# Package: flipped (via r-universe)

October 21, 2024

Title Applies various odd models for coin flipping

Version 0.0.1

**Description** Everyone uses the binomial as the distribution for coin flipping: this assumes for a given coin, the probability of landing heads is constant for all time. It is likely a very sound assumption. However, even for this simple example other models may be possible. This package contains such models.

**License** GPL (>= 3)

Imports nloptr, stats

Suggests testthat (>= 3.0.0)

**Config/testthat/edition** 3

Encoding UTF-8

LazyData true

**Roxygen** list(markdown = TRUE)

RoxygenNote 7.1.1

Repository https://phylotastic.r-universe.dev

RemoteUrl https://github.com/bomeara/flipped

RemoteRef HEAD

RemoteSha 70e934a083e24603a2b2868577e299f9f0c91faa

## Contents

pin_exponential_decay	2
pin_from_probability	3
vin_linear	3
coin_multiplicative	5
d_congruent_models	5
_possibilities	6
b_heads_exponential_decay	7
b_heads_linear	7
b_heads_multiplicative	8

profile_exponential_decay_model	 . 8
profile_linear_model	 . 9
try_many_vectors	 . 10
	12

## Index

dcoin\_exponential\_decay

Compute probability of observations given an exponential decay model The idea is that the coin before handling has 100% chance of heads, but each time it is picked up that probability will decrease (maybe it is bent by the statistician's mighty thumb). After halflife times handling it, the probability of heads is 50%, and it keeps dropping from there.

#### Description

Compute probability of observations given an exponential decay model The idea is that the coin before handling has 100% chance of heads, but each time it is picked up that probability will decrease (maybe it is bent by the statistician's mighty thumb). After halflife times handling it, the probability of heads is 50%, and it keeps dropping from there.

#### Usage

```
dcoin_exponential_decay(
   nheads,
   nflips,
   halflife,
   log = FALSE,
   possibilities = get_possibilities(nheads, nflips)
)
```

#### Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)
halflife	How many flips to get to 50% heads
log	If TRUE return log transformed probabilities.
possibilities	All possible sequences of flips that lead to the observed number of heads

## Value

The likelihood of the data (or log likelihood if log=TRUE)

dcoin\_from\_probability

*Compute probability of observations given a vector of probability of heads* 

## Description

Compute probability of observations given a vector of probability of heads

## Usage

```
dcoin_from_probability(
   pheads,
   nheads,
   nflips,
   log = FALSE,
   possibilities = get_possibilities(nheads, nflips),
   diff_value = NULL
)
```

## Arguments

pheads	Vector with the probability of a heads on flip 1, 2, etc.
nheads	Number of heads
nflips	Total number of flips (heads and tails)
log	If TRUE return log transformed probabilities.
possibilities	All possible sequences of flips that lead to the observed number of heads
diff_value	If not NULL, the final likelihood will be abs(likelihood - diff_value) for mini mizing a function

## Value

The likelihood of the data (or log likelihood if log=TRUE)

dcoin_linear	Compute probability of observations given linear change model This
	is essentially stats::dbinom() but allowing for the probability of heads
	to linearly change from the starting value. By default it increases by
	10% per flip, but this can be set to other values.
	to linearly change from the starting value. By default it increases by 10% per flip, but this can be set to other values.

#### Description

The idea is that the coin before handling has some probability of heads, but each time it is picked up that probability could change (maybe it is bent by the statistician's mighty thumb). The slope gives the amount of change in this probability each flip: for example, a coin that starts fair and which has a slope of 0.01 has a probability of heads of 0.51 (0.50 + 0.01) on its first flip, 0.52 on its second, and so forth. If the act of flipping has absolutely no effect on the probability of heads, slope can be set to be zero, though using stats::dbinom() for this particular edge case should be faster.

#### Usage

```
dcoin_linear(
   nheads,
   nflips,
   preflip_prob = 0.5,
   slope = 0.1,
   log = FALSE,
   outside_bounds_is_NA = FALSE
)
```

#### Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)
preflip_prob	Probability of heads before the coin is handled
slope	How much the probability changes each time the coin is flipped
log	If TRUE return log transformed probabilities.
outside_bounds_	is_NA
	If TRUE, if any probability of heads is outside the bounds of probability, the

#### Details

Of course, if all we have is the total number of heads and total number of flips, we do not know if it was HTT, THT, or TTH. For the particular case of a slope set to exactly zero the order does not matter, but in the general case it will. For example, if the probability of heads increases with each flip, HTT is less likely than TTH even though each has one heads out of three flips. The current code looks at all possibilities exhaustively, but more efficient ways to calculate this undoubtedly exist. Pull requests are welcome. It also means this may be slow as the number of flips increases.

function returns NA. Otherwise, it sets the value to the nearer bound.

For some slopes and preflip probabilities, the probabilities of heads on a given flip may be outside the 0 to 1 bounds. By default, if this happens the function returns NA. If outside\_bounds\_is\_NA is FALSE, it moves the probabilities to the nearer bound.

#### Value

The likelihood of the data (or log likelihood if log=TRUE)

4

d\_coin\_multiplicative Compute probability of observations given an exponential decay model

## Description

Compute probability of observations given an exponential decay model

#### Usage

```
d_coin_multiplicative(
    nheads,
    nflips,
    multiplier,
    log = FALSE,
    possibilities = get_possibilities(nheads, nflips),
    outside_bounds_is_NA = FALSE
)
```

#### Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)
multiplier	How much to multiply by each flip
log	If TRUE return log transformed probabilities.
possibilities	All possible sequences of flips that lead to the observed number of heads
outside_bounds_	is_NA
	If TRUE, if any probability of heads is outside the bounds of probability, the function returns NA. Otherwise, it sets the value to the nearer bound.

## Value

The likelihood of the data (or log likelihood if log=TRUE)

find\_congruent\_models Find congruent models to a simple binomial model This will find the parameter values for other models that equal the likelihood for a simple binomial model. This may not be the MLE for these other models

## Description

Find congruent models to a simple binomial model This will find the parameter values for other models that equal the likelihood for a simple binomial model. This may not be the MLE for these other models

## Usage

```
find_congruent_models(
   nheads,
   nflips,
   slopes = c(0, 0.1, -0.05),
   stopval = 1e-04
)
```

## Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)
slopes	Vector of slopes to use
stopval	How large a difference in probability is considered close enough between the flat model and others

## Value

A list containing the parameter estimates with likelihoods for each model and the probabilities for heads at each model

<pre>get_possibilities</pre>	Exhaustively get all possible sets of outcomes that result in a specified
	number of heads out of a certain number of flips

## Description

This grows very large with the number of flips. It will throw an error if you try too many flips.

#### Usage

```
get_possibilities(nheads, nflips)
```

## Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)

## Value

data.frame with each potential trial as a row. 1=heads, 0=tails.

6

prob\_heads\_exponential\_decay

Compute the probability of heads with each flip given an exponential model The model assumes 100% chance of heads before a coin is picked up and it drops exponentially each time the coin is handled.

## Description

Compute the probability of heads with each flip given an exponential model The model assumes 100% chance of heads before a coin is picked up and it drops exponentially each time the coin is handled.

#### Usage

```
prob_heads_exponential_decay(nflips, halflife)
```

#### Arguments

nflips	Total number of flips (heads and tails)
halflife	How many flips to get to 50% heads

#### Value

Vector of probability of heads for the first flip, second flip, etc.

<pre>prob_heads_linear</pre>	Compute the probability of heads with each flip given a linear change
	model.

#### Description

Compute the probability of heads with each flip given a linear change model.

#### Usage

```
prob_heads_linear(nflips, preflip_prob = 0.5, slope = 0.1)
```

#### Arguments

nflips	Total number of flips (heads and tails)
preflip_prob	Probability of heads before the coin is handled
slope	How much the probability changes each time the coin is flipped

## Value

Vector of probability of heads for the first flip, second flip, etc.

#### prob\_heads\_multiplicative

Compute the probability of heads with each flip given a multiplier model The model assumes 50% chance of heads before a coin is picked up and it changes as a percentage of the previous value each flip. i.e., the probability of heads is 101% of the probability the previous flip with a multiplier of 1.01.

#### Description

Compute the probability of heads with each flip given a multiplier model The model assumes 50% chance of heads before a coin is picked up and it changes as a percentage of the previous value each flip. i.e., the probability of heads is 101% of the probability the previous flip with a multiplier of 1.01.

#### Usage

```
prob_heads_multiplicative(nflips, multiplier, outside_bounds_is_NA = FALSE)
```

#### Arguments

nflips	Total number of flips (heads and tails)	
multiplier	Factor to multiply the previous probability by	
outside_bounds_is_NA		
	If TRUE, if any probability of heads is outside the bounds of probability, the	
	function returns NA. Otherwise, it sets the value to the nearer bound.	

#### Value

Vector of probability of heads for the first flip, second flip, etc.

## Description

Computes the likelihood for a range of values using an exponential coin model

## profile\_linear\_model

#### Usage

```
profile_exponential_decay_model(
   nheads,
   nflips,
   param_range = c(0, nflips * 10),
   number_of_steps = 1000,
   log = FALSE
)
```

#### Arguments

nflips	Total number of flips (heads and tails)
param_range	Range of parameters to try
number_of_steps	
	How many values of the parameter to try

## Value

vector of likelihoods

#### Examples

```
nheads <- 8
nflips <- 10
exp_results <- profile_exponential_decay_model(nheads, nflips)
plot(x=exp_results$preflip_prob, y=exp_results$likelihood, type="1")
best_param <- exp_results$halflife[which.max(exp_results$likelihood, na.rm=TRUE)]
print(best_param)</pre>
```

profile\_linear\_model Computes the likelihood for a range of values using a linear coin model

## Description

Computes the likelihood for a range of values using a linear coin model

#### Usage

```
profile_linear_model(
   nheads,
   nflips,
   param_range = c(0, 1),
   slope = 0.1,
   number_of_steps = 1000,
   log = FALSE,
   outside_bounds_is_NA = FALSE
)
```

#### Arguments

nflips	Total number of flips (heads and tails)	
param_range	Range of parameters to try	
slope	How much the probability changes each time the coin is flipped	
number_of_steps		
	How many values of the parameter to try	

#### Value

vector of likelihoods

## Examples

```
nheads <- 8
nflips <- 10
linear_results <- profile_linear_model(nheads, nflips)
plot(x=linear_results$preflip_prob, y=linear_results$likelihood, type="1")
dbinom_proportions <- seq(from=0, to=1, length.out=1000)
lines(dbinom_proportions, dbinom(nheads, nflips, dbinom_proportions), col="red")
best_param <- linear_results$preflip_prob[which.max(linear_results$likelihood, na.rm=TRUE)]
print(best_param)</pre>
```

try_many_vectors	Compute probability of observations across many potential vectors
	This will try (1/stepsize)^nflips possible vectors, computing the proba-
	bility of the observation for each

## Description

Compute probability of observations across many potential vectors This will try (1/stepsize)^nflips possible vectors, computing the probability of the observation for each

#### Usage

```
try_many_vectors(
   nheads,
   nflips,
   number_samples = 1000,
   stopval = 1e-05,
   log = FALSE,
   possibilities = get_possibilities(nheads, nflips)
)
```

## Arguments

nheads	Number of heads
nflips	Total number of flips (heads and tails)
number_samples	How many vectors to sample
stopval	How large a difference in probability is considered close enough between the flat model and others
log	If TRUE return log transformed probabilities.
possibilities	All possible sequences of flips that lead to the observed number of heads

## Value

The likelihood of the data (or log likelihood if log=TRUE)

# Index

d\_coin\_multiplicative, 5
dcoin\_exponential\_decay, 2
dcoin\_from\_probability, 3
dcoin\_linear, 3

 $\texttt{find\_congruent\_models, 5}$ 

get\_possibilities, 6

prob\_heads\_exponential\_decay, 7
prob\_heads\_linear, 7
prob\_heads\_multiplicative, 8
profile\_exponential\_decay\_model, 8
profile\_linear\_model, 9

 $try_many_vectors, 10$