiple subjects and default quantile values

# only select the congruent data from the example data set
data <- subset(exampleData, exampleData$congruency == "congruent")

# get the CDFs
getCaf <- caf(data)

#-- example of single subject and different quantile values

# only select subject 1 from the example data. Also, select only the
# "absent" condition and incongruent trials. This is an example when working
# with multiple conditions (besides target congruency).
data <- subset(exampleData, exampleData$subject == 1 &
  exampleData$condition == "absent" &
  exampleData$congruency == "incongruent")

# set new quantile values
newQuantiles <- c(.2, .4, .6, .8)

# get the CAFs
getCaf <- caf(data, quantiles = newQuantiles, multipleSubjects = FALSE)
```

---

cdf	<i>Find cumulative distribution function (CDF) values for a single condition</i>
-----	--

---

**Description**

cdf takes a data frame for a single experimental condition and returns a vector of requested CDF values.

**Usage**

```
cdf(
  data,
  quantiles = c(0.1, 0.3, 0.5, 0.7, 0.9),
  correctTrials = 1,
  multipleSubjects = TRUE
)
```

## Arguments

data	A data frame containing the data to be passed to the function. At the very least, the data frame must contain columns named "accuracy" logging the accuracy (1 for correct, 0 for error) and "rt" containing the response time data. If the user wishes to find the average CDFs across multiple subjects, then another column must be included ("subject") with numbers identifying unique subjects. See <code>?exampleData</code> for a data frame formatted correctly.
quantiles	The quantile values to be found by the function. By default, the function finds the .1, .3, .5, .7, and .9 CDF values.
correctTrials	If set to 1, the function will find the CDFs of correct trials. Set to 2 to find the CDFs of error trials. Set to 3 to find CDFs of ALL trials. Note, though, that CDFs of error trials may be less accurate due to usually-low number of error trials.
multipleSubjects	Inform the function whether the data frame contains data from multiple subjects. If set to TRUE, the function returns the average CDF values across all subjects. If set to FALSE, the function assumes all data being passed is just from one subject.

## Details

The function only deals with one experimental condition. There is another function (`cdfAll`) which will return CDFs for all experimental conditions. If there are more than one subject in the data frame being passed to this function, the function first finds the CDF values for each subject, and then takes the average for each quantile. This average is then returned to the user.

## Value

Returns a vector of response times for each quantile of CDF requested.

## Examples

```
### example of multiple subjects and default quantile values

# only select the congruent data from the example data set
data <- subset(exampleData, exampleData$congruency == "congruent")

# get the CDFs
getCDF <- cdf(data)

### example of single subject and different quantile values

# only select subject 1 from the example data. Also, select only the
# "absent" condition and incongruent trials. This is an example when working
# with multiple conditions (besides target congruency).
data <- subset(exampleData, exampleData$subject == 1 &
  exampleData$condition == "absent" &
  exampleData$congruency == "incongruent")
```

```
# set new quantile values
newQuantiles <- c(.1, .2, .3, .4, .5, .6, .7, .8, .9)

# get the CDFs
getCDF <- cdf(data, quantiles = newQuantiles, multipleSubjects = FALSE)
```

---

exampleData

*Example response time data set for multiple subjects.*

---

## Description

An example data set containing multiple participants' data for a response time study involving two experimental conditions, as well as the congruency manipulation. The data includes response time and accuracy.

## Usage

```
exampleData
```

## Format

A data frame with 12524 rows and 5 variables:

**subject** The subject identification number. Note that some participants were removed due to poor accuracy.

**condition** The experimental condition (2 in this example). It logs the presence/absence of an auditory tone before stimulus onset.

**congruency** The congruency of the flanker stimulus.

**accuracy** Accuracy of the response; 1 = correct, 0 = error

**rt** Response time, coded in seconds.

## Details

This data is taken from Grange, J.A. (in preparation). The effect of accessory stimuli on choice response time in the flanker task.

---

fitDSTP

*Fit the DSTP model to human data*


---

### Description

fitDSTP fits the DSTP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously).

### Usage

```
fitDSTP(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

```
fit_dstp(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).
parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, C, driftTarget, driftFlanker, diftStimSelection, driftRS2, ter.

<code>cdfs</code>	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
<code>cafs</code>	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
<code>maxParms</code>	A vector containing upper limits on possible parameter values.
<code>nTrials</code>	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
<code>multipleSubjects</code>	A boolean stating whether the fit is to multiple subjects ( <code>multipleSubjects = TRUE</code> ) or to a single subject ( <code>multipleSubjects = FALSE</code> ).
<code>seed</code>	The value for the <code>set.seed</code> function to set random generation state.

## Details

This function can be employed by the user to find the best-fitting parameters of the DSTP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data.

## Value

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio ( $G^2$ ) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

## Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fit the model to the condition "present" in the example data set using
# the default settings in the model.
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)

fit <- fitDSTP(data = exampleData, conditionName = "present", nTrials = 100)
```

---

fitDSTP\_fixed

*Fit the DSTP model to human data with some fixed parameters*


---

### Description

fitDSTP\_fixed fits the DSTP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously). This function allows the user to fix some parameters (using the fixed variable).

### Usage

```
fitDSTP_fixed(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE),
  seed = NULL
)
```

```
fit_dstp_fixed(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE),
  seed = NULL
)
```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).

parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, C, driftTarget, driftFlanker, driftStimSelection, driftRS2, ter.
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
fixed	A vector of TRUE/FALSE stating whether each parameter should be fixed (TRUE) or free (FALSE) during the fitting routine. Must be in the order: A, C, driftTarget, driftFlanker, driftStimSelection, driftRS2, ter.
seed	The value for the set.seed function to set random generation state.

## Details

This function can be employed by the user to find the best-fitting parameters of the DSTP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data.

## Value

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio (G2) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

## Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fix the first parameter (A) during the fit.
parms <- c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24)
fixed <- c(TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE)

# fit the model.
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
```

```
fit <- fitDSTP_fixed(exampleData, conditionName = "present", parms = parms,
                    fixed = fixed, nTrials = 100)
```

---

fitMultipleDSTP

*Fit the DSTP model to human data with mutiple starting parmaeters*


---

## Description

fitMultipleDSTP fits the DSTP model to a single experimental condition of human data (besides congruency, which it accounts for simutaneously). This function explores multiple starting parameters.

## Usage

```
fitMultipleDSTP(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

```
fit_multiple_dstp(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

```
fit_multiple_dstp(
  data,
```

```

conditionName = NULL,
parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
var = 10,
nParms = 20,
cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
cafs = c(0.25, 0.5, 0.75),
maxParms = c(1, 1, 1, 1, 1, 2, 1),
nTrials = 50000,
multipleSubjects = TRUE,
seed = NULL
)

```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).
parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, C, driftTarget, driftFlanker, driftStimSelection, driftRS2, ter. These parameters will be the starting point for the random parameters.
var	An integer stating the percentage of each parameter value that should be used for finding random parameter starting points.
nParms	An integer stating how many random starting points to explore
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
seed	The value for the set.seed function to set random generation state.

### Details

This function can be employed by the user to find the best-fitting parameters of the DSTP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the

Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data. Multiple starting points of parameters are used.

### Value

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio (G2) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

### Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fit the model to the condition "present" in the example data set using
# the default settings in the model.
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)

fit <- fitMultipleDSTP(data = exampleData, conditionName = "present", nTrials = 100)
```

---

`fitMultipleDSTP_fixed` *Fit the DSTP model to human data with multiple starting parameters with some fixed parameters*

---

### Description

`fitMultipleDSTP_fixed` fits the DSTP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously). This function explores multiple starting parameters and allows user to fix model parameters.

### Usage

```
fitMultipleDSTP_fixed(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE)
```

```

)

fit_multiple_dstp_fixed(
  data,
  conditionName = NULL,
  parms = c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 1, 2, 1),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE)
)

```

### Arguments

<code>data</code>	A data frame containing human data. See <code>?exampleData</code> for data formatted correctly.
<code>conditionName</code>	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., <code>conditionName</code> is set to <code>NULL</code> ).
<code>parms</code>	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, C, <code>driftTarget</code> , <code>driftFlanker</code> , <code>diftStimSelection</code> , <code>driftRS2</code> , <code>ter</code> . These parameters will be the starting point for the random parameters.
<code>var</code>	An integer stating the percentage of each parameter value that should be used for finding random parameter starting points.
<code>nParms</code>	An integer stating how many random starting points to explore
<code>cdfs</code>	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
<code>cafs</code>	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
<code>maxParms</code>	A vector containing upper limits on possible parameter values.
<code>nTrials</code>	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
<code>multipleSubjects</code>	A boolean stating whether the fit is to multiple subjects ( <code>multipleSubjects = TRUE</code> ) or to a single subject ( <code>multipleSubjects = FALSE</code> ).
<code>fixed</code>	A vector of <code>TRUE/FALSE</code> stating whether each parameter should be fixed ( <code>TRUE</code> ) or free ( <code>FALSE</code> ) during the fitting routine. Must be in the order: A, C, <code>driftTarget</code> , <code>driftFlanker</code> , <code>diftStimSelection</code> , <code>driftRS2</code> , <code>ter</code> .

**Details**

This function can be employed by the user to find the best-fitting parameters of the DSTP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data. Multiple starting points of parameters are used.

**Value**

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio (G2) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

**Examples**

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fit the model whilst fixing the first parameter (A)
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
parms <- c(0.145, 0.08, 0.1, 0.07, 0.325, 1.3, 0.24)
fixed <- c(TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE)

fit <- fitMultipleDSTP_fixed(exampleData, conditionName = "present",
                             parms = parms, fixed = fixed, nTrials = 100)
```

---

fitMultipleSSP

*Fit the SSP model to human data with multiple starting parameters*


---

**Description**

`fitSSPMultiple` fits the SSP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously).

**Usage**

```
fitMultipleSSP(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
```

```

    cafs = c(0.25, 0.5, 0.75),
    maxParms = c(1, 1, 1, 1, 2),
    nTrials = 50000,
    multipleSubjects = TRUE,
    seed = NULL
)

fit_multiple_ssp(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)

fit_multiple_ssp(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)

```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).
parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, ter, p, rd, sda. These parameters will be the starting point for the random parameters.

var	An integer stating the percentage of each parameter value that should be used for finding random parameter starting points.
nParms	An integer stating how many random starting points to explore
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
seed	The value for the set.seed function to set random generation state.

### Details

This function can be employed by the user to find the best-fitting parameters of the SSP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data.

### Value

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio ( $G^2$ ) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

### Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fit the model to the condition "present" in the example data set using
# the default settings in the model.
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)

fit <- fitMultipleSSP(data = exampleData, conditionName = "present",
                     nTrials = 100)
```

---

fitMultipleSSP\_fixed *Fit the SSP model to human data with mutiple starting parmaeters with some fixed parameters*

---

### Description

fitMultipleSSP\_fixed fits the SSP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously). This function explores multiple starting parameters and allows user to fix model parameters.

### Usage

```
fitMultipleSSP_fixed(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE)
)
```

```
fit_multiple_ssp_fixed(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  var = 10,
  nParms = 20,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE)
)
```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present").

	The function by default assumes no additional condition (e.g., conditionName is set to NULL).
parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, ter, p, rd, sda. These parameters will be the starting point for the random parameters.
var	An integer stating the percentage of each parameter value that should be used for finding random parameter starting points.
nParms	An integer stating how many random starting points to explore
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
fixed	A vector of TRUE/FALSE stating whether each parameter should be fixed (TRUE) or free (FALSE) during the fitting routine. Must be in the order: A, ter, p, rd, sda.

## Details

This function can be employed by the user to find the best-fitting parameters of the SSP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data. Multiple starting points of parameters are used.

## Value

**bestParameters** A vector of the best-fitting parameters found by the current fit run.

**g2** The value of Wilks likelihood ratio (G2) obtained by the current fit run.

**bBIC** The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence **bBIC** (binned BIC).

## Examples

```
data(exampleData)
# Fix the first parameter (A) during the fit.
parms = c(0.050, 0.300, 0.400, 0.050, 1.500)
fixed <- c(TRUE, FALSE, FALSE, FALSE, FALSE)
```

```
# run the model
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)

fit <- fitMultipleSSP_fixed(exampleData, conditionName = "present",
                           parms = parms, fixed = fixed, nTrials = 100)
```

---

fitSSP

*Fit the SSP model to human data*


---

### Description

fitSSP fits the SSP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously).

### Usage

```
fitSSP(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 3),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

```
fit_ssp(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 3),
  nTrials = 50000,
  multipleSubjects = TRUE,
  seed = NULL
)
```

### Arguments

**data** A data frame containing human data. See ?exampleData for data formatted correctly.

conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).
parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, ter, p, rd, sda.
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
seed	The value for the set.seed function to set random generation state.

### Details

This function can be employed by the user to find the best-fitting parameters of the SSP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data.

### Value

`bestParameters` A vector of the best-fitting parameters found by the current fit run.

`g2` The value of Wilks likelihood ratio ( $G^2$ ) obtained by the current fit run.

`bBIC` The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence `bBIC` (binned BIC).

### Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)

# Fit the model to the condition "present" in the example data set using
# the default settings in the model.
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
fit <- fitSSP(data = exampleData, conditionName = "present", nTrials = 100)
```

---

fitSSP\_fixed

*Fit the SSP model to human data with some fixed parameters*


---

### Description

fitSSP\_fixed fits the SSP model to a single experimental condition of human data (besides congruency, which it accounts for simultaneously). This function allows the user to fix some parameters (using the fixed variable).

### Usage

```
fitSSP_fixed(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE),
  seed = NULL
)
```

```
fit_ssp_fixed(
  data,
  conditionName = NULL,
  parms = c(0.05, 0.3, 0.4, 0.05, 1.5),
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  maxParms = c(1, 1, 1, 1, 2),
  nTrials = 50000,
  multipleSubjects = TRUE,
  fixed = c(FALSE, FALSE, FALSE, FALSE, FALSE),
  seed = NULL
)
```

### Arguments

data	A data frame containing human data. See ?exampleData for data formatted correctly.
conditionName	If there is an additional experimental manipulation (i.e., other than target congruency) the model can only be fit to one at a time. Tell the function which condition is currently being fit by passing a string to the function (e.g., "present"). The function by default assumes no additional condition (e.g., conditionName is set to NULL).

parms	A vector of starting parameters to use in the minimisation routine. Must be in the order: A, ter, p, rd, sda.
cdfs	A vector of quantile values for cumulative distribution functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
cafs	A vector of quantiles for conditional accuracy functions to be estimated from the human data. The model will attempt to find the best-fitting parameters that match this distributional data.
maxParms	A vector containing upper limits on possible parameter values.
nTrials	An integer stating how many trials to simulate per iteration of the fitting cycle for each congruency type.
multipleSubjects	A boolean stating whether the fit is to multiple subjects (multipleSubjects = TRUE) or to a single subject (multipleSubjects = FALSE).
fixed	A vector of TRUE/FALSE stating whether each parameter should be fixed (TRUE) or free (FALSE) during the fitting routine. Must be in the order: A, ter, p, rd, sda.
seed	The value for the set.seed function to set random generation state.

## Details

This function can be employed by the user to find the best-fitting parameters of the SSP model to fit the human data of a single experimental condition. The fitting procedure accounts for congruent and incongruent trials simultaneously. The fit is obtained by a gradient-descent method (using the Nelder-Mead method contained in R's `optim` function) and is fit to the proportion of data contained in human CDF and CAF distributional data.

## Value

**bestParameters** A vector of the best-fitting parameters found by the current fit run.

**g2** The value of Wilks likelihood ratio (G2) obtained by the current fit run.

**bBIC** The value of the Bayesian Information Criterion (BIC) obtained by the current fit run. This is calculated using the BIC equation for binned data, hence **bBIC** (binned BIC).

## Examples

```
# Load the example data the comes with the \code{flankr} package
data(exampleData)
# Fix the first parameter (A) during the fit.
parms = c(0.050, 0.300, 0.400, 0.050, 1.500)
fixed <- c(TRUE, FALSE, FALSE, FALSE, FALSE)

# fit the model
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)

fit <- fitSSP_fixed(exampleData, conditionName = "present", parms = parms,
```

```
fixed = fixed, nTrials = 100)
```

---

plotFitDSTP

*Plot the fit of the DSTP model to human data.*

---

## Description

plotFitDSTP will plot the fit of the model to human distributional data.

## Usage

```
plotFitDSTP(  
  modelFit,  
  data,  
  conditionName = NULL,  
  nTrials = 50000,  
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),  
  cafs = c(0.25, 0.5, 0.75),  
  multipleSubjects = TRUE,  
  seed = NULL  
)
```

```
plot_fit_dstp(  
  modelFit,  
  data,  
  conditionName = NULL,  
  nTrials = 50000,  
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),  
  cafs = c(0.25, 0.5, 0.75),  
  multipleSubjects = TRUE,  
  seed = NULL  
)
```

## Arguments

modelFit	The object obtained by the model fit.
data	The data frame of human data.
conditionName	The name of the condition that was fit. By default, it is set to conditionName = NULL.
nTrials	How many trials used to generate the model's best predictions. This should be higher than that used to fit the model.
cdfs	The cut-off points for the cumulative distribution functions.
cafs	The cut-off points for the conditional accuracy functions.

`multipleSubjects` A boolean stating whether the fit is to multiple subjects (`multipleSubjects = TRUE`) or to a single subject (`multipleSubjects = FALSE`).

`seed` The value for the `set.seed` function to set random generation state.

### Details

This function is passed the object obtained by the model fitting procedure, as well as the human data and the condition that was fitted by the routine. The function simulates 100,000 trials (by default) using the best-fitting parameters found by the fit procedure. This synthetic data is then considered as the model's best predictions. The function then provides a plot of the model fit to cumulative distribution functions (CDFs) of correct response time, and conditional accuracy functions (CAFs) to show fit to accuracy data. The function also returns the data used to plot the fit so that the user can use their own plotting methods.

### Value

`cdfs` The CDF values requested by the user.

`cafs` The CAF values requested by the user.

`humanConCDFs` The response time cut-off values for each CDF bin for congruent human data.

`humanInconCDFs` The response time cut-off values for each CDF bin for incongruent human data.

`humanConCAFsRT` The mean response times for each bin of the CAF functions for congruent human data.

`humanInconCAFsRT` The mean response times for each bin of the CAF functions for incongruent human data.

`humanConCAFsError` The percent accuracy for each bin of the CAF functions for congruent human data.

`humanInconCAFsError` The percent accuracy for each bin of the CAF functions for incongruent human data.

`modelConCDFs` The quantile cut-off points for the model predictions for congruent data. A perfect fit would match the `cdfs` asked for by the user (e.g., .1, .3, .5, .7, .9).

`modelInconCDFs` The quantile cut-off points for the model predictions for incongruent data. A perfect fit would match the `cdfs` asked for by the user (e.g., .1, .3, .5, .7, .9).

`modelConCAFs` The percentage accuracy predicted for each CAF bin by the model for congruent data.

`modelInconCAFs` The percentage accuracy predicted for each CAF bin by the model for incongruent data.

### Examples

```
# Assume that the model was just fit to the data contained in
# \code{exampleData} (condition "present") and saved to the variable called
# "fit", then we can obtain a plot of that fit by using the following:
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
fit <- fitDSTP(data = exampleData, conditionName = "present", nTrials = 100)
```

```
plot <- plotFitDSTP(modelFit = fit, data = exampleData,
                   conditionName = "present", nTrials = 100)
```

---

plotFitSSP

*Plot the fit of the SSP model to human data.*


---

## Description

plotFitSSP will plot the fit of the model to human distributional data.

## Usage

```
plotFitSSP(
  modelFit,
  data,
  conditionName = NULL,
  nTrials = 50000,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  multipleSubjects = TRUE,
  seed = NULL
)
```

```
plot_fit_ssp(
  modelFit,
  data,
  conditionName = NULL,
  nTrials = 50000,
  cdfs = c(0.1, 0.3, 0.5, 0.7, 0.9),
  cafs = c(0.25, 0.5, 0.75),
  multipleSubjects = TRUE,
  seed = NULL
)
```

## Arguments

modelFit	The object obtained by the model fit.
data	The data frame of human data.
conditionName	The name of the condition that was fit. By default, it is set to conditionName = NULL.
nTrials	How many trials used to generate the model's best predictions. This should be higher than that used to fit the model.
cdfs	The cut-off points for the cumulative distribution functions.
cafs	The cut-off points for the conditional accuracy functions.

`multipleSubjects` A boolean stating whether the fit is to multiple subjects (`multipleSubjects = TRUE`) or to a single subject (`multipleSubjects = FALSE`).

`seed` The value for the `set.seed` function to set random generation state.

### Details

This function is passed the object obtained by the model fitting procedure, as well as the human data and the condition that was fitted by the routine. The function simulates 100,000 trials (by default) using the best-fitting parameters found by the fit procedure. This synthetic data is then considered as the model's best predictions. The function then provides a plot of the model fit to cumulative distribution functions (CDFs) of correct response time, and conditional accuracy functions (CAFs) to show fit to accuracy data. The function also returns the data used to plot the fit so that the user can use their own plotting methods.

### Value

`cdfs` The CDF values requested by the user.

`cafs` The CAF values requested by the user.

`humanConCDFs` The response time cut-off values for each CDF bin for congruent human data.

`humanInconCDFs` The response time cut-off values for each CDF bin for incongruent human data.

`humanConCAFsRT` The mean response times for each bin of the CAF functions for congruent human data.

`humanInconCAFsRT` The mean response times for each bin of the CAF functions for incongruent human data.

`humanConCAFsError` The percent accuracy for each bin of the CAF functions for congruent human data.

`humanInconCAFsError` The percent accuracy for each bin of the CAF functions for incongruent human data.

`modelConCDFs` The quantile cut-off points for the model predictions for congruent data. A perfect fit would match the `cdfs` asked for by the user (e.g., .1, .3, .5, .7, .9).

`modelInconCDFs` The quantile cut-off points for the model predictions for incongruent data. A perfect fit would match the `cdfs` asked for by the user (e.g., .1, .3, .5, .7, .9).

`modelConCAFs` The percentage accuracy predicted for each CAF bin by the model for congruent data.

`modelInconCAFs` The percentage accuracy predicted for each CAF bin by the model for incongruent data.

### Examples

```
# Assume that the model was just fit to the data contained in
# \code{exampleData} (condition "present") and saved to the variable called
# "fit", then we can obtain a plot of that fit by using the following:
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
fit <- fitSSP(data = exampleData, conditionName = "present", nTrials = 100)
```

```
plot <- plotFitSSP(modelFit = fit, data = exampleData,
                  conditionName = "present", nTrials = 100)
```

---

simulateDSTP

*Obtain simulated response times and accuracy from the DSTP model*


---

### Description

simulateDSTP generates synthetic data from the DSTP model in the form of response time (RT) in seconds and accuracy for both congruent and incongruent trials.

### Usage

```
simulateDSTP(parms, nTrials, var = 0.01, dt = 1/1000, seed = NULL)
```

```
simulate_dstp(parms, nTrials, var = 0.01, dt = 1/1000, seed = NULL)
```

### Arguments

parms	The set of parameters to use to simulate the data. Must be contained in a vector in the order: A, C, driftTarget, driftFlanker, driftStimSelection, driftRS2, ter.
nTrials	How many trials to simulate per congruency condition.
var	The variance of the diffusion process. By default this is set to 0.01.
dt	The diffusion scaling parameter (i.e., time steps). By default, this is set to 0.001.
seed	The value for the set.seed function to set random generation state.

### Details

This function can be employed by the user to generate synthetic data, but its main purpose is to be used by the fitting procedure to generate model predictions for a set of parameter values when trying to find the best-fitting values.

### Value

Returns a data frame with three columns: rt (response time) in seconds, accuracy of the model's response (1 for correct, 0 for error), and congruency condition.

### Examples

```
# declare the parameters
parms <- c(0.070, 0.086, 0.045, 0.065, 0.368, 1.575, 0.225)

# simulate the data. (Note this is a toy example with very low trial numbers
# to speed up the example. For proper use, increase nTrials.)
modelData <- simulateDSTP(parms, nTrials = 100)
```

---

 simulateSSP

*Obtain simulated response times and accuracy from the SSP model*


---

### Description

simulateSSP generates synthetic data from the DSTP model in the form of response time (RT) in seconds and accuracy for both congruent and incongruent trials.

### Usage

```
simulateSSP(parms, nTrials, var = 0.01, dt = 1/1000, seed = NULL)
```

```
simulate_ssp(parms, nTrials, var = 0.01, dt = 1/1000, seed = NULL)
```

### Arguments

parms	The set of parameters to use to simulate the data. Must be contained in a vector in the order: A, ter, p, rd, sda.
nTrials	How many trials to simulate per congruency condition.
var	The variance of the diffusion process. By default this is set to 0.01.
dt	The diffusion scaling parameter (i.e., time steps). By default, this is set to 0.001.
seed	The value for the set.seed function to set random generation state.

### Details

This function can be employed by the user to generate synthetic data, but its main purpose is to be used by the fitting procedure to generate model predictions for a set of parameter values when trying to find the best-fitting values.

### Value

Returns a data frame with three columns: rt (response time) in seconds, accuracy of the model's response (1 for correct, 0 for error), and congruency condition.

### Examples

```
# declare the parameters
parms <- c(0.050, 0.300, 0.400, 0.040, 1.500)

# simulate the data
# (Note this is a toy example with very low trial numbers to speed up the
# example. For proper use, increase nTrials.)
modelData <- simulateSSP(parms, nTrials = 100)
```

# Index

## \* datasets

- exampleData, 5
  
- caf, 2
- cdf, 3
  
- exampleData, 5
  
- fit\_dstp (fitDSTP), 6
- fit\_dstp\_fixed (fitDSTP\_fixed), 8
- fit\_multiple\_dstp (fitMultipleDSTP), 10
- fit\_multiple\_dstp\_fixed (fitMultipleDSTP\_fixed), 12
- fit\_multiple\_ssp (fitMultipleSSP), 14
- fit\_multiple\_ssp\_fixed (fitMultipleSSP\_fixed), 17
- fit\_ssp (fitSSP), 19
- fit\_ssp\_fixed (fitSSP\_fixed), 21
- fitDSTP, 6
- fitDSTP\_fixed, 8
- fitMultipleDSTP, 10
- fitMultipleDSTP\_fixed, 12
- fitMultipleSSP, 14
- fitMultipleSSP\_fixed, 17
- fitSSP, 19
- fitSSP\_fixed, 21
  
- plot\_fit\_dstp (plotFitDSTP), 23
- plot\_fit\_ssp (plotFitSSP), 25
- plotFitDSTP, 23
- plotFitSSP, 25
  
- simulate\_dstp (simulateDSTP), 27
- simulate\_ssp (simulateSSP), 28
- simulateDSTP, 27
- simulateSSP, 28