Package 'truthiness'

May 24, 2021

Title Illusory Truth Longitudinal Study

Version 1.2.5

Date 2021-05-23

URL https://github.com/dalejbarr/truthiness

Description Data and functions for analyzing and simulating illusory truth datasets, developed as part of a longitudinal study by Henderson, Barr, and Simons (2020). The illusory truth effect is the observation that people rate repeated statements as more likely to be true than novel statements. We tested the trajectory of the illusory truth effect by collecting truth ratings for statements repeated across four time intervals: immediately, one day, one week, and one month following initial presentation. The package contains the anonymized data from the study along with stimulus materials, as well as functions for analyzing the data, running simulations, and calculating power. Further details about the project are available at <https://osf.io/nvugt/>, which includes Stage 1 of the Registered Report at the Journal of Cognition (<https://osf.io/vqnx2/>).

License CC BY 4.0

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports ordinal, magrittr, dplyr, MASS, tibble, tidyr, stats, lme4, readr, purrr, rmarkdown, emmeans, DT, Rdpack, ggplot2, forcats, ez

RdMacros Rdpack

RoxygenNote 7.1.1

Suggests testthat (>= 2.1.0)

NeedsCompilation no

Author Dale Barr [aut, cre], Emma Henderson [ctb]

Maintainer Dale Barr <dalejbarr@protonmail.com>

Repository CRAN

Date/Publication 2021-05-24 08:00:03 UTC

R topics documented:

`allsimp.emmc` *Custom Contrast Function for emmeans*

Description

Create a contrast matrix for equivalence test of a 2x4 interaction.

Usage

```
allsimp.emmc(levels, ...)
```

Arguments

`levels`	Interaction levels (should be of length 8).
`...`	Any other arguments (NB: currently ignored).

Details

Runs all six ways of comparing the simple effects of the two-level factor. For use with the emmeans package.

Value

A data frame to be passed to the `specs` argument of `emmeans`, each column of which represents predictor codings that contrast the illusory truth effect across two intervals.

Examples

```
library(ordinal)
library(emmeans)

## create data frame with predictor codings
moddata <- get_model_data()

## use 'allsimp' with emmeans for equivalence test
mod_emm <- emmeans(truth_trajectory_models[["ix2"]],
                   allsimp ~ Rep * Int, data = moddata)

mod_emm
```

alpha_6_to_7 *Convert Threshold Values from Six to Seven Point Scale*

Description

Convert Threshold Values from Six to Seven Point Scale

Usage

```
alpha_6_to_7(thresh = truthiness::clmm_maximal$alpha)
```

Arguments

thresh	Log-odds thresholds from a cumulative logit model fit. Should be a five-element vector.

Details

The basic algorithm is to copy the top and bottom thresholds and then to shrink the spaces between the thresholds on the six-point scale to 80% of their size. The left over space is allocated to the middle category on the seven-point scale.

Value

A six-element vector representing the thresholds on a seven-point scale.

Examples

```
clmm_maximal$alpha # original thresholds
alpha_6_to_7(clmm_maximal$alpha)
```

check_fake *Is Data Flagged As Simulated?*

Description

Check whether the data in the subdirectory is flagged as simulated.

Usage

```
check_fake(path)
```

Arguments

path	Name of the subdirectory.

Value

TRUE if the data in the subdirectory is tagged as simulated, FALSE otherwise.

---

clmm_maximal *Fitted Cumulative Link Mixed Model for Nadarevic and Erdfelder Data*

Description

Fitted Cumulative Link Mixed Model for Nadarevic and Erdfelder Data

Usage

```
clmm_maximal
```

Format

An object of class "clmm", resulting from a call to the clmm function.

Details

The object is the result of the function call

`ordinal::clmm(trating ~ R * D + (R * D | subj_id) + (R * D | item_id),NE_exp1).`

The fitted model is stored as an independent object in the package because the fitting process is too slow to allow it to be re-created whenever it is needed.

See Also

NE_exp1

---

codebook *Compile and Display Codebook and Materials*

Description

Compile and display the codebook for anonymized data and stimulus materials.

Usage

```
codebook(show_stim = TRUE, browse = TRUE)
```

Arguments

show_stim	Whether to include the stimulus materials.
browse	Whether to open the codebook in a browser. Otherwise, it just prints the filename.

Value

Path to the codebook file.

`derive_fixed` *Derive Fixed-Effects Parameters from Phase-by-Phase Effects*

Description

Derive Fixed-Effects Parameters from Phase-by-Phase Effects

Usage

```
derive_fixed(phase_eff)
```

Arguments

`phase_eff`	A four-element vector specifying the illusory truth effect at each of the four testing phases (in log odds units).

Value

Vector with eight elements containing the fixed effects coefficients for deviation-coded predictors.

`equivtest` *Run Equivalence Tests on Existing CLMM Object*

Description

Run Equivalence Tests on Existing CLMM Object

Usage

```
equivtest(mod, .data, main_effect = FALSE, delta = 0.14)
```

Arguments

`mod`	Fitted model object, result of call to `clmm`.
`.data`	Data frame containing source data.
`main_effect`	Whether to perform the test for the main effect (TRUE) or interaction (FALSE).
`delta`	Delta (SESOI) for the equivalence test, in raw log odds units.

Value

A vector with p-values from the equivalence test(s); elements named `simple` test simple effects, while elements named `equiv` contain the corresponding equivalence test results.

Examples

```
moddata <- get_model_data()

equivtest(truth_trajectory_models[["main2"]], moddata,
          main_effect = TRUE)
```

`eta2resp` *Simulate Ordinal Response Choices from Log Odds*

Description

Simulate Ordinal Response Choices from Log Odds

Usage

```
eta2resp(eta, thresh)
```

Arguments

`eta`	Predicted response tendency or tendencies on log odds scale.
`thresh`	Cut-points (thresholds).

Value

A vector of the same length as `eta` with simulated integer response values, one for each `eta` value.

Examples

```
# N=10 with eta = 0 and 6 point scale from N&E
eta2resp(rep(0, 10), clmm_maximal$alpha)

# N=10 with eta = 0 and 7 point scale
eta2resp(rep(0, 10), alpha_6_to_7(clmm_maximal$alpha))

# N=10 with eta = 4 and 6 point scale from N&E
eta2resp(rep(4, 10), clmm_maximal$alpha)

# N=10 with eta = 4 and 7 point scale
eta2resp(rep(4, 10), alpha_6_to_7(clmm_maximal$alpha))
```

`fit_clmm` *Fit Cumulative Link Mixed-Effects Model to Simulated Ratings*

Description

Fit Cumulative Link Mixed-Effects Model to Simulated Ratings

Usage

```
fit_clmm(.data, main_effect = FALSE)
```

Arguments

`.data`	Data frame, with the format as resulting from a call to `gen_data`.
`main_effect`	Whether to test the main effect of repetition (TRUE) or the repetition-by-interval interaction (FALSE; the default).

Details

Fits a cumulative link mixed-effects model to the data and tests the specified effect (interaction or main effect) using a likelihood-ratio test using `ordinal::clmm()`. The function's main purpose is to be used in power simulation.

If the interaction is to be tested, the following two models are compared:

`trating ~ R * (I1 + I2 + I3) + (1 + R:I1 + R:I2 + R:I3 | subj_id) + (1 + R:I1 + R:I2 + R:I3 | stim_id)`

`trating ~ R + I1 + I2 + I3 + (1 + R:I1 + R:I2 + R:I3 | subj_id) + (1 + R:I1 + R:I2 + R:I3 | stim_id)`.

If the main effect is to be tested, then the following two models are compared.

`trating ~ R * (I1 + I2 + I3) + (1 + R | subj_id) + (1 + R | stim_id)`

`trating ~ I1 + I2 + I3 + R:I1 + R:I2 + R:I3) + (1 + R | subj_id) + (1 + R | stim_id)`.

Value

A vector, with the following elements.

`R` Fixed-effects estimate of the main effect of repetition.

`I1` Fixed-effects estimate of the main effect of interval (1).

`I2` Fixed-effects estimate of the main effect of interval (2).

`I3` Fixed-effects estimate of the main effect of interval (3).

`R:I1` Fixed-effects estimate of the interaction (1).

`R:I2` Fixed-effects estimate of the interaction (2).

`R:I3` Fixed-effects estimate of the interaction (3).

dev1 Deviance for the model including the effect(s) of interest.

dev2 Deviance for the model excluding the effect(s) of interest.

chisq_RI Chi-square value for the likelihood ratio test.

p_RI Associated p-value.

thresh.1|2 First cut-point (threshold).

thresh.2|3 Second cut-point.

thresh.3|4 Third cut-point.

thresh.4|5 Fourth cut-point.

thresh.5|6 Fifth cut-point.

thresh.6|7 Sixth cut-point.

See Also

gen_data, power_sim.

Examples

```
set.seed(62)
dat <- gen_data(24) # test main effect

fit_clmm(dat, TRUE) # takes a few minutes
```

`fit_lmem` *Fit Linear Mixed-Effects Model to Simulated Ratings*

Description

Fit a linear mixed-effects model (LMM) to simulated ratings data.

Usage

```
fit_lmem(.data, main_effect = FALSE)
```

Arguments

`.data`	Data frame, with the format as resulting from a call to gen_data.
`main_effect`	Whether to test the main effect of repetition (TRUE) or the repetition-by-interval interaction (FALSE; the default).

Details

This function is used to estimate parameters for power analysis with simulated data. `fit_lmem` fits a linear-mixed effects model to the data with `lmer` and tests the specified effect (interaction or main effect) using a likelihood-ratio test. If the interaction is to be tested, the following two models are compared.

`trating ~ R * (I1 + I2 + I3) + (1 + R:I1 + R:I2 + R:I3 || subj_id) + (1 + R:I1 + R:I2 + R:I3 || stim_id)`

`trating ~ R + I1 + I2 + I3 + (1 + R:I1 + R:I2 + R:I3 || subj_id) + (1 + R:I1 + R:I2 + R:I3 || stim_id)`.

If the main effect is to be tested, then the following two models are compared.

`trating ~ R * (I1 + I2 + I3) + (1 + R || subj_id) + (1 + R || stim_id)`

`trating ~ I1 + I2 + I3 + R:I1 + R:I2 + R:I3) + (1 + R || subj_id) + (1 + R || stim_id)`.

Value

A vector, with the following elements.

`(Intercept)` Fixed-effects estimate of the intercept.

`R` Fixed-effects estimate of the main effect of repetition.

`I1` Fixed-effects estimate of the main effect of interval (1).

`I2` Fixed-effects estimate of the main effect of interval (2).

`I3` Fixed-effects estimate of the main effect of interval (3).

`R:I1` Fixed-effects estimate of the interaction (1).

`R:I2` Fixed-effects estimate of the interaction (2).

`R:I3` Fixed-effects estimate of the interaction (3).

dev1 Deviance for the model including the effect(s) of interest.

dev2 Deviance for the model excluding the effect(s) of interest.

chisq_RI Chi-square value for the likelihood ratio test.

p_RI Associated p-value.

m1_singular Whether the covariance matrix for model 1 was singular.

m2_singular Whether the covariance matrix for model 2 was singular.

m1_conv Whether model 1 converged.

m2_conv Whether model 2 converged.

See Also

`gen_data`, `power_sim`.

Examples

```
set.seed(62)
dat <- gen_data(40)
fit_lmem(dat, TRUE) # test main effect
```

flag_fake *Flag Subdirectory as Having Simulated Data*

Description

Flag Subdirectory as Having Simulated Data

Usage

```
flag_fake(path)
```

Arguments

path	Path to subdirectory.

Details

This function tags data in a subdirectory as simulated so that it is not confused with genuine data. When an analysis report is compiled against data from that subdirectory, the report will contain a warning that the data is not real.

Value

No return value, called only for its side effect.

gen_data *Simulate Truth Rating Data*

Description

Simulate Truth Rating Data

Usage

```
gen_data(
  nsubj,
  phase_eff = rep(0, 4),
  thresh = alpha_6_to_7(truthiness::clmm_maximal$alpha),
  subj_rfx = ordinal::VarCorr(truthiness::clmm_maximal)$subj_id,
  item_rfx = ordinal::VarCorr(truthiness::clmm_maximal)$item_id,
  dropout = c(0.05, 0.1, 0.1)
)
```

Arguments

`nsubj`	Number of subjects. Because of counterbalancing, must be a multiple of 8.
`phase_eff`	A four-element vector giving the size of the illusory truth effect at each of the four phases (on the log odds scale). Use `rep(0,4)` for testing Type I error rate. A value of .14 gives an effect of approximately 1/10 of a scale point.
`thresh`	Cut-points (thresholds) for the seven point scale (must be a six-element vector).
`subj_rfx`	A 4x4 covariance matrix with by-subject variance components for the intercept, main effect of repetition, main effect of interval, and repetition-by-interval interaction. Only the variances (elements on the diagonal) are used in the simulation (see Details).
`item_rfx`	A 4x4 covariance matrix with by-statement variance components for the intercept, main effect of repetition, main effect of interval, and repetition-by-interval interaction. Only the variances (elements on the diagonal) are used in the simulation (see Details).
`dropout`	A vector encoding assumptions about the proportion of subjects dropping out of the study over the four testing intervals (immediate, 1 day, 1 week, 1 month). The first element represents the proportion of subjects who completed the first phase (immediate) but who drop out before the next interval one day later. The second element represents the proportion of the remaining participants dropping out after 1 day and before 1 week. The third and final element represents the proportion of remaining participants dropping out after 1 week and before 1 month. For example, the default values of `c(.05,.1,.1)` encode dropout rates of 5%, 10%, and 10%.

Details

By default, the thresholds and parameter estimates for variance components used in the simulation are from the cumulative link mixed model fit to the Nadarevic and Erdfelder data. Only the variances from the by-subject and by-item covariance matrices are used. Unlike Nadarevic and Erdfelder, who only had two testing intervals, the simulated study assumes four intervals, coded by three predictors for the main effect and three for the interaction with repetition. The code below depicts how the four-element variance vector from the original study is translated into the eight variances needed for the simulated data.

```
newvar_subj <-rep(diag(subj_rfx),c(1,1,3,3))
```

```
newvar_item <-rep(diag(item_rfx),c(1,1,3,3))
```

The simulated data includes ratings for 128 stimulus items for each subject. Half of the statements are repeated (old) and half are new. A quarter of the items (32) are tested at each phase.

It is assumed that the key effect present in the data is the interaction term, which is designed to represent an illusory-truth effect that first appears at the second testing interval (1 day) and remains over the subsequent two intervals without changing size. All other fixed effects in the model (main effect of R and three effects encoding the main effect of interval) are driven by the interaction term.

Value

A data frame, with `nsubj * 128` rows and 11 variables, where:

`subj_id` Unique subject identifier.

`list_id` Which set of statements the subject received.

`stim_id` Unique stimulus (statement) identifier.

`repetition` Whether the statement was old or new.

`interval` Testing interval (immediate, 1 day, 1 week, 1 month).

`eta` The simulated response tendency, on the log odds scale.

`trating` The simulated rating value.

`R` Deviation-coded predictor for repetition (old = 1/2, new = -1/2).

`I1` Deviation-coded predictor for interval comparing baseline (immediate) to 1 day.

`I2` Deviation-coded predictor for interval comparing baseline (immediate) to 1 week.

`I3` Deviation-coded predictor for interval comparing baseline (immediate) to 1 month.

See Also

`clmm_maximal`, `NE_exp1`

Examples

```
# demonstrate how to convert from four variances to eight
four_var <- diag(ordinal::VarCorr(clmm_maximal)$subj_id)
four_var
rep(four_var, c(1, 1, 3, 3))

# basic usage
dat <- gen_data(256)

# demonstrate deviation coding
dat %>% dplyr::distinct(repetition, interval, R, I1, I2, I3)

# demonstrate dropouts
dat %>% dplyr::distinct(subj_id, interval) %>% dplyr::count(interval)
```

`get_model_data` *Get Ratings Data with Model Predictors*

Description

Apply participant/phase-level exclusions and then add numeric and factor predictors to the ratings data.

Usage

```
get_model_data()
```

Value

A data frame, with columns:

subj_id Unique subject identifier.

stim_id Unique stimulus identifier.

repetition Whether the statement was repeated or new.

interval Presentation interval.

R Deviation-coded numerical predictor for repetition.

I1 Deviation-coded numerical predictor for interval (1 day vs. immediate).

I2 Deviation-coded numerical predictor for interval (1 week vs. immediate).

I3 Deviation-coded numerical predictor for interval (1 month vs. immediate).

Rep Deviation-coded factor for repetition.

Int Deviation-coded factor for interval.

See Also

`truth_trajectory_data`

Examples

```
get_model_data()
```

---

locate_data_files *Locate Raw Data Files from Longitudinal Illusory Truth Study*

Description

Look in a subdirectory and find files containing the raw data.

Usage

```
locate_data_files(path, full.names = TRUE)
```

Arguments

`path`	Path to data files.
`full.names`	If 'TRUE', the directory path is prepended to the file names to give a relative file path. If 'FALSE', the file names (rather than paths) are returned.

Details

Looks for files matching the regular expression `^[Pp][1-4]\.[Cc][Ss][Vv]$` and performs basic error-checking.

Value

A character vector with the paths to the files.

NE_exp1 *Data from Experiment 1 of Nadarevic and Erdfelder*

Description

A dataset containing truth ratings from Experiment 1 of Nadarevic and Erdfelder (2014).

Usage

```
NE_exp1
```

Format

A data frame with 14,950 observations on 7 variables:

`subj_id` Unique subject identifier.

`item_id` Unique stimulus (statement) identifier.

`repetition` Whether the statement was repeated (`old`) or not (`new`).

`delay` Testing interval, ten minutes (`10m`) or one week (`1w`) after initial exposure.

`trating` Truth rating on a six-point scale, with higher values corresponding to greater perceived truth.

`R` Deviation-coded numerical predictor for `repetition`, with `old` = .5 and `new` = -.5.

`D` Deviation-coded numerical predictor for `delay`, with `10m` = .5 and `1w` = -.5.

Source

The source data is from Nadarevic and Erdfelder (2014), which is freely available for download from https://osf.io/eut35/. The data included here has been reorganized for statistical modeling.

References

Nadarevic L, Erdfelder E (2014). "Initial judgment task and delay of the final validity-rating task moderate the truth effect." *Consciousness and Cognition*, **23**, 74–84.

See Also

`link{NE_items}`

NE_items *Stimulus Information from Experiment 1 of Nadarevic and Erdfelder*

Description

A dataset describing the statements used as stimuli in Experiment 1 of Nadarevic and Erdfelder (2014).

Usage

```
NE_items
```

Format

A data frame with 176 rows and 4 variables:

`item_id` Unique stimulus (statement) identifier.

`statement` Statement (in German).

`set` Which set the statement belong to, used for counterbalancing.

`status` Actual truth of the statement.

References

Nadarevic L, Erdfelder E (2014). "Initial judgment task and delay of the final validity-rating task moderate the truth effect." *Consciousness and Cognition*, **23**, 74–84.

normalize_path *Get Rid of Trailing Slash*

Description

Remove extra trailing slash from file path.

Usage

```
normalize_path(path)
```

Arguments

`path`	Directory name.

Value

Directory name without a trailing slash.

power_equiv *Power Simulation For Equivalence Test*

Description

Power Simulation For Equivalence Test

Usage

```
power_equiv(phase_eff, delta, target_effect, nsubj, nruns, outfile = ".AUTO.")
```

Arguments

phase_eff	A four-element vector, each element of which specifies the illusory truth effect at the corresponding phase, on the log odds scale (see gen_data).
delta	Smallest (raw) effect size of interest, on log odds scale; NULL to store fitted model object.
target_effect	Which effect to test, the main effect ('main') or the interaction effect ('interaction').
nsubj	Number of subjects.
nruns	How many simulations to run.
outfile	One of three options: (1) file name to save the results in (with extension .rds); (2) ".AUTO." to create a descriptive filename automatically; or (3) NULL to return the results of the simulation.

Value

Either the name of the file where results are saved or a matrix containing results of fit_lmem or fit_clmm).

Examples

```
set.seed(62)


## takes a few minutes to complete
power_equiv(c(0, .14, .14, .14), .1, "main", 24, 1, NULL)
```

power_sim *Run Power Simulations*

Description

Run Power Simulations

Usage

```
power_sim(
  model,
  phase_eff,
  target_effect,
  nsubj,
  nruns,
  outfile = sprintf("%s_%s_%s_%04d_128_%05d_%s_%d.rds", model,
    sprintf("%0.2f~%0.2f~%0.2f~%0.2f", phase_eff[1], phase_eff[2], phase_eff[3],
   phase_eff[4]), target_effect, nsubj, nruns, Sys.info()[["nodename"]], Sys.getpid())
)
```

Arguments

model	Which type of model to fit: use 'lmem' for linear mixed-effects model and 'clmm' for cumulative link mixed-effects model.
phase_eff	A four-element vector, each element of which specifies the illusory truth effect at the corresponding phase, on the log odds scale (see gen_data).
target_effect	Which effect to test, the main effect ('main') or the interaction effect ('interaction').
nsubj	Number of subjects.
nruns	How many simulations to run.
outfile	Name of output file; NULL to return the simulation results.

Value

Either the name of the outfile (if outfile is non-null) or the results of the simulation (a matrix containing results of fit_lmem or fit_clmm).

Examples

```
set.seed(62)
power_sim("lmem", c(0, .14, .14, .14), "main", 40, 1, NULL)
```

preprocess *Importing and Preprocessing Longitudinal Illusory Truth Data*

Description

Functions to import and preprocess raw (or simulated) data.

Usage

```
preprocess(path, outpath = NULL, report = NULL)

preprocess_simulated(path, outpath = NULL, report = NULL)

import_sessions(path)

import_sessions_simulated(path)

import_phase_info(path)

import_phase_info_simulated(path)

import_cjudgments_simulated(path)

import_cjudgments(path)

import_tratings(path)

import_tratings_simulated(path)

read_sessions(path)

read_sessions_simulated(path)

read_cjudgments(path)

read_cjudgments_simulated(path)

read_tratings(path)

read_tratings_simulated(path)
```

Arguments

path	Path to the directory containing raw data files.
outpath	Path to the directory where anonymized data will be saved.
report	Filename of the HTML preprocessing report.

Details

The purpose of these functions are to import, transform, and anonymize raw data files from the Truth Trajectory study by Henderson et al. (2020). As few users other than the researchers will have access to the original non-anonymized data, functions are also supplied to perform the same set of actions on simulated data. There are two versions of each function, an original version (e.g., `preprocess`) and a simulated version (e.g., `preprocess_simulated`). We include two sets of functions because the simulated functions were built during the planning stage of the study, based on assumptions about the structure of the raw data files that turned out to be incorrect once we obtained pilot data. Rather than laboriously re-write the simulation functions to match the new data structure, we decided to preserve the old functions and split them off from the new versions. They perform the same set of actions and yield the same end products, but import and transform the data differently because of the differing nature of the raw data files.

The "preprocessing" functions are the high-level functions and the only ones that most users will need. The "import" and "read" are lower-level functions that are called by the "preprocess" functions, and are described here for completeness.

Value

A string with the path to the generated HTML report.

Preprocessing

Generally, users will not have access to the non-anonymized raw data and so will not need to use any of these functions, except when working with simulated data. The data objects resulting from the preprocessing of the original raw data are available as built-in data objects documented in `truth_trajectory_data`. Users interested in reproducing the results from the anonymized data should start with the documentation for `reproduce_analysis`.

The `preprocess` functions load in the data from the raw data files and write out (1) non-anonymized, preprocessed data files; (2) anonymized, preprocessed data files; and (3) an HTML report. It performs these actions by running scripts derived from R Markdown templates included in the package. It is not necessary to view these scripts, but if you wish to do so, use `draft`; R Studio users can also access the templates from the "New File > R Markdown" pull down menu and then selecting the appropriate template in the dialog box.

To access this preprocessing script for simulated data:

```
rmarkdown::draft("preprocessing-simulated.Rmd","illusory-truth-preprocessing-sim","truthiness")
```

and the preprocessing script for real data:

```
rmarkdown::draft("preprocessing.Rmd","illusory-truth-preprocessing","truthiness")
```

The processing script outputs four anonymized data files into the subdirectory named in the `outpath` argument. For maximum portability, each file is stored in two versions: binary (RDS) format as well as comma-separated values (CSV). These files are called `ANON_sessions`, `ANON_phases`, `ANON_categories`, and `ANON_ratings` and the data they contain is described in the `codebook`.

In addition to the anonymized data, the preprocessing scripts output two files with non-anonymized data. These files contain sensitive information (Prolific IDs and answers to open-ended questions) and are named `NOT_ANONYMIZED_sessions.rds` and `NOT_ANONYMIZED_phases.rds`. They are written to the "target directory", which is the directory just above the subdirectory with the anonymized data as specified by `outpath`; if `outpath` is `NULL`, then a subdirectory is created in the

working directory for the anonymized files and the target directory will be the working directory. The compiled HTML report is also stored in the target directory. If the filename is not specified by the user (NULL), then one is generated, with a prefix corresponding to the name of the subdirectory where the anonymized data is stored, and the suffix "-preprocessing.html". The return value of the preprocessing function is the file path to this report.

Users can manually add exclusions by editing the files manually_exclude_participants.csv and manually_exclude_phases.csv in the target directory; if they don't exist, then they will be written to the target directory when the script is first run. Thus, it is wise to run the preprocessing script twice: once to create the files so that the user can see how the entries in these files should be structured, and once again after filling in the data to apply the manual exclusions.

Import and Read Functions

The import_* and read_* functions are not intended to be called directly; instead, the user will typically call the preprocess or preprocess_simulated function, or render the R Markdown preprocessing template (using draft). These lower-level functions are invoked by these higher-level functions, and are documented here for completeness.

The import_* functions extract session, phase, category judgments, or ratings data from the full set of raw data files in subdirectory path and return a (non-anonymized) data frame with the corresponding data. They do this by calling the corresponding read_* function for each of the single input files in the subdirectory, and transforming and combining the information as required.

References

Henderson EL, Simons DJ, Barr DJ (2020). "The Trajectory of Truth: A Longitudinal Study of the Illusory Truth Effect (Stage 1 Registered Report)." *Journal of Cognition*. https://osf.io/vqnx2/.

Examples

```
td_raw <- tempfile()  # temp dir for raw data
td_anon <- tempfile() # temp dir for preprocessed data

## simulate data and preprocess it

set.seed(62)
simulate_resp_files(40, path = td_raw, overwrite = TRUE)


## run the built-in R Markdown script
tf1 <- tempfile(fileext = ".html") # temporary file for report
report <- preprocess_simulated(td_raw, td_anon, tf1)

browseURL(report) # view the HTML preprocessing report

file.remove(report) # clean up


sess <- import_sessions_simulated(td_raw)
sess_p1 <- read_sessions_simulated(file.path(td_raw, "P1L1.csv"))
```

```
# clean up temp files
unlink(td_raw, TRUE, TRUE)
unlink(td_anon, TRUE, TRUE)
```

`reproduce_analysis` *Reproduce the Analysis for Longitudinal Illusory Truth Study*

Description

Re-run the analysis for the Henderson et al. (2020) longitudinal truth study.

Usage

```
reproduce_analysis(
  outfile = "analysis.html",
  refit = FALSE,
  savefig = FALSE,
  recipe = FALSE,
  parallel = TRUE,
  infile = NULL
)
```

Arguments

`outfile`	Path to the HTML output file.
`refit`	Whether to re-fit the cumulative link mixed model `TRUE` or to use the built-in model fits (`FALSE`). Due to the extremely time-consuming nature of model estimation, the default is set to `FALSE`.
`savefig`	Whether to save the two plots as separate PNG files (`means_plot.png` and `validation_plot.png`).
`recipe`	Include instructions on how to reproduce the analysis.
`parallel`	Whether to fit models using a single CPU processing core (`FALSE`) or multiple cores (`TRUE`, the default). If `refit` is `FALSE`, this parameter is ignored.
`infile`	Path to the R Markdown script; `NULL` to use the built-in script.

Details

Runs R Markdown script containing the analysis code. The analysis is performed on the built-in preprocessed anonymized data (documented in `truth_trajectory_data`). The script output is rendered as an HTML report, specified by `outfile`. Although it is not necessary to do so, the master R Markdown script for processing real data can be accessed using

```
rmarkdown::draft("analysis.Rmd","illusory-truth-analysis","truthiness")
```

Value

A string with the path to the generated HTML report.

Path to the rendered HTML report.

See Also

reproduce_analysis_sim

Examples

```
tf <- tempfile(fileext = ".html")

## Run the built-in R Markdown script without refitting models.
## To re-fit the models, set refit = TRUE
## (NB: refitting can take ~ 24 hours)
reproduce_analysis(tf)

browseURL(tf)

## clean up
if (file.exists(tf)) file.remove(tf)
```

---

reproduce_analysis_sim

Simulate the Analysis for Longitudinal Illusory Truth Study

Description

Runs the main analysis for Henderson et al. (2020) on simulated data.

Usage

```
reproduce_analysis_sim(
  path,
  outfile = "analysis.html",
  recipe = FALSE,
  parallel = TRUE,
  infile = NULL
)
```

Arguments

path	Path to a subdirectory containing the preprocessed anonymized (simulated) data files (see simulate_resp_files).
outfile	Path to the HTML output file.
recipe	Include instructions on how to reproduce the analysis.
parallel	Whether to fit models using a single CPU processing core (FALSE) or multiple cores (TRUE, the default).
infile	Path to the R Markdown script; NULL to use the built-in script.

Details

Runs R Markdown script on the data in the provided subdirectory and renders the HTML report to outfile. The master R Markdown script can be accessed using:

```
rmarkdown::draft("analysis.Rmd","illusory-truth-analysis-sim","truthiness")
```

Note that this script can take *very* long to run, depending on the size of the simulated dataset, the number of processing cores, and the computational power of the hardware.

Examples

```
td_raw <- tempfile()  # temp dir for raw data
td_anon <- tempfile() # temp dir for preprocessed data

## simulate data and preprocess it

set.seed(62)
simulate_resp_files(32, path = td_raw, overwrite = TRUE)


## temporary files
tf1 <- tempfile(fileext = ".html")
tf2 <- tempfile(fileext = ".html")

## run the built-in R Markdown preprocessing script
pp_report <- preprocess_simulated(path = td_raw, outpath = td_anon,
                                  report = tf1)

## run the built-in R Markdown analysis script
## this can take very long due to the CLMM fits
a_report <- reproduce_analysis_sim(path = td_anon,
                                   outfile = tf2,
                                   parallel = FALSE)

browseURL(a_report)

## clean up
file.remove(pp_report)
file.remove(a_report)
```

```
## clean up
unlink(td_raw, TRUE, TRUE)
unlink(td_anon, TRUE, TRUE)
```

`run_equiv` *Fit CLMM and Run Equivalence Test*

Description

Fit CLMM and Run Equivalence Test

Usage

```
run_equiv(.data, main_effect = FALSE, delta = 0.14)
```

Arguments

`.data`	Data frame, with the format as resulting from a call to `gen_data`.
`main_effect`	Whether to test the main effect of repetition (TRUE) or the repetition-by-interval interaction (FALSE; the default).
`delta`	Smallest (raw) effect size of interest (log odds scale).

Details

This function is intended to be used in data simulation.

Value

A vector with p-values; the element(s) named `simple` provide p-values for simple effects; the element(s) named `equiv` provides the p-value for the corresponding equivalence test.

Examples

```
set.seed(62)
dat <- gen_data(24)
run_equiv(dat, main_effect = TRUE)
```

simulate_category_guess *Simulate Guessing During the Categorization Task*

Description

Run simulations tabulating the number of correct guesses assuming a participant is just guessing during the categorization task. This can be used to estimate a chance baseline on the 64 categorization trials.

Usage

```
simulate_category_guess(nruns = 10000)
```

Arguments

nruns	Number of simulation runs.

Value

A vector of length nruns with the number of correct guesses.

Examples

```
n_correct <- simulate_category_guess(1000)
hist(n_correct)
mean(n_correct)
```

simulate_resp_files *Simulate Response Data Files From Longitudinal Illusory Truth Study*

Description

Simulate Response Data Files From Longitudinal Illusory Truth Study

Usage

```
simulate_resp_files(
  nsubj,
  phase_eff = c(0, 0, 0, 0),
  path,
  overwrite = FALSE,
  p_too_fast = 0.01,
  p_too_slow = 0.01,
  p_incomplete = 0.01,
```

```
  p_cheat = 0.01,
  p_no_consent_all = 0.01,
  p_no_consent_phase = 0.01,
  p_nonnative = 0.01,
  p_repeater = 0.01,
  duration_range_1 = c(180, 2400),
  duration_range_all = c(60, 1800)
)
```

Arguments

nsubj	Number of subjects; must be a multiple of 8.
phase_eff	A four-element vector giving the size of the illusory truth effect at each of the four phases (on the log odds scale). Use `rep(0,4)` for testing Type I error rate. A value of .14 gives an effect of approximately 1/10 of a scale point.
path	Path to subdirectory where resulting files will be stored; will be created if it does not exist.
overwrite	Whether to overwrite the subdirectory if it exists.
p_too_fast	Probability that the respondent completed the task faster than the cutoff time ('Duration (in seconds)' less than `duration_range_1[1]` for Phase 1, less than `duration_range_all[1]` for all other phases).
p_too_slow	Probability that the respondent completed the task slower than the cutoff time ('Duration (in seconds)' greater than `duration_range_1[2]` for Phase 1, greater than `duration_range_all[2]` for all other phases).
p_incomplete	Probability that the respondent failed to complete the task ('Finished' = FALSE).
p_cheat	Probability that the respondent looked up answers ('cheat' = "Yes...")
p_no_consent_all	Probability the respondent refused consent to the full study.
p_no_consent_phase	Probability the respondent refused consent to a phase of the study.
p_nonnative	Probability the respondent is not a native English speaker.
p_repeater	Probability that the respondent just pressed the same key over and over for at least one phase.
duration_range_1	Two-element vector giving the range of acceptable task durations for Phase 1.
duration_range_all	Two-element vector giving the range of acceptable task durations for Phases 2, 3, and 4.

Details

Simulates response data and writes a set of CSV files out to `path` in Qualtrics format. The file names are of the format `PXLY.csv`, where X is the phase number (1-4) and Y is the list number (1-8). So P2L6.csv is the file for phase 2 of list 6. When we ran a pilot study, we discovered that the data files had a somewhat different structure from this, but we nevertheless opted to retain this function rather than rewriting it to match the new format.

Value

A character vector with the names of the data files.

Examples

```
td <- tempdir()
simulate_resp_files(40, path = td, overwrite = TRUE)
dir(td) # show the response files
unlink(td, TRUE, TRUE) # cleanup
```

`truthiness` *truthiness: Longitudinal Study of the Illusory Truth effect*

Description

The truthiness package provides functions and data associated with a longitudinal study of the Illusory Truth effect conducted by Henderson et al. (2020). The illusory truth effect is the observation that people rate repeated statements as more likely to be true. We tested the trajectory of the illusory truth effect by collecting truth ratings for statements repeated across four time intervals: immediately, one day, one week, and one month following initial presentation. The package contains the anonymized data for the study as well as functions for analyzing the data, running simulations, and calculating power. Further details about the project are available at https://osf.io/nvugt/ including Stage 1 of the Registered Report at the Journal of Cognition (https://osf.io/vqnx2/).

References

Henderson EL, Simons DJ, Barr DJ (2020). "The Trajectory of Truth: A Longitudinal Study of the Illusory Truth Effect (Stage 1 Registered Report)." *Journal of Cognition*. https://osf.io/vqnx2/.

`truth_trajectory_data` *Data from the Longitudinal Illusory Truth Study*

Description

A collection of four data frames representing the anonymized longitudinal data in tidy format from Henderson et al. (2020).

Usage

```
sessions

phases

cjudgments

ratings
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 631 rows and 17 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2282 rows and 8 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 39406 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 72215 rows and 4 columns.

Details

Each data frame contains a subset of the following variables:

`ID` Participant identifier.

`list_id` Stimulus list identifier.

`phase_id` Phase number (1-4).

`stim_id` Stimulus identifier.

`Age` Age of participant in years.

`Gender` Gender of participant.

`Nationality` Nationality of participant.

`NativeLang` Native language of participant.

`duration_secs` Duration of the phase in seconds.

`category` Category the participant selected for this statement.

`trating` Truth rating on a seven-point scale, 1=Definitely False, 7=Definitely True.

`excl_phase` Phase in which participant was excluded (`NA` if never excluded).

`excl_reason` Reason for participant exclusion.

`p_excl_reason` Reason for phase exclusion.

`chk_anydata` Whether there is ratings data for at least one phase for this participant after phase-level exclusions.

`chk_consent_all` Whether participant gave consent for all phases.

`chk_consent` Whether participant gave consent for this phase.

`chk_dur_all` Whether all phase durations for this participant were within an acceptable range.

`chk_finished` Whether participant completed the rating task for this phase.

`chk_native` Whether participant is a native speaker of English.

`chk_nocheat` Whether participant never looked up answers.

`chk_noduplicates` Whether there were no duplicated sessions.

`chk_noflatline` Whether the participant did not produce 'flatline' responses.

`chk_notmanex` Whether the participant (or phase) is not manually excluded.

`keep` Logical value, whether to keep (TRUE) or exclude (FALSE) participant (or phase data); this is a boolean "and" of all of the exclusion criteria (`chk_*` variables) for that participant (or phase).

The sessions data frame contains information about the 631 participants who were recruited to the study. The chk_* variables are logical variables representing exclusion criteria. The variable keep is a boolean "AND" of these criteria, and thus has a value of TRUE for participants who are to be included and FALSE for those who are to be excluded.

The phases data frame contains data from the 2,282 phases that were initiated by participants. Each participant who was not excluded during data collection had the opportunity to complete up to four phases of data collection taking place (1) immediately after the exposure phase; (2) one day after exposure; (3) one week after exposure; and (4) one month after exposure. The chk_* variables in this data frame represent exclusion criteria, and keep is a boolean "AND" of those criteria along with the keep variable from the sessions table. In other words, to apply the full set of participant-level and phase-level exclusion criteria for the study, simply include those rows in phases where keep is set to TRUE, and join this table to the others in the set; see the example below.

The cjudgments table contains 39,406 category judgments that were produced in the exposure phase (phase 1) of the study.

The ratings data frame contains 72,215 truth ratings of the stimulus statements used in the study. Ratings were on a 1-7 scale (1 = definitely false; 7 = definitely true).

References

Henderson EL, Simons DJ, Barr DJ (2020). "The Trajectory of Truth: A Longitudinal Study of the Illusory Truth Effect (Stage 1 Registered Report)." *Journal of Cognition*. https://osf.io/vqnx2/.

Examples

```
library(dplyr)

## apply exclusions and merge with ratings data
ratings_incl <- phases %>%
  filter(keep) %>%                                          # apply exclusions
  inner_join(sessions %>% select(ID, list_id), "ID") %>%  # get list ID
  inner_join(ratings, c("ID", "phase_id"))

## look up conditions and calculate cell means
ratings_incl %>%
  inner_join(stimulus_conditions, c("list_id", "stim_id")) %>% # lookup condition
  group_by(repetition, interval) %>%
  summarize(rating_mean = mean(trating),
            rating_sd = sd(trating),
            N = n()) %>%
  ungroup()
```

---

truth_trajectory_design

Design of the Longitudinal Illusory Truth Study

Description

A collection of four data frames in tidy format that contain information about stimuli and experimental design for Henderson et al. (2020).

Usage

```
stimulus_materials

stimulus_categories

stimulus_conditions

presentation_lists
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 128 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 170 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1024 rows and 4 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1536 rows and 5 columns.

Details

Each data frame contains a subset of the following 11 variables:

`stim_id` Unique identifier of the stimulus.

`statement` The statement.

`actual_truth` Actual truth or falsity of the statement.

`choice` Category number; allows for more than one correct categorization.

`category` Name of category.

`list_id` Unique identifier of stimulus presentation list.

`repetition` Whether the statement was repeated or novel.

`interval` Interval (phase) in which the truth-rating was performed (immediate = 1, 1 day = 2, 1 week = 3, 1 month = 4).

`phase_id` Phase (1-4) in which the stimulus will be presented.

`task` Which task the stimulus is presented in (categorize or rate truth).

`task_id` Unique identifier of the task/stimulus combination.

The `stimulus_materials` data frame lists all 128 trivia statements used in the study. These statements were adapted from Nadarevic and Erdfelder (2014) and De Keersmaecker et al. (2020), who in turn adapted their materials from those original compiled by Unkelbach and Rom (2017). Each stimulus is given a unique identifier, `stim_id`, that appears across related tables.

The `stimulus_categories` data frame gives information about which category each stimulus statement belongs to. Note that each statement can belong to more than one category simultaneously.

The stimulus_conditions data frame provides a 'lookup' table that associates each stimulus (stim_id) from each presentation list (list_id) with its experimental conditions (repetition, and interval).

There were eight separate presentation lists used in the study for counterbalancing purposes. These lists are provided in the presentation_lists data frame.

References

De Keersmaecker J, Dunning D, Pennycook G, Rand DG, Sanchez C, Unkelbach C, Roets A (2020). "Investigating the robustness of the illusory truth effect across individual differences in cognitive ability, need for cognitive closure, and cognitive style." *Personality and Social Psychology Bulletin*, **46**(2), 204–215.

Henderson EL, Simons DJ, Barr DJ (2020). "The Trajectory of Truth: A Longitudinal Study of the Illusory Truth Effect (Stage 1 Registered Report)." *Journal of Cognition*. https://osf.io/vqnx2/.

Nadarevic L, Erdfelder E (2014). "Initial judgment task and delay of the final validity-rating task moderate the truth effect." *Consciousness and Cognition*, **23**, 74–84.

Unkelbach C, Rom SC (2017). "A referential theory of the repetition-induced truth effect." *Cognition*, **160**, 110–126.

---

truth_trajectory_models

Fitted Models from the Longitudinal Illusory Truth Study

Description

Fitted models from the pre-registered analysis of Henderson et al. (2020), which have been stored as an objects in the package because the fitting process is too slow to allow them to be re-created when needed.

Usage

```
truth_trajectory_models
```

Format

This object is a named list with six elements, with each element representing a fitted model object of class "clmm", resulting from a call to the clmm function. The named elements are:

main_base Base model for testing the main effect; model formula is mod1 <-T ~ R + I1 + I2 + I3 + R:I1 + R:I2 + R:I3 + (R | subj_id) + (R | stim_id).

main_comp Comparison model for testing the main effect; model formula identical to main_base except the fixed effect R has been excluded.

`ix_base` Base model for testing the repetition-by-interval interaction; Model formula is `mod3 <-T ~ R + I1 + I2 + I3 + R:I1 + R:I2 + R:I3 + (R:I1 + R:I2 + R:I3 | subj_id) + (R:I1 + R:I2 + R:I3 | stim_id)`.

`ix_comp` Comparison model for testing the interaction; Model formula is identical to `ix_base` except for exclusion of the fixed effects terms `R:I1`, `R:I2`, and `R:I3`.

`ix2` Same as `ix_base` except predictors included as factors rather than numerical predictors to enable use of functions from the `emmeans` package (for equivalence and follow-up tests).

`main2` Same as `main_base` except predictors included as factors rather than numerical predictors to enable use of functions from the `emmeans` package (for equivalence and follow-up tests).

Examples

```
library(ordinal)

## print model information
summary(truth_trajectory_models$ix_base)

## likelihood ratio test, testing repetition-by-interval interaction
anova(truth_trajectory_models$ix_base,
      truth_trajectory_models$ix_comp)
```

---

validate_filenames *Validate Simulated Data Filenames*

Description

Make sure all the files needed for the analysis are present in a directory containing simulated data.

Usage

```
validate_filenames(path)
```

Arguments

`path`	Path to the files.

Details

Output files from the study must match the pattern `PXLY.csv` where X is phase number (1-4) and Y is list number (1-8).

Value

`TRUE`, if files in the directory `path` have names in the expected format; otherwise, an error is thrown.

warn *Warn About Simulated Data*

Description

Check whether the data in `subdir` is simulated data and generate a warning to include in an R Markdown document.

Usage

```
warn(subdir)
```

Arguments

`subdir`	Subdirectory with the anonymized data.

Value

A character vector containing a warning about simulated data, which is wrapped in HTML `<div>` tags for inclusion in the HTML document output by `preprocess_simulated`.

Index