

# Package ‘beezdiscounting’

May 7, 2026

**Title** Behavioral Economic Easy Discounting

**Version** 0.3.2

**Date** 2025-01-08

**Maintainer** Brent A. Kaplan <bkaplan.ku@gmail.com>

**Description** Facilitates some of the analyses performed in studies of behavioral economic discounting. The package supports scoring of the 27-Item Monetary Choice Questionnaire (see Kaplan et al., 2016; <doi:10.1007/s40614-016-0070-9>), calculating k values (Mazur's simple hyperbolic and exponential) using nonlinear regression, calculating various Area Under the Curve (AUC) measures, plotting regression curves for both fit-to-group and two-stage approaches, checking for unsystematic discounting (Johnson & Bickel, 2008; <doi:10.1037/1064-1297.16.3.264>) and scoring of the minute discounting task (see Koffarnus & Bickel, 2014; <doi:10.1037/a0035973>) using the Qualtrics 5-trial discounting template (see the Qualtrics Minute Discounting User Guide; <doi:10.13140/RG.2.2.26495.79527>), which is also available as a .qsf file in this package.

**License** GPL (>= 2)

**URL** <https://github.com/brentkaplan/beezdiscounting>

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** R (>= 2.10)

**Imports** beezdemand, broom, dplyr, ggplot2, gtools, magrittr, minpack.lm, psych, purrr, stringr, tibble, tidyr

**LazyData** true

**NeedsCompilation** no

**Author** Brent A. Kaplan [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0002-3758-6776>>)

**Repository** CRAN

**Date/Publication** 2025-01-09 00:40:12 UTC

## Contents

ans_dd . . . . .	2
ans_pd . . . . .	3
calc_auc	4
calc_conf_int . . . . .	5
calc_dd . . . . .	5
calc_pd . . . . .	6
calc_r2 . . . . .	7
check_unsystematic . . . . .	7
dd_ip . . . . .	8
fit_dd . . . . .	9
five.fivetrial_dd . . . . .	10
five.fivetrial_pd . . . . .	10
generate_data_mcq . . . . .	11
get_lookup_table . . . . .	11
inn . . . . .	12
long_to_wide_mcq . . . . .	12
long_to_wide_mcq_excel . . . . .	13
mcq27 . . . . .	13
plot.prop_ss_output . . . . .	14
plot.score_mcq27_output . . . . .	15
plot_dd . . . . .	15
prop_ss . . . . .	16
results_dd . . . . .	17
score_dd . . . . .	18
score_mcq27 . . . . .	18
score_one_mcq27 . . . . .	19
score_pd . . . . .	20
summarize_mcq . . . . .	20
timing_dd . . . . .	21
timing_pd . . . . .	21
wide_to_long_mcq . . . . .	22
wide_to_long_mcq_excel . . . . .	22
<b>Index</b>	<b>24</b>

---

ans_dd	<i>Converts answers from 5.5 trial delay discounting from Qualtrics template</i>
--------	--

---

### Description

Converts answers from 5.5 trial delay discounting from Qualtrics template

### Usage

ans\_dd(df)

`ans_pd`

3

### Arguments

`df` A dataframe containing all the columns

### Value

A dataframe with the ResponseId, index, and response (ss or ll).

### Examples

```
ans_dd(five.fivetrial_dd)
```

---

<code>ans_pd</code>	<i>Converts answers from 5.5 trial probability discounting from Qualtrics template</i>
---------------------	--

---

### Description

Converts answers from 5.5 trial probability discounting from Qualtrics template

### Usage

```
ans_pd(df)
```

### Arguments

`df` A dataframe containing all the columns

### Value

A dataframe with the ResponseId, index, and response (sc or lu).

### Examples

```
ans_pd(five.fivetrial_pd)
```

---

calc_auc	<i>Calculate Area-Under-the-Curve (AUC) Metrics for Delay Discounting Data</i>
----------	--

---

### Description

This function calculates three types of Area-Under-the-Curve (AUC) metrics for delay discounting data: regular AUC (using raw delays), log10 AUC (using logarithmically scaled delays), and ordinal AUC (using ordinally scaled delays). These metrics provide different perspectives on the rate of delay discounting.

### Usage

```
calc_auc(dat)
```

### Arguments

dat	A data frame containing delay discounting data. It must include the following columns: <ul style="list-style-type: none"><li>• id: Participant or group identifier.</li><li>• x: Delay values (e.g., in days).</li><li>• y: Indifference point values (e.g., subjective value of the delayed reward).</li></ul>
-----	---

### Value

A tibble with the following columns:

- id: The participant or group identifier.
- auc\_regular: The regular AUC, calculated using the raw delay values.
- auc\_log10: The log10 AUC, calculated using logarithmically transformed delay values.
- auc\_rank: The rank AUC, calculated using ordinally scaled delay values.

### Examples

```
# Example data
data <- data.frame(
  id = rep("P1", 6),
  x = c(1, 7, 30, 90, 180, 365),
  y = c(0.8, 0.5, 0.3, 0.2, 0.1, 0.05)
)

# Calculate AUC metrics for a single participant
calc_auc(data)
```

---

calc_conf_int	<i>Calculate Confidence Intervals for a Parameter</i>
---------------	---

---

**Description**

This function computes the lower and upper bounds of the confidence interval for a parameter estimate, given its standard error, a specified significance level, and the degrees of freedom from the model.

**Usage**

```
calc_conf_int(estimate, std_error, model, alpha = 0.05)
```

**Arguments**

estimate	A numeric value representing the parameter estimate.
std_error	A numeric value representing the standard error of the parameter estimate.
model	A fitted model object that provides the residual degrees of freedom via <code>df.residual()</code> .
alpha	A numeric value representing the significance level. Default is 0.05 (95% confidence interval).

**Value**

A numeric vector of length two:

- First element: Lower bound of the confidence interval.
- Second element: Upper bound of the confidence interval.

**Examples**

```
# Example using a linear model
data <- data.frame(x = 1:10, y = c(2.3, 2.1, 3.7, 4.5, 5.1, 6.8, 7.3, 7.9, 9.2, 10.1))
lm_model <- lm(y ~ x, data = data)
calc_conf_int(estimate = 0.5, std_error = 0.1, model = lm_model, alpha = 0.05)
```

---

calc_dd	<i>Calculate scores, answers, and timing for 5.5 trial delay discounting from Qualtrics template</i>
---------	--

---

**Description**

Calculate scores, answers, and timing for 5.5 trial delay discounting from Qualtrics template

**Usage**

```
calc_dd(df)
```

**Arguments**

df                    A dataframe containing all the columns from the template.

**Value**

A dataframe with k/ed50 values, answers, timing

**Examples**

```
calc_dd(five.fivetrials_dd)
```

---

calc_pd	<i>Calculate scores, answers, and timing for 5.5 trial probability discounting from Qualtrics template</i>
---------	--

---

**Description**

Calculate scores, answers, and timing for 5.5 trial probability discounting from Qualtrics template

**Usage**

```
calc_pd(df)
```

**Arguments**

df                    A dataframe containing all the columns from the template.

**Value**

A dataframe with h/ep50 values, answers, timing

**Examples**

```
calc_pd(five.fivetrials_pd)
```

---

**calc\_r2** *Calculate R-Squared for a Model*

---

**Description**

This function calculates the coefficient of determination ( $R^2$ ) for a given model by comparing the sum of squared errors (SSE) to the total sum of squares (SST).

**Usage**

```
calc_r2(model)
```

**Arguments**

**model** A fitted model object. The model must have `resid()` and `fitted()` methods to extract residuals and fitted values.

**Value**

A numeric value representing the  $R^2$  value of the model. Returns NA if the model is NULL.

**Examples**

```
# Example using a simple linear model
data <- data.frame(x = 1:10, y = c(1, 2, 3, 4, 5, 6, 7, 9, 10, 11))
lm_model <- lm(y ~ x, data = data)
calc_r2(lm_model)
```

---

**check\_unsystematic** *Check for Unsystematic Data Violations*

---

**Description**

This function checks a dataset for violations of two criteria commonly used to identify unsystematic delay-discounting data:

- Criterion 1: Any subsequent value of  $y$  exceeds the previous value by more than a specified proportion of the larger later reward (`ll`).
- Criterion 2: The last value of  $y$  is not at least a specified proportion less than the first value of  $y$ .

**Usage**

```
check_unsystematic(dat, ll = 1, c1 = 0.2, c2 = 0.1)
```

**Arguments**

dat	A data frame containing the delay-discounting data. It must have at least two columns: <ul style="list-style-type: none"> <li>• id: A unique identifier for the data set.</li> <li>• y: The indifference points to be analyzed.</li> </ul>
ll	A numeric value representing the larger later reward. Default is 1.
c1	A numeric value for the threshold proportion for Criterion 1. Default is 0.2.
c2	A numeric value for the threshold proportion for Criterion 2. Default is 0.1.

**Value**

A tibble with the following columns:

- id: The unique identifier for the data set.
- c1\_pass: Logical value indicating whether Criterion 1 was passed.
- c2\_pass: Logical value indicating whether Criterion 2 was passed.

**Examples**

```
data <- tibble::tibble(
  id = c(rep("P1", 6)),
  x = c(1, 7, 30, 90, 180, 365), # delays
  y = c(0.9, 0.5, 0.3, 0.2, 0.1, 0.05) # indifference points
)
check_unsystematic(data, ll = 1, c1 = 0.2, c2 = 0.1)
```

---

dd\_ip

*Delay Discounting Data*


---

**Description**

A dataset containing a set of fake delay discounting responses

**Usage**

```
dd_ip
```

**Format**

A data frame with delay discounting responses

---

fit_dd	<i>Fit Delay-Discounting Model</i>
--------	------------------------------------

---

### Description

This function fits a delay-discounting model to the given dataset using the specified equation and method.

### Usage

```
fit_dd(dat, equation, method)
```

### Arguments

dat	A data frame containing delay (x) and indifference point (y) data. For "two stage" methods, the data must include an id column to identify participants.
equation	A character string specifying the delay-discounting equation to use. Options include: <ul style="list-style-type: none"><li>• "mazur" or "hyperbolic": Hyperbolic delay-discounting model (<math>y = 1/(1 + k \cdot x)</math>).</li><li>• "exponential": Exponential delay-discounting model (<math>y = \exp(-k \cdot x)</math>).</li></ul>
method	A character string specifying the method for fitting the model. Options include: <ul style="list-style-type: none"><li>• "pooled" or "agg": Fits the model using pooled data.</li><li>• "mean": Fits the model using the mean of indifference points at each delay.</li><li>• "ts" or "two stage": Fits the model separately for each participant (requires an id column in dat).</li></ul>

### Value

A list object of class "fit\_dd", containing:

- The fitted model(s).
- The original dataset (dat).
- The specified method (method).

### Examples

```
data <- data.frame(  
  id = rep(1:2, each = 6),  
  x = rep(c(1, 7, 30, 90, 180, 365), 2),  
  y = c(0.9, 0.5, 0.3, 0.2, 0.1, 0.05, 0.85, 0.55, 0.35, 0.15, 0.1, 0.05)  
)  
fit_dd(data, equation = "mazur", method = "two stage")
```

---

five.fivetrial_dd	<i>Example Qualtrics output from the 5.5 trial delay discounting template.</i>
-------------------	--

---

**Description**

An example dataset containing four participants' data (two typical discounting patterns and two patterns suggesting potential misattention to the task).

**Usage**

five.fivetrial\_dd

**Format**

Example Qualtrics output

---

five.fivetrial_pd	<i>Example Qualtrics output from the 5.5 trial probability discounting template.</i>
-------------------	--

---

**Description**

An example dataset containing four participants' data.

**Usage**

five.fivetrial\_pd

**Format**

Example Qualtrics output

---

generate_data_mcq	<i>Generate fake MCQ data</i>
-------------------	-------------------------------

---

**Description**

Generate fake MCQ data

**Usage**

```
generate_data_mcq(n_ids = 100, n_items = 27, seed = 1234, prop_na = 0)
```

**Arguments**

n_ids	Number of subjectids
n_items	Number of trials
seed	Random seed
prop_na	Proportion of NAs in the entire data set

**Value**

Dataframe of subjectid, questionid, and response

**Examples**

```
generate_data_mcq(n_ids = 2, n_items = 27, prop_na = .01)
```

---

get_lookup_table	<i>Get internal lookup table for the 27-item MCQ</i>
------------------	--

---

**Description**

Get internal lookup table for the 27-item MCQ

**Usage**

```
get_lookup_table()
```

**Value**

Dataframe with questionid, magnitude, and kindiff

**Examples**

```
get_lookup_table()
```

---

inn	<i>Calculates item nearest neighbor imputation approach discussed by Yeh et al. (2023)</i>
-----	--

---

**Description**

Calculates item nearest neighbor imputation approach discussed by Yeh et al. (2023)

**Usage**

```
inn(dat, random, verbose)
```

**Arguments**

dat	A single subject's 27-item MCQ data in long form
random	Boolean whether to insert a random draw (0 or 1) for NAs
verbose	Boolean whether to print subject and question ids pertaining to missing data

**Value**

An imputed data set to be scored

---

long_to_wide_mcq	<i>Reshape MCQ data long to wide</i>
------------------	--------------------------------------

---

**Description**

Reshape MCQ data long to wide

**Usage**

```
long_to_wide_mcq(dat, q_col = "questionid", ans_col = "response")
```

**Arguments**

dat	Long format MCQ
q_col	Name of the question column (default is "questionid")
ans_col	Name of the answer column (default is "response")

**Value**

Wide format data frame

---

 long\_to\_wide\_mcq\_excel

*Reshape MCQ data from long to wide (as used in the 21- and 27-Item Monetary Choice Questionnaire Automated Scorer)*

---

### Description

Reshape MCQ data from long to wide (as used in the 21- and 27-Item Monetary Choice Questionnaire Automated Scorer)

### Usage

```
long_to_wide_mcq_excel(dat, subj_col = "subjectid", ans_col = "response")
```

### Arguments

dat	Long format MCQ data
subj_col	Character column name of subject ids
ans_col	Character column name of responses

### Value

Wide format MCQ data that can be used in the Excel Automated Scorers

### Examples

```
long_to_wide_mcq_excel(generate_data_mcq(2))
```

---

 mcq27

*Example 27-item MCQ data*

---

### Description

A dataset containing two participants' data (same data as in the paper by Kaplan et al., 2016)

### Usage

```
mcq27
```

### Format

Long-form data.frame with columns: subjectid, questionid, response.

---

plot.prop\_ss\_output     *Plot Proportion of SIR/SS Choices by k Value*

---

### Description

This function creates a plot of the proportion of SIR/SS choices by k value using the output of the prop\_ss function.

### Usage

```
## S3 method for class 'prop_ss_output'
plot(
  x,
  ...,
  pt_shape = 21,
  pt_fill = "white",
  pt_size = 3,
  title = "Proportion of SIR/SS choices by k value",
  xlab = "k value rank",
  ylab = "Proportion of SS choices"
)
```

### Arguments

x	Output from the prop_ss function
...	Additional arguments passed to ggplot2::geom_point()
pt_shape	Shape of the points in the plot. Default is 21.
pt_fill	Fill color of the points in the plot. Default is "white".
pt_size	Size of the points in the plot. Default is 3.
title	Title of the plot. Default is "Proportion of SIR/SS choices by k value".
xlab	Label for the x-axis. Default is "k value rank".
ylab	Label for the y-axis. Default is "Proportion of SS choices".

### Value

A ggplot object.

### Examples

```
plot(prop_ss(mcq27))
```

---

```
plot.score_mcq27_output
      Plot MCQ-27 Scores
```

---

**Description**

This function creates a plot of the MCQ-27 scores for different metrics (`small_k`, `medium_k`, `large_k`, `geomean_k`, `overall_k`). The function handles different logarithmic transformations of the `k`-values and adjusts the `y`-axis label accordingly.

**Usage**

```
## S3 method for class 'score_mcq27_output'
plot(x, ..., xlab = "Metric", alpha = 0.3)
```

**Arguments**

<code>x</code>	A data frame returned by the <code>score_mcq27</code> function.
<code>...</code>	Additional arguments passed to methods.
<code>xlab</code>	Label for the <code>x</code> -axis. Default is "Metric".
<code>alpha</code>	Transparency of the points in the plot. Default is 0.3.

**Value**

A ggplot object showing the boxplot of MCQ-27 scores.

**Examples**

```
plot(score_mcq27(mcq27))
```

---

```
plot_dd      Plot Delay-Discounting Model
```

---

**Description**

This function generates a plot of the delay-discounting data and the fitted model.

**Usage**

```
plot_dd(
  fit_dd_object,
  xlabel = "Delay",
  ylabel = "Indifference Point",
  title = "",
  logx = TRUE
)
```

**Arguments**

fit_dd_object	A fitted delay-discounting model object of class "fit_dd", created by the fit_dd() function.
xlabel	A character string specifying the label for the x-axis. Default is "Delay".
ylabel	A character string specifying the label for the y-axis. Default is "Indifference Point".
title	A character string specifying the plot title. Default is "".
logx	Logical. If TRUE, the x-axis is log-transformed. Default is TRUE.

**Value**

A ggplot object representing the fitted model and data.

**Examples**

```
data <- data.frame(
  id = rep(1:2, each = 6),
  x = rep(c(1, 7, 30, 90, 180, 365), 2),
  y = c(0.9, 0.5, 0.3, 0.2, 0.1, 0.05, 0.85, 0.55, 0.35, 0.15, 0.1, 0.05)
)
fit <- fit_dd(data, equation = "mazur", method = "mean")
plot_dd(fit)
```

---

prop\_ss

*Calculate proportion of SIR/SS responses at each k value*


---

**Description**

Calculate proportion of SIR/SS responses at each k value

**Usage**

```
prop_ss(dat)
```

**Arguments**

dat	Dataframe (longform) with subjectid, questionid, and response (0 for SIR/SS and 1 for LDR/LL)
-----	---

**Value**

Dataframe with proportion of SIR/SS responses at each k rank

**Examples**

```
prop_ss(mcq27)
```

---

`results_dd`*Extract Results from Delay-Discounting Model*

---

## Description

This function extracts model parameter estimates, fit statistics, and confidence intervals from a fitted delay-discounting model.

## Usage

```
results_dd(fit_dd_object)
```

## Arguments

`fit_dd_object` A fitted delay-discounting model object of class "fit\_dd", created by the `fit_dd()` function.

## Value

A tibble containing the following columns:

- `id`: The participant or group ID (if applicable).
- `term`: The model parameter (e.g., `k`).
- `estimate`: The estimated value of the parameter.
- `std.error`: The standard error of the parameter estimate.
- `statistic`: The t-statistic for the parameter estimate.
- `p.value`: The p-value for the parameter estimate.
- `conf_low`: The lower bound of the 95% confidence interval.
- `conf_high`: The upper bound of the 95% confidence interval.
- `R2`: The coefficient of determination ( $R^2$ ).

## Examples

```
data <- data.frame(  
  id = rep(1:2, each = 6),  
  x = rep(c(1, 7, 30, 90, 180, 365), 2),  
  y = c(0.9, 0.5, 0.3, 0.2, 0.1, 0.05, 0.85, 0.55, 0.35, 0.15, 0.1, 0.05)  
)  
fit <- fit_dd(data, equation = "mazur", method = "two stage")  
results_dd(fit)
```

---

score_dd	<i>Score 5.5 trial delay discounting from Qualtrics template</i>
----------	--

---

**Description**

Score 5.5 trial delay discounting from Qualtrics template

**Usage**

```
score_dd(df)
```

**Arguments**

df                    A dataframe containing all the columns

**Details**

Currently assumes the attending questions are present and labeled "Attend-LL" and "Attend-SS"

**Value**

A dataframe with id, indexes, response, k value, and effective delay 50.

**Examples**

```
score_dd(five.fivetrial_dd)
```

---

score_mcq27	<i>Score 27-item MCQ</i>
-------------	--------------------------

---

**Description**

Score 27-item MCQ

**Usage**

```
score_mcq27(  
  dat = dat,  
  impute_method = "none",  
  round = 6,  
  random = FALSE,  
  trans = "none",  
  return_data = FALSE,  
  verbose = FALSE  
)
```

**Arguments**

dat	Dataframe (longform) with subjectid, questionid, and response (0 for SIR/SS and 1 for LDR/LL)
impute_method	One of: "none", "ggm", "GGM", "inn", "INN"
round	Numeric specifying number of decimal places (passed to base::round())
random	Boolean whether to insert a random draw (0 or 1) for NAs. Default is FALSE
trans	Transformation to apply to k values: "none", "log", or "ln". Default is "none"
return_data	Boolean whether to return the original data and new imputed responses. Default is FALSE.
verbose	Boolean whether to print subject and question ids pertaining to missing data. Default is FALSE.

**Value**

Summary dataframe

**Examples**

```
score_mcq27(mcq27)
```

---

score_one_mcq27	<i>Score one subject's 27-item MCQ</i>
-----------------	--

---

**Description**

Score one subject's 27-item MCQ

**Usage**

```
score_one_mcq27(dat, impute_method = "none", round = 6)
```

**Arguments**

dat	One subject's 27 items from the MCQ
impute_method	One of: "none", "ggm", "GGM", "inn", "INN"
round	Numeric specifying number of decimal places (passed to base::round())

**Value**

Vector with scored 27-item MCQ metrics

**Examples**

```
beezdiscounting:::score_one_mcq27(mcq27[mcq27$subjectid %in% 1, ])
```

---

score_pd	<i>Score 5.5 trial probability discounting from Qualtrics template</i>
----------	--

---

**Description**

Score 5.5 trial probability discounting from Qualtrics template

**Usage**

```
score_pd(df)
```

**Arguments**

df	A dataframe containing all the columns
----	--

**Details**

Currently assumes the attending questions are present and labeled "Attend-LL" and "Attend-SS"

**Value**

A dataframe with id, indexes, response, h value, and effective probability 50.

**Examples**

```
score_pd(five.fivetrial_pd)
```

---

summarize_mcq	<i>Provide a summary of the results from the MCQ ouput table.</i>
---------------	---

---

**Description**

Provide a summary of the results from the MCQ ouput table.

**Usage**

```
summarize_mcq(res, na.rm = TRUE)
```

**Arguments**

res	Dataframe with MCQ results (output from the calc_mcq function)
na.rm	Boolean whether to remove NAs from the calculation

**Value**

Dataframe with summary statistics



**Details**

Currently assumes the attending questions are present and labeled "Attend-LL" and "Attend-SS"

**Value**

A dataframe with ResponseId, indexes, values and timing

**Examples**

```
timing_pd(five.fivetrial_pd)
```

---

wide_to_long_mcq	<i>Reshape MCQ data wide to long</i>
------------------	--------------------------------------

---

**Description**

Reshape MCQ data wide to long

**Usage**

```
wide_to_long_mcq(dat, items = 27)
```

**Arguments**

dat	Wide format MCQ assuming subject id is in column 1
items	Number of MCQ questions

**Value**

Long format data frame

---

wide_to_long_mcq_excel	<i>Reshape MCQ data from wide (as used in the 21- and 27-Item Monetary Choice Questionnaire Automated Scorer) to long</i>
------------------------	---

---

**Description**

Reshape MCQ data from wide (as used in the 21- and 27-Item Monetary Choice Questionnaire Automated Scorer) to long

**Usage**

```
wide_to_long_mcq_excel(dat)
```

**Arguments**

dat                   Wide format MCQ data as used in the Excel Automated Scorers

**Value**

Long format data frame

**Examples**

```
wide_to_long_mcq_excel(long_to_wide_mcq_excel(generate_data_mcq(2)))
```

# Index

## \* datasets

- dd\_ip, 8
- five.fivetrial\_dd, 10
- five.fivetrial\_pd, 10
- mcq27, 13

ans\_dd, 2

ans\_pd, 3

calc\_aucs, 4

calc\_conf\_int, 5

calc\_dd, 5

calc\_pd, 6

calc\_r2, 7

check\_unsystematic, 7

dd\_ip, 8

fit\_dd, 9

five.fivetrial\_dd, 10

five.fivetrial\_pd, 10

generate\_data\_mcq, 11

get\_lookup\_table, 11

inn, 12

long\_to\_wide\_mcq, 12

long\_to\_wide\_mcq\_excel, 13

mcq27, 13

plot.prop\_ss\_output, 14

plot.score\_mcq27\_output, 15

plot\_dd, 15

prop\_ss, 16

results\_dd, 17

score\_dd, 18

score\_mcq27, 18

score\_one\_mcq27, 19

score\_pd, 20

summarize\_mcq, 20

timing\_dd, 21

timing\_pd, 21

wide\_to\_long\_mcq, 22

wide\_to\_long\_mcq\_excel, 22