Package: simling (via r-universe)

November 17, 2026

Title Simulated Linguistic Data for Teaching and Learning Statistics

Version 0.1.1

Description Simulated data sets for teaching and learning statistics in linguistics, from phonetics, phonology, prosody, psycholinguistics, syntax, morphology, semantics and pragmatics, conversation analysis, corpus linguistics, sociolinguistics, dialectology, bilingualism, second language research, language acquisition, clinical linguistics, sign language, forensic speech science, historical linguistics, language evolution and typology, and from languages including German, Italian, Portuguese, Spanish, Hungarian, Mandarin, Cantonese, Korean and German Sign Language. Responses include continuous measures, reaction times, binary and multi-category outcomes, ratings, counts, censored ages, effect sizes, eye-tracking and EEG measures and time series, in data sets from 18 to over 20,000 rows, plus one messy export for data cleaning. Each help page states the true model the data were simulated from, so students can check whether their analysis recovers it; the simulation code is in the source repository. Researchers planning a study can use the data sets to see what a type of data looks like and to try out its analysis before collecting their own. A function lists the data sets by branch, language, size and response.

License MIT + file LICENSE

URL https://github.com/mcanzi/simling, https://mcanzi.de/simling/

BugReports https://github.com/mcanzi/simling/issues

Depends R (>= 3.5)

Imports tibble

Suggests lme4, MASS, metafor, mgcv, nnet, ordinal, pscl, randomForest, rpart, spelling, survival, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

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

Date/Publication 2026-11-17 07:29:33 UTC

RemoteUrl https://github.com/mcanzi/simling

RemoteRef HEAD

RemoteSha 76bbb6a0985dc83b5179f37dcef8285872e0047b

Contents

accent . . . 3
acceptability . . . 4
address_forms . . . 6
agreement_attraction . . . 7
aphasia_naming . . . 8
bouba_kiki_meta . . . 9
cantonese_tone_merger . . . 11
code_switching . . . 12
dgs_iconicity . . . 13
dialect_survey . . . 15
disfluencies . . . 16
do_support . . . 17
first_words . . . 19
forensic_comparison . . . 20
formants . . . 21
freq_length . . . 22
german_plurals . . . 23
german_relatives . . . 24
hungarian_harmony . . . 25
iterated_learning . . . 26
korean_stops . . . 28
l2_frequency . . . 29
language_attitudes . . . 30
lexical_accuracy . . . 31
lexical_decision . . . 32
lexical_decision_l2 . . . 33
mandarin_tones . . . 34
megastudy . . . 35
messy_export . . . 36
n400 . . . 37
naming_pilot . . . 39
overregularization . . . 40
passives . . . 41

perception_training . . . 42
price_f2 . . . 43
pupil_noise . . . 44
reading_skipping . . . 45
relativizers . . . 46
rhythm_metrics . . . 47
rt_frequency . . . 49
scalar_implicature . . . 50
simling_datasets . . . 51
speech_styles . . . 52
speech_to_song . . . 52
style_voicing . . . 54
td_deletion . . . 55
tobi_agreement . . . 56
turn_taking . . . 57
visual_world . . . 59
vocabulary_growth . . . 60
vot_identification . . . 62
vowel_discrimination . . . 63
vowel_duration . . . 64
vowel_voicing . . . 65
web_registers . . . 66
word_order . . . 68
zipf_rt . . . 70

Index **71**

`accent` *Acoustic cues to a foreign accent*

Description

Six acoustic measures from 40 tokens of each of 30 speakers of English: 15 L1 English and 15 L1 Italian. Three measures differ by L1, one by gender and two by nothing. For classification with random forests and cross-validation by speaker.

Usage

```
accent
```

Format

A tibble with 1200 rows and 9 columns:

speaker `S01` to `S30`.

l1 `"English"` or `"Italian"`.

gender `"female"` or `"male"`.

vot Voice onset time of /p t k/ in ms.

vowel_ratio Ratio of stressed to unstressed vowel duration.

rate Speech rate in syllables per second.

f0 Mean f0 in Hz.

h1h2 H1-H2 in dB.

intensity Mean intensity in dB.

Details

True model: speaker means for English vs Italian are 70 vs 45 ms VOT, 1.6 vs 1.2 vowel ratio and 5.2 vs 4.7 syllables/s. VOT and vowel ratio vary log-normally around the speaker mean, so they stay positive. f0 is 210 Hz for female and 120 Hz for male speakers. H1-H2 differs between speakers but not by L1 or gender, and intensity varies only between tokens.

Source

`data-raw/accent.R`

Examples

```
aggregate(cbind(vot, vowel_ratio, rate, f0) ~ l1 + gender, data = accent, FUN = mean)
speakers <- levels(accent$speaker)
test <- is.element(accent$speaker, speakers[seq(1, length(speakers), by = 5)])
m <- glm(l1 ~ vot + vowel_ratio + rate, data = accent[!test, ], family = binomial)
predicted <- predict(m, accent[test, ], type = "response") > 0.5
mean(predicted == (accent$l1[test] == "Italian"))
if (requireNamespace("randomForest", quietly = TRUE)) {
  set.seed(1)
  rf <- randomForest::randomForest(l1 ~ vot + vowel_ratio + rate + f0 + h1h2 + intensity,
                                   data = accent[!test, ], importance = TRUE)
  print(round(randomForest::importance(rf, type = 1), 1))
  mean(predict(rf, accent[test, ]) == accent$l1[test])
}
```

---

acceptability *Acceptability judgments for whether-islands*

Description

7-point acceptability ratings in a 2 x 2 factorial design: dependency length (short or long) by structure (non-island or island). Island effects show up as an interaction: long dependencies out of islands are worse than the two main effects predict. 48 participants rate 32 items, each item in one condition per participant (four counterbalanced lists).

Usage

```
acceptability
```

Format

A tibble with 1536 rows and 6 columns:

participant Participant ID, `P01` to `P48`.

list Counterbalancing list, 1 to 4.

item Item ID, `i01` to `i32`.

length Dependency length, `"short"` or `"long"`.

structure `"non-island"` or `"island"`.

rating Rating from 1 (unacceptable) to 7 (acceptable).

Details

True model: a cumulative logit model with thresholds at -2.5, -1.5, -0.5, 0.5, 1.5 and 2.5 on a latent scale `1.5 - 0.6 * long - 0.4 * island - 0.9 * long * island`. By-participant intercepts (SD 0.8) and interaction slopes (SD 0.3); by-item intercepts (SD 0.4). The design follows Experiment 1 of Sprouse, Wagers and Phillips (2012), who crossed gap position (matrix or embedded clause, i.e. a short or long dependency) with structure and used a 7-point scale; for whether islands they found both main effects and a superadditive interaction. The number of participants and items, the effect sizes and the ratings are simulated.

Source

`data-raw/acceptability.R`

References

Sprouse, J., Wagers, M. & Phillips, C. (2012). A test of the relation between working-memory capacity and syntactic island effects. *Language*, 88(1), 82-123. doi:10.1353/lan.2012.0004

Examples

```
round(tapply(acceptability$rating, acceptability[c("length", "structure")], mean), 2)

if (requireNamespace("ordinal", quietly = TRUE)) {
  acceptability$rating <- factor(acceptability$rating, ordered = TRUE)
  m <- ordinal::clmm(rating ~ length * structure + (1 | participant) + (1 | item),
                     data = acceptability)
  coef(summary(m))
}
```

address_forms *Forms of address: du or Sie*

Description

Whether German speakers address a stranger with informal *du* or formal *Sie*, by speaker age group and setting. 30 speakers per cell, one observation each. For contingency tables and chi-squared tests.

Usage

```
address_forms
```

Format

A tibble with 180 rows and 4 columns:

speaker Speaker ID, `S001` to `S180`.

age_group `"under 30"` or `"over 50"`.

setting `"university"`, `"office"` or `"shop"`.

form `"Sie"` or `"du"`.

Details

True model: the probability of *du* is 0.85 (university), 0.35 (shop) and 0.45 (office) for speakers under 30, and 0.40, 0.08 and 0.15 for speakers over 50. These probabilities are simulated. Strangers are by default addressed with *Sie*, and variation in address depends more on age than on region (Kretzenbacher 2011). In a 2021 survey reported by Kretzenbacher and Hensel-Börner (2024), 44% of customers aged 18 to 29 liked being addressed with *du* in shops, against 23% of those aged 60 and over.

Source

`data-raw/address_forms.R`

References

Kretzenbacher, H. L. (2011). Perceptions of national and regional standards of addressing in Germany and Austria. *Pragmatics*, 21(1), 69-83. doi:10.1075/prag.21.1.04kre

Kretzenbacher, H. L. & Hensel-Börner, S. (2024). Pronominal address in German sales talk: Effects on the perception of the salesperson. *Languages*, 9(10), 316. doi:10.3390/languages9100316

Examples

```
tab <- xtabs(~ setting + form + age_group, data = address_forms)
round(prop.table(tab, c(1, 3))[, "du", ], 2)
chisq.test(tab[, , "under 30"])
chisq.test(tab[, , "over 50"])
```

`agreement_attraction` *Eye-tracking while reading: agreement attraction*

Description

Reading measures on the verb in sentences like *The key to the cabinet(s) was/were rusty*: grammaticality by number of the attractor noun. A plural attractor makes ungrammatical *were* easier to read. 40 participants read 32 items, each in one condition (four counterbalanced lists).

Usage

```
agreement_attraction
```

Format

A tibble with 1280 rows and 10 columns:

participant Participant ID, `P01` to `P40`.

list Counterbalancing list, 1 to 4.

item Item ID, `i01` to `i32`.

grammaticality `"grammatical"` or `"ungrammatical"`.

attractor Number of the attractor noun, `"singular"` or `"plural"`.

skipped 1 if the verb was skipped in first-pass reading.

first_fixation Duration of the first fixation on the verb in ms; `NA` if skipped.

gaze_duration Sum of first-pass fixations in ms; `NA` if skipped.

regression 1 if the reader made a regression out of the verb in first pass; `NA` if skipped.

total_time Sum of all fixations on the verb in ms; `NA` if never fixated.

Details

True model:

```
log(gaze_duration) = 5.55 + 0.14 * ungrammatical
  - 0.09 * ungrammatical * plural
logit(regression) = -1.9 + 0.9 * ungrammatical
  - 0.5 * ungrammatical * plural
```

Gaze durations have by-participant (SD 0.15) and by-item (SD 0.08) intercepts and residual SD 0.3. Regressions out of the verb in first pass have a by-participant SD of 0.5. Skipping rate about 14%, unrelated to condition. The first fixation equals the gaze duration in 65% of trials and is 50 to 85% of it otherwise. Total time adds a rereading time (log-normal, median about 220 ms) with probability `plogis(-1 + 0.6 * ungrammatical + 1.5 * regression)`. The sentences follow Bock and Miller (1991), who found attraction errors in production. In reading, a plural attractor reduces the disruption caused by an ungrammatical verb (Wagers, Lau and Phillips 2009). The effect sizes and data are simulated.

Source

data-raw/agreement_attraction.R

References

Bock, K. & Miller, C. A. (1991). Broken agreement. *Cognitive Psychology*, 23(1), 45-93. doi:10.1016/00100285(91)900037

Wagers, M. W., Lau, E. F. & Phillips, C. (2009). Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, 61(2), 206-237. doi:10.1016/j.jml.2009.04.002

Examples

```
aggregate(gaze_duration ~ grammaticality + attractor, data = agreement_attraction,
          FUN = mean)
mean(agreement_attraction$skipped)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(log(gaze_duration) ~ grammaticality * attractor + (1 | participant) +
                    (1 | item), data = agreement_attraction)
  lme4::fixef(m)
}
```

---

aphasia_naming *Picture naming before and after therapy for word-finding difficulties*

Description

16 people with chronic aphasia name 120 pictures each, twice before and once after eight weeks of cueing therapy. Half of each person's items were practised in therapy and half were not. Naming of treated items improves for almost every participant, and the two baselines do not differ. Naming of untreated items improves only for the five participants with a phonological profile: relatively good semantic processing and impaired phonological output. For mixed logistic regression of repeated measures, three-way interactions and generalisation in therapy studies.

Usage

```
aphasia_naming
```

Format

A tibble with 5760 rows and 6 columns:

participant Participant ID, `P01` to `P16`.

profile `"other"` or `"phonological"`.

item Picture, numbered within participant (`P01_001`).

set `"untreated"` or `"treated"`.

session `"baseline 1"`, `"baseline 2"` or `"post-therapy"`.

correct 1 if the picture was named correctly, 0 if not.

Details

True model: the logit of a correct response is `p0 + i0 + post * gain`. Participant intercepts `p0` have mean -0.45 (SD 0.9) and item effects `i0` SD 1.2. For treated items, `gain` is the participant's own gain (mean 1, SD 0.35). For untreated items it is 0.7 in the phonological profile and 0.05 otherwise, plus noise (SD 0.1). The design follows Best et al. (2013). In their 16 participants, naming of treated items rose from 43.5 to 62% correct, and only participants with relatively good semantic and impaired phonological processing generalised to untreated items. The number of items, the size of the subgroup and all data are simulated.

Source

`data-raw/aphasia_naming.R`

References

Best, W., Greenwood, A., Grassly, J., Herbert, R., Hickin, J. & Howard, D. (2013). Aphasia rehabilitation: Does generalisation from anomia therapy occur and is it predictable? A case series study. *Cortex*, 49(9), 2345-2357. doi:10.1016/j.cortex.2013.01.005

Examples

```
round(tapply(aphasia_naming$correct, aphasia_naming[c("set", "session")], mean), 2)
round(tapply(aphasia_naming$correct,
             aphasia_naming[c("profile", "set", "session")], mean), 2)
p01 <- subset(aphasia_naming, participant == "P01" & set == "treated" &
                session != "baseline 1")
mcnemar.test(table(before = p01$correct[p01$session == "baseline 2"],
                   after = p01$correct[p01$session == "post-therapy"]))

if (requireNamespace("lme4", quietly = TRUE)) {
  d <- subset(aphasia_naming, session != "baseline 1")
  m <- lme4::glmer(correct ~ session * set * profile + (1 | participant) + (1 | item),
                   data = d, family = binomial,
                   control = lme4::glmerControl(optimizer = "bobyqa"))
  round(lme4::fixef(m), 2)
}
```

---

bouba_kiki_meta *A meta-analysis of the bouba/kiki effect*

Description

40 published studies of the bouba/kiki effect: how many participants matched *bouba* with a round and *kiki* with a spiky shape. The studies tested speakers of 21 languages. Studies with a non-significant result were less likely to be published, so the published effects are inflated. For random-effects meta-analysis, moderators, funnel plots and publication bias.

Usage

```
bouba_kiki_meta
```

Format

A tibble with 40 rows and 8 columns:

study Study ID, `S01` to `S40`.

year Year of publication.

language Language of the participants.

script Writing system of the language: `"other"` or `"Roman"`.

n Number of participants.

congruent Participants giving the congruent match.

log_odds Log-odds of a congruent match, `log((congruent + 0.5) / (n - congruent + 0.5))`.

se Standard error of `log_odds`.

Details

True model: each study's true log-odds is `0.85 + 0.2 * Roman` plus a study effect with SD (tau) 0.35, about 70% congruent matches. Of 90 simulated studies, those with a significant positive result were published with probability 0.95 and the rest with probability 0.3; the first 40 published studies are included. Studies, years and samples are invented.

Source

`data-raw/bouba_kiki_meta.R`

References

Ćwiek, A., Fuchs, S., Draxler, C., Asu, E. L., Dediu, D., Hiovain, K., Kawahara, S., Koutalidis, S., Krifka, M., Lippus, P., Lupyan, G., Oh, G. E., Paul, J., Petrone, C., Ridouane, R., Reiter, S., Schümchen, N., Szalontai, Á., Ünal-Logacev, Ö., Zeller, J., Perlman, M. & Winter, B. (2022). The bouba/kiki effect is robust across cultures and writing systems. *Philosophical Transactions of the Royal Society B*, 377(1841), 20200390. doi:10.1098/rstb.2020.0390

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  m <- metafor::rma(yi = log_odds, sei = se, data = bouba_kiki_meta)
  print(m)
  print(metafor::rma(yi = log_odds, sei = se, mods = ~ script, data = bouba_kiki_meta))
  print(metafor::regtest(m))
  metafor::funnel(m)
}
```

`cantonese_tone_merger` *Merging of the Cantonese rising tones*

Description

Hong Kong Cantonese has two rising tones, T2 (high rising, 25) and T5 (low rising, 23), which some younger speakers merge. 90 speakers, half aged 18 to 25 and half 26 to 74, each produce 10 T2 and 10 T5 words; the size of the f0 rise is measured in semitones. For a sound change in apparent time and for individual differences: mergers form a small, separate group of speakers, most of them young.

Usage

```
cantonese_tone_merger
```

Format

A tibble with 1800 rows and 7 columns:

speaker `S01` to `S90`.

age Age in years.

gender `"female"` or `"male"`.

word `w01` to `w20`.

tone `"T2"` or `"T5"`.

log_freq log10 word frequency, 1 to 5; no effect.

rise f0 rise from onset to offset in semitones.

Details

True model: each speaker is a merger with probability `0.22 * plogis((30 - age) / 4)`, about 0.2 for speakers under 25. Non-mergers rise by 7 semitones in T2 and 3.5 in T5; mergers by 5.4 and 5.0, so they keep a small difference. By-speaker intercepts have SD 0.6 and token SD is 0.9; word frequency and gender have no effect. Mok, Zuo and Wong (2013) judged 28 of 169 students aged 18 to 22 to merge tones. Their mergers still kept T2 and T5 slightly apart, and neither word frequency nor gender made a difference. Because all their speakers were young, the decline of merging with age is a design choice, and the values and data are simulated.

Source

`data-raw/cantonese_tone_merger.R`

References

Mok, P. P. K., Zuo, D. & Wong, P. W. Y. (2013). Production and perception of a sound change in progress: Tone merging in Hong Kong Cantonese. *Language Variation and Change*, 25(3), 341-370. doi:10.1017/S0954394513000161

Examples

```
rise <- tapply(cantonese_tone_merger$rise,
               cantonese_tone_merger[c("speaker", "tone")], mean)
age <- tapply(cantonese_tone_merger$age, cantonese_tone_merger$speaker, unique)
difference <- rise[, "T2"] - rise[, "T5"]
hist(difference, breaks = 20)
plot(age, difference)
table(merger = difference < 2, under_40 = age < 40)
```

---

code_switching *Code-switching by Portuguese-German bilinguals*

Description

Switches between Portuguese and German in conversations of 90 speakers of Portuguese heritage in Germany, 30 from each of three generations. Each speaker is recorded in four conversations on different topics, with a bilingual or a monolingual (German) interlocutor. Second-generation speakers switch most, and more of their switches fall within a sentence. Speakers rarely switch with monolingual interlocutors. For count models with an exposure.

Usage

```
code_switching
```

Format

A tibble with 360 rows and 8 columns:

speaker Speaker ID, `S01` to `S90`.

generation `"first"`, `"second"` or `"third"`.

conversation Conversation ID (`speaker_topic`).

topic `"family"`, `"work"`, `"leisure"` or `"migration"`.

interlocutor `"monolingual"` or `"bilingual"`.

utterances Utterances by the speaker in the conversation.

switches Switches between the languages.

intrasentential Of these, switches within a sentence.

Details

True model: switches are negative binomial (size 6) with mean `rate * utterances`, where `rate` is 0.04 (first), 0.12 (second) or 0.07 (third generation), multiplied by 1.3 (family), 0.8 (work), 1 (leisure) or 1.1 (migration), by 0.25 with a monolingual interlocutor, and by a speaker effect (SD 0.35 on the log scale). Intrasentential switches are binomial with probability 0.25, 0.6 and 0.4 for the three generations. These generation differences are a design choice. They resemble the pattern Backus (1996) describes for Turkish immigrants in the Netherlands, where the intermediate generation switched most often, but how much and in what way each generation switches differs between communities.

Source

data-raw/code_switching.R

References

Backus, A. (1996). Het intergenerationele codewisseling-continuum binnen de Turkse gemeenschap in Nederland. *Toegepaste Taalwetenschap in Artikelen*, 54, 25-37. doi:10.1075/ttwia.54.03bac

Poplack, S. (1980). Sometimes I'll start a sentence in Spanish y termino en español: Toward a typology of code-switching. *Linguistics*, 18(7-8), 581-618. doi:10.1515/ling.1980.18.78.581

Examples

```
if (requireNamespace("MASS", quietly = TRUE)) {
 m <- MASS::glm.nb(switches ~ generation + topic + interlocutor + offset(log(utterances)),
                   data = code_switching)
  round(exp(coef(m)), 2)
}
totals <- aggregate(cbind(intrasentential, switches) ~ generation, data = code_switching,
                    FUN = sum)
totals$intrasentential / totals$switches
```

---

dgs_iconicity *Iconicity ratings for signs of German Sign Language*

Description

30 deaf signers of German Sign Language (DGS) and 30 hearing non-signers rate 300 DGS signs for iconicity (1 = not iconic, 7 = very iconic). Ratings by the two groups are correlated (r about 0.78 for sign means, as in Trettenbrein et al. 2021). Signers rate signs as more iconic overall, and non-signers rate verbs as more iconic than nouns. More iconic signs are acquired earlier. For mixed models of ratings with crossed raters and items.

Usage

```
dgs_iconicity
```

Format

A tibble with 18000 rows and 7 columns:

rater Rater ID, `R01` to `R60`.

group `"hearing non-signer"` or `"deaf signer"`.

sign Sign ID, `sign001` to `sign300`.

category `"noun"` or `"verb"`.

handedness `"one-handed"` or `"two-handed"`; no effect.

aoa Age of acquisition of the sign in years, as rated by deaf signers.

rating Iconicity rating, 1 to 7.

Details

True model:

```
rating = 3.3 + iconicity + g0 + r0 + signer + 0.5 * verb * non_signer + e
```

rounded and limited to 1-7. Each sign's iconicity has SD 1.2, `g0` is a deviation of each sign within each group (SD 0.6), raters differ by SD 0.5 and `e` has SD 0.9. Age of acquisition is `5.5 - 0.6 * iconicity` years plus noise (SD 1.3). The group differences follow findings for DGS (signers higher) and for ASL (Sehyr and Emmorey 2019) and Israeli Sign Language (Fuks 2023) (verbs higher for non-signers).

Source

`data-raw/dgs_iconicity.R`

References

Trettenbrein, P. C., Pendzich, N.-K., Cramer, J.-M., Steinbach, M. & Zaccarella, E. (2021). Psycholinguistic norms for more than 300 lexical signs in German Sign Language (DGS). *Behavior Research Methods*, 53(5), 1817-1832. doi:10.3758/s1342802001524y

Fuks, O. (2023). Iconicity perception under the lens of iconicity rating and transparency tasks in Israeli Sign Language (ISL). *Sign Language Studies*, 24(1), 46-92.

Sehyr, Z. S. & Emmorey, K. (2019). The perceived mapping between form and meaning in American Sign Language depends on linguistic knowledge and task: Evidence from iconicity and transparency judgments. *Language and Cognition*, 11(2), 208-234. doi:10.1017/langcog.2019.18

Thompson, R. L., Vinson, D. P., Woll, B. & Vigliocco, G. (2012). The road to language learning is iconic: Evidence from British Sign Language. *Psychological Science*, 23(12), 1443-1448. doi:10.1177/0956797612459763

Examples

```
round(tapply(dgs_iconicity$rating, dgs_iconicity[c("group", "category")], mean), 2)
signs <- aggregate(rating ~ sign + group, data = dgs_iconicity, FUN = mean)
signs <- reshape(signs, idvar = "sign", timevar = "group", direction = "wide")
cor(signs[, 2], signs[, 3])
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(rating ~ group * category + (1 | rater) + (1 | sign), data = dgs_iconicity)
  lme4::fixef(m)
}
```

`dialect_survey` *A dialect survey of degemination in Italy*

Description

Degemination (*gatto* pronounced with a single *t*) in the speech of 4 speakers at each of 145 towns in mainland Italy and Sicily, with coordinates for mapping. Northern dialects degeminate, central and southern ones keep geminates, with a sharp transition along a stylised La Spezia-Rimini line at about 44.1° N. Older speakers degeminate more. Each speaker reads 10 words with a geminate.

Usage

```
dialect_survey
```

Format

A tibble with 580 rows and 7 columns:

town Town name.

lat, long Coordinates of the town in decimal degrees.

speaker Speaker ID, `town_1` to `town_4`.

age Age in years.

words Words read with a geminate: always 10.

degeminated Of these, words produced with a single consonant.

Details

True model:

```
logit(degemination) = -1.5 + 6 * (lat - 44.1) + t0 + 0.03 * (age - 50)
```

with by-town intercepts `t0` (SD 0.6). Town names and coordinates are the Italian towns of over 50,000 inhabitants in `maps::world.cities` (without Sardinia). Northern dialects have no consonant length contrast, and geminates are less prominent in the Italian of northern speakers than in central and southern varieties (Bertinetto and Loporcaro 2005). In read Italian the regional difference is small (Mairano and De Iacovo 2020), so the dataset describes dialect speech. The sharp line, the age effect and all data are simulated.

Source

`data-raw/dialect_survey.R`

References

Bertinetto, P. M. & Loporcaro, M. (2005). The sound pattern of Standard Italian, as compared with the varieties spoken in Florence, Milan and Rome. *Journal of the International Phonetic Association*, 35(2), 131-151. doi:10.1017/S0025100305002148

Mairano, P. & De Iacovo, V. (2020). Gemination in Northern versus Central and Southern varieties of Italian: A corpus-based investigation. *Language and Speech*, 63(3), 608-634. doi:10.1177/0023830919875481

Examples

```
dialect_survey$p <- dialect_survey$degeminated / dialect_survey$words
plot(lat ~ long, data = dialect_survey, col = grey(1 - p), pch = 16, asp = 1.3)
abline(h = 44.1, lty = 2)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::glmer(cbind(degeminated, words - degeminated) ~ I(lat - 44.1) + I(age - 50) +
                     (1 | town), data = dialect_survey, family = binomial)
  lme4::fixef(m)
}
```

---

disfluencies *Filled pauses in Italian dialogue*

Description

Filled pauses (*ehm*, *eeh*) in 60 turns from each of 40 speakers in 20 Italian Map Task dialogues, in which an instruction giver guides a follower along a route. Filled pauses are rare: about 4% of turns contain one, about 0.4 per 100 syllables. Many turns are fluent and cannot contain a filled pause at all, so the data have more zeros than a Poisson model predicts. Givers produce more filled pauses than followers. For zero-inflated count models with an exposure.

Usage

```
disfluencies
```

Format

A tibble with 2400 rows and 6 columns:

dialogue Dialogue ID, `D01` to `D20`.

speaker Speaker ID, `S01` to `S40`.

role `"follower"` or `"giver"`.

turn Turn number, 1 to 60.

syllables Length of the turn in syllables.

filled_pauses Number of filled pauses in the turn.

Details

True model: with probability 0.55 a turn is fluent and has no filled pauses. Otherwise the count is Poisson with mean `rate * syllables / 100`, where `rate` is 1.2 for givers and 0.6 for followers, multiplied by a speaker effect (SD 0.5 on the log scale). Turn lengths are log-normal with a median of 9 syllables.

Source

`data-raw/disfluencies.R`

References

Di Napoli, J. (2020). Filled pauses and prolongations in Roman Italian task-oriented dialogue. In *Laughter and Other Non-Verbal Vocalisations Workshop: Proceedings (2020)* (pp. 24-27). Bielefeld. doi:10.4119/lw2020915

Examples

```
m0 <- glm(filled_pauses ~ role + offset(log(syllables / 100)), data = disfluencies,
          family = poisson)
c(observed_zeros = mean(disfluencies$filled_pauses == 0),
  poisson_zeros = mean(dpois(0, fitted(m0))))
if (requireNamespace("pscl", quietly = TRUE)) {
  m <- pscl::zeroinfl(filled_pauses ~ role + offset(log(syllables / 100)) | 1,
                      data = disfluencies)
  exp(coef(m, "count"))
  plogis(coef(m, "zero"))
}
```

---

do_support *The rise of periphrastic do*

Description

Use of *do* (*Did he go?*, *He did not go*) in 300 texts dated 1400 to 1575, in four sentence types: the number of clauses with *do* out of all eligible clauses. In negative declaratives, negative questions and affirmative questions the change follows S-curves with the same slope and different starting points (the Constant Rate Effect). *Do* stays rare in affirmative declaratives, which rise more slowly.

Usage

```
do_support
```

Format

A tibble with 1200 rows and 5 columns:

text Text ID, `T001` to `T300`.

year Year of the text.

context `"affirmative declarative"`, `"negative declarative"`, `"affirmative question"` or `"negative question"`.

do Clauses with *do*.

eligible All clauses in this context in this text.

Details

True model, in log-odds: `intercept + slope * (year - 1350) / 100`, plus by-text intercepts (SD 0.4). The slope is 3.7 per century in the three rising contexts and 2.82 in affirmative declaratives. The intercepts at 1350 are -8.33 (negative declaratives), -5.57 (negative questions), -6.58 (affirmative questions) and -8.32 (affirmative declaratives). These are Kroch's (1989, Tables 4 and 9) estimates from Ellegård's counts for 1400 to 1575; the affirmative questions are his transitive adverbial and yes/no questions. After 1575 the contexts diverge and *do* in affirmative declaratives declines, so the simulation stops at 1575. Texts and counts are simulated.

Source

`data-raw/do_support.R`

References

Ellegård, A. (1953). *The auxiliary do: The establishment and regulation of its use in English*. Stockholm: Almqvist & Wiksell.

Kroch, A. S. (1989). Reflexes of grammar in patterns of language change. *Language Variation and Change*, 1(3), 199-244. doi:10.1017/S0954394500000168

Examples

```
do_support$century <- (do_support$year - 1350) / 100
m <- glm(cbind(do, eligible - do) ~ 0 + context + context:century, data = do_support,
         family = binomial)
round(coef(m), 2)
rising <- subset(do_support, context != "affirmative declarative")
anova(glm(cbind(do, eligible - do) ~ context + century, data = rising, family = binomial),
      glm(cbind(do, eligible - do) ~ context * century, data = rising, family = binomial),
       test = "Chisq")
```

`first_words` *Age of first production of words*

Description

Parents of 60 children acquiring European Portuguese report each month, from 10 to 24 months, whether their child says each of 40 words. For each child and word, the data give the age at which the child first said the word, or 24 months if the child had not said it by the end of the study (a censored observation). Frequent words and nouns are learned earlier. For survival analysis.

Usage

```
first_words
```

Format

A tibble with 2400 rows and 6 columns:

child Child ID, `C01` to `C60`.

word Word ID, `w01` to `w40`.

category `"noun"`, `"verb"` or `"other"`.

log_freq log10 frequency of the word in child-directed speech.

age Age in months at first production, or 24 if censored.

produced 1 if the child said the word by 24 months, 0 if censored.

Details

True model: a discrete-time hazard. In each month from 10 to 24 the probability of first saying a word is `plogis(-5.2 + 0.28 * (age - 10) + 0.6 * (log_freq - 2) + category + c0 + w0)`, with category effects 0.5 (noun), -0.3 (verb) and -0.6 (other), by-child effects `c0` (SD 0.6) and by-word effects `w0` (SD 0.4). About 37% of observations are censored.

Source

`data-raw/first_words.R`

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  km <- survival::survfit(survival::Surv(age, produced) ~ category, data = first_words)
  plot(km, col = 1:3, xlim = c(10, 24))
  m <- survival::coxph(survival::Surv(age, produced) ~ category + log_freq +
                         survival::frailty(child), data = first_words)
  coef(m)
}
```

`forensic_comparison` *Forensic voice comparison scores*

Description

Scores from two systems comparing pairs of recordings of 80 male speakers of German: an automatic speaker recognition system and a formant-based system. Every speaker is compared with himself (same-speaker comparisons, recordings from two sessions) and with every other speaker (different-speaker comparisons). Higher scores mean more similar voices. For calibrating scores into likelihood ratios, computing Cllr and comparing or fusing systems.

Usage

```
forensic_comparison
```

Format

A tibble with 3240 rows and 6 columns:

comparison Comparison ID, `C0001` to `C3240`.

speaker_a, speaker_b The two speakers, `S01` to `S80`.

same_speaker `"no"` or `"yes"`.

score_auto Score of the automatic system.

score_formant Score of the formant-based system.

Details

True model: automatic scores are normal with mean 2.5 (SD 1) for same-speaker and -1 (SD 1.3) for different-speaker comparisons; formant scores have means 1 (SD 1) and -0.5 (SD 1.1). Each speaker shifts his scores by a speaker effect (SD 0.4 automatic, 0.3 formant; the mean of the two speakers in different-speaker comparisons). The automatic system discriminates much better: after logistic-regression calibration its Cllr is about 0.25, against about 0.7 for the formant system.

Source

`data-raw/forensic_comparison.R`

References

Brümmer, N. & du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech and Language*, 20(2-3), 230-275. doi:10.1016/j.csl.2005.08.001

Examples

```
cllr <- function(score, same) {
  m <- glm(same ~ score, family = binomial)
  log_lr <- predict(m) - log(mean(same) / (1 - mean(same)))
  0.5 * (mean(log2(1 + exp(-log_lr[same]))) + mean(log2(1 + exp(log_lr[!same]))))
}
same <- forensic_comparison$same_speaker == "yes"
c(automatic = cllr(forensic_comparison$score_auto, same),
  formant = cllr(forensic_comparison$score_formant, same))
boxplot(score_auto ~ same_speaker, data = forensic_comparison)
```

---

formants *Formants of three vowels from two speakers*

Description

F1 and F2 of /i a u/, 20 tokens per vowel from each of two speakers. Every formant of speaker B is 1.2 times the same formant of speaker A. This uniform scaling is a simplification. Between real speakers, and between women and men, the ratio differs across vowels and formants (Fant 1966), so normalisation methods that assume one factor per speaker work better here than on real data.

Usage

```
formants
```

Format

A tibble with 120 rows and 5 columns:

speaker `"A"` or `"B"`.

vowel `"i"`, `"a"` or `"u"`.

token Token number, 1 to 20.

f1, f2 First and second formant in Hz.

Details

True model: speaker A's mean F1 and F2 are 300 and 2300 Hz for /i/, 750 and 1300 Hz for /a/, and 320 and 800 Hz for /u/; speaker B's are 1.2 times higher. Token SD is 40 Hz for F1 and 120 Hz for F2.

Source

`data-raw/formants.R`

References

Fant, G. (1966). A note on vocal tract size factors and non-uniform F-pattern scalings. *Speech Transmission Laboratory Quarterly Progress and Status Report*, 7(4), 22-30.

Examples

```
aggregate(cbind(f1, f2) ~ speaker + vowel, data = formants, FUN = mean)
formants$f1_z <- ave(formants$f1, formants$speaker, FUN = function(x) c(scale(x)))
formants$f2_z <- ave(formants$f2, formants$speaker, FUN = function(x) c(scale(x)))
aggregate(cbind(f1_z, f2_z) ~ speaker + vowel, data = formants, FUN = mean)
```

---

freq_length *Reaction times with a frequency-by-length interaction*

Description

400 words whose frequency and length interact in a continuous-by-continuous interaction. The length effect is smaller for frequent words.

Usage

```
freq_length
```

Format

A tibble with 400 rows and 5 columns:

log_freq log10 word frequency, 1 to 6.

length Length in letters, 3 to 12.

lf_c `log_freq`, centred.

len_c `length`, centred.

rt Reaction time in ms.

Details

True model: `log(rt) = 6.63 - 0.04 * lf_c + 0.018 * len_c - 0.0065 * lf_c * len_c + e`, with `e` normal, SD 0.05.

Source

`data-raw/freq_length.R`

Examples

```
m <- lm(log(rt) ~ lf_c * len_c, data = freq_length)
coef(m)
```

`german_plurals` *A wug test of German noun plurals*

Description

60 adult native speakers of German give the plural of 48 nonce nouns, each presented with its article (*der*, *die*, *das*). The plural ending depends on gender and on how the noun ends: schwa-final and feminine nouns mostly take *-(e)n*, nouns in *-el*/*-er* mostly no ending, masculine and neuter nouns ending in a consonant mostly *-e*, and nouns ending in a full vowel *-s*. Speakers differ in how readily they use *-s*. A five-way outcome.

Usage

```
german_plurals
```

Format

A tibble with 2880 rows and 7 columns:

participant Participant ID, `P01` to `P60`.

item Item ID, `i01` to `i48`.

nonce The nonce noun, e.g. `"Pünde"`.

article `"der"`, `"die"` or `"das"`.

gender `"masculine"`, `"feminine"` or `"neuter"`.

final How the noun ends: `"schwa"`, `"-el/-er"`, `"consonant"` or `"full vowel"`.

plural `"-e"`, `"-(e)n"`, `"-er"`, `"-s"` or `"zero"`.

Details

True model: each gender-by-ending cell has its own probabilities for the five endings (see the data script), perturbed on the log scale by a by-participant preference for *-s* (SD 0.6) and by response-level noise (SD 0.3).

Source

`data-raw/german_plurals.R`

References

Berko, J. (1958). The child's learning of English morphology. *Word*, 14(2-3), 150-177. doi:10.1080/00437956.1958.11659661

Examples

```
round(prop.table(table(german_plurals$final, german_plurals$plural), 1), 2)
round(prop.table(table(german_plurals$gender, german_plurals$plural), 1), 2)
if (requireNamespace("nnet", quietly = TRUE)) {
  m <- nnet::multinom(plural ~ gender + final, data = german_plurals, trace = FALSE)
  round(coef(m), 2)
}
```

---

german_relatives *Self-paced reading of German relative clauses*

Description

48 participants read sentences word by word, such as *Der Lehrer, der den Schüler gesehen hat, lachte laut über den Witz* (subject relative) and *Der Lehrer, den der Schüler gesehen hat, ...* (object relative). Object relatives slow readers down at the end of the relative clause and on the main verb. 24 items, two counterbalanced lists, 12 words per sentence.

Usage

```
german_relatives
```

Format

A tibble with 13824 rows and 8 columns:

participant Participant ID, `P01` to `P48`.

list Counterbalancing list, 1 or 2.

item Item ID, `i01` to `i24`.

condition `"subject"` or `"object"` relative clause.

position Word position in the sentence, 1 to 12.

region Role of the word in the sentence, with levels in sentence order: determiner (`"det"`) and noun of the main clause, relative pronoun, then determiner, noun, participle and auxiliary of the relative clause (`"rc_det"` to `"rc_auxiliary"`), main verb, three spillover words and the final word. `levels(german_relatives$region)` gives the codes.

length Word length in letters.

rt Reading time in ms.

Details

True model:

```
log(rt) = 5.85 + p0 + i0 + 0.012 * (length - 5) - 0.008 * position
  + cost * object + e
```

The object-relative cost is 0.01 to 0.03 in the relative clause, 0.04 on the participle, 0.05 on the auxiliary and the main verb and 0.02 on the next word, with by-participant variation (SD 0.03). By-participant intercepts `p0` have SD 0.2 and by-item intercepts `i0` SD 0.04; residual SD 0.2. The cost on the auxiliary matches the neutral condition of Schriefers, Friederici and Kühn (1995, Experiment 1: 1350 against 1424 ms). Their relative clauses were ambiguous until the auxiliary and had no main clause verb after them, so the cost on the other words and all data are simulated.

Source

`data-raw/german_relatives.R`

References

Schriefers, H., Friederici, A. D. & Kühn, K. (1995). The processing of locally ambiguous relative clauses in German. *Journal of Memory and Language*, 34(4), 499-520. doi:10.1006/jmla.1995.1023

Examples

```
round(tapply(german_relatives$rt, german_relatives[c("region", "condition")], mean))
if (requireNamespace("lme4", quietly = TRUE)) {
  mv <- subset(german_relatives, region == "main_verb")
  m <- lme4::lmer(log(rt) ~ condition + (1 | participant) + (1 | item), data = mv)
  lme4::fixef(m)
}
```

---

hungarian_harmony *Hungarian vowel harmony in a wug test*

Description

50 Hungarian speakers choose between a back (*-nak*) and a front (*-nek*) dative suffix for 84 nonce stems. Stems with only back vowels take back suffixes and stems with front rounded vowels front ones. Stems with a back vowel followed by neutral vowels vary: the lower the last neutral vowel (*i* > *é* > *e*), the more front suffixes (the Height Effect), and two neutral vowels give more front suffixes than one (the Count Effect).

Usage

```
hungarian_harmony
```

Format

A tibble with 4200 rows and 6 columns:

participant Participant ID, `P01` to `P50`.

item Item ID, `i01` to `i84`.

stem_type Vowels of the stem: `"B"` (back), `"F"` (front rounded), `"N"` (one neutral vowel), `"Bi"`, `"Bé"`, `"Be"` (back vowel plus one neutral vowel) or `"BNN"` (back vowel plus two neutral vowels, the last one *e*).

n_neutral Number of neutral vowels after the last back vowel.

last_vowel Last vowel of the stem: `"back"`, `"front rounded"`, `"high"`, `"mid"` or `"low"`.

suffix `"back"` or `"front"`.

Details

True model: the probability of a back suffix is 0.99 (B), 0.01 (F), 0.06 (N), 0.95 (Bi), 0.38 (Bé), 0.07 (Be) and 0.02 (BNN), shifted on the logit scale by participant (SD 0.5) and item (SD 0.4) effects. These are the rounded proportions of back responses in the wug test of Hayes and Londe (2006, Table III); BNN uses their stems ending in *e*. Items are identified by their stem type; the nonce word forms are left out. Participants, items and responses are simulated.

Source

`data-raw/hungarian_harmony.R`

References

Hayes, B. & Londe, Z. C. (2006). Stochastic phonological knowledge: The case of Hungarian vowel harmony. *Phonology*, 23(1), 59-104. doi:10.1017/S0952675706000765

Hayes, B., Zuraw, K., Siptár, P. & Londe, Z. (2009). Natural and unnatural constraints in Hungarian vowel harmony. *Language*, 85(4), 822-863. doi:10.1353/lan.0.0169

Examples

```
round(prop.table(table(hungarian_harmony$stem_type, hungarian_harmony$suffix), 1), 2)
if (requireNamespace("lme4", quietly = TRUE)) {
  mixed <- subset(hungarian_harmony, is.element(stem_type, c("Bi", "Bé", "Be", "BNN")))
  m <- lme4::glmer(suffix ~ stem_type + (1 | participant) + (1 | item),
                   data = droplevels(mixed), family = binomial)
  lme4::fixef(m)
}
```

---

iterated_learning *Iterated learning of an artificial language*

Description

Eight transmission chains of ten learners pass on an artificial language that names 27 moving coloured shapes. Each learner is trained on 14 of the 27 word-picture pairs produced by the previous learner and then names all 27 pictures; the first learner is trained on a random language. In four chains the training pairs are passed on as they are. In the other four, words that name more than one picture are filtered out before training. Transmission error falls and structure rises across generations in both conditions. Without the filter, the number of distinct words collapses. For growth across generations, mixed models with few clusters and counts.

Usage

```
iterated_learning
```

Format

A tibble with 80 rows and 6 columns:

condition `"no filter"` or `"filter"`.

chain Chain ID: `A1` to `A4` without and `B1` to `B4` with the filter.

generation Learner's position in the chain, 1 to 10.

error Transmission error: normalised edit distance between this learner's words and the previous learner's, averaged over the 27 pictures.

words Number of distinct words the learner produced.

structure Structure: z-score of the correlation between the distances among words and the distances among meanings, against random pairings of words and meanings.

Details

True model: on the logit scale, transmission error starts at 0.80 (no filter) or 0.72 (filter) and falls by 0.55 or 0.20 per generation, with chain (SD 0.3) and residual (SD 0.3) variation. Structure rises by 0.6 (no filter) or 0.75 (filter) per generation, scaled by a chain effect (SD 0.15), with noise SD 1.2. The number of distinct words is 1 plus a Poisson count, capped at 27, whose mean starts near 20 and falls by 18% per generation without the filter and by 4% with it. The design follows Kirby, Cornish and Smith (2008), in whose two experiments transmission error fell by 0.75 and 0.43 and structure rose by 5.6 and 6.8 over ten generations. The chains are simulated.

Source

`data-raw/iterated_learning.R`

References

Kirby, S., Cornish, H. & Smith, K. (2008). Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31), 10681-10686. doi:10.1073/pnas.0707835105

Examples

```
aggregate(cbind(error, words, structure) ~ condition + generation,
          data = subset(iterated_learning, generation %in% c(1, 10)), FUN = mean)
interaction.plot(iterated_learning$generation, iterated_learning$chain,
                 iterated_learning$words, legend = FALSE, xlab = "Generation",
                 ylab = "Distinct words")
m <- glm(words ~ generation * condition, data = iterated_learning, family = poisson)
round(exp(coef(m)), 3)
if (requireNamespace("lme4", quietly = TRUE)) {
 m <- lme4::lmer(structure ~ generation * condition + (1 | chain), data = iterated_learning)
  round(lme4::fixef(m), 2)
}
```

`korean_stops` *The three-way stop contrast in Seoul Korean*

Description

Word-initial lenis, tense and aspirated stops from 20 older and 20 younger speakers of Seoul Korean: VOT and the f0 of the following vowel. Older speakers keep lenis and aspirated stops apart by VOT as well as by f0. Younger speakers produce aspirated stops with a shorter VOT, so lenis and aspirated VOT overlap and f0 alone keeps them apart. For cluster analysis and classification, and for a sound change in progress.

Usage

```
korean_stops
```

Format

A tibble with 1440 rows and 8 columns:

speaker `S01` to `S40`.

generation `"older"` or `"younger"`.

gender `"female"` or `"male"`.

stop `"lenis"`, `"tense"` or `"aspirated"`.

place `"labial"`, `"coronal"` or `"velar"`.

rep Repetition, 1 to 4.

vot Voice onset time in ms.

f0 f0 at the start of the following vowel in Hz.

Details

True model: mean VOT for tense, lenis and aspirated stops is 12, 60 and 94 ms for older and 12, 62 and 70 ms for younger speakers, 5 ms shorter for labials and 10 ms longer for velars; token variation is log-normal (SD 0.3 for tense, 0.18 otherwise) and by-speaker SD 0.1. f0 relative to the speaker's mean (about 200 Hz female, 115 Hz male) is +1, -3 and +2 semitones for tense, lenis and aspirated stops in both generations, token SD 1.5 semitones. The pattern follows Silva (2006): mean aspirated VOT fell from 94 ms in speakers born before 1965 to 70 ms in those born later, lenis and tense VOT stayed about the same, and the low f0 after lenis stops did not change with age. The other values and the data are simulated.

Source

`data-raw/korean_stops.R`

References

Silva, D. J. (2006). Acoustic evidence for the emergence of tonal contrast in contemporary Korean. *Phonology*, 23(2), 287-308. doi:10.1017/S0952675706000911

Examples

```
aggregate(cbind(vot, f0) ~ generation + stop, data = korean_stops, FUN = mean)
korean_stops$f0_st <- 12 * log2(korean_stops$f0 /
                                  ave(korean_stops$f0, korean_stops$speaker))
la <- subset(korean_stops, stop != "tense")
la$aspirated <- as.integer(la$stop == "aspirated")
accuracy <- function(cue, d) {
  m <- glm(reformulate(cue, "aspirated"), data = d, family = binomial)
  mean((fitted(m) > 0.5) == d$aspirated)
}
sapply(split(la, la$generation), function(d) c(vot = accuracy("vot", d),
                                               f0 = accuracy("f0_st", d)))
```

---

l2_frequency *Frequency effects in L1 and L2 speakers*

Description

Reaction times of L1 and L2 speakers to words of varying frequency. The frequency effect is stronger for L2 speakers, so frequency and group interact (a continuous-by-categorical interaction).

Usage

```
l2_frequency
```

Format

A tibble with 400 rows and 3 columns:

group `"L1"` or `"L2"`, 200 responses each.

log_freq log10 word frequency, 1 to 6.

rt Reaction time in ms.

Details

True model: `log(rt) = 6.56 - 0.04 * log_freq` for L1 and `6.85 - 0.07 * log_freq` for L2 speakers, plus normal noise with SD 0.06.

Source

`data-raw/l2_frequency.R`

Examples

```
m <- lm(log(rt) ~ log_freq * group, data = l2_frequency)
coef(m)
plot(log(rt) ~ log_freq, data = l2_frequency, col = group)
```

`language_attitudes` *Attitudes towards a regional dialect*

Description

320 respondents rate speakers of a regional dialect on 12 statements (1 = strongly disagree, 5 = strongly agree). The items measure three correlated attitudes: status, solidarity and vitality. Two items are negatively worded and need reverse-coding. Dialect speakers rate solidarity and vitality higher. About 2% of answers are missing. For factor analysis and scale construction.

Usage

```
language_attitudes
```

Format

A tibble with 320 rows and 15 columns:

respondent Respondent ID, `R001` to `R320`.

age Age in years.

dialect_speaker Whether the respondent speaks the dialect, `"no"` or `"yes"`.

intelligent, educated, competent, lazy Status items: dialect speakers sound intelligent / educated / competent / lazy. *lazy* is negatively worded.

friendly, warm, trustworthy, cold Solidarity items. *cold* is negatively worded.

used_by_young, will_survive, teach_school, identity Vitality items: the dialect is used by young people / will survive / should be taught at school / is important for local identity.

Details

True model: three latent attitudes with correlations 0.2 (status-solidarity), 0.3 (status-vitality) and 0.5 (solidarity-vitality). Dialect speakers score 0.6 SD higher on solidarity and 0.5 SD higher on vitality; vitality falls by 0.01 SD per year of age. Loadings are 0.55 to 0.8 (negative for *lazy* and *cold*); responses are the latent item scores cut at -1.3, -0.4, 0.4 and 1.3.

Source

`data-raw/language_attitudes.R`

Examples

```
la <- language_attitudes
la$lazy <- 6 - la$lazy
la$cold <- 6 - la$cold
items <- na.omit(la[, 4:15])
fa <- factanal(items, factors = 3, rotation = "promax")
print(fa$loadings, cutoff = 0.3)
aggregate(cbind(intelligent, friendly, will_survive) ~ dialect_speaker, data = la,
          FUN = mean)
```

`lexical_accuracy` *Accuracy in an Italian lexical decision task*

Description

30 native speakers of Italian decide whether each of 60 Italian words is a word; the outcome is whether the response was correct. For mixed-effects logistic regression.

Usage

```
lexical_accuracy
```

Format

A tibble with 1800 rows and 5 columns:

participant Participant ID, `P01` to `P30`.

word Word ID, `w01` to `w60`.

log_freq log10 word frequency, 1 to 6.

freq_c `log_freq`, centred on the mean of the 60 words.

correct 1 for a correct response, 0 for an error.

Details

True model, in log-odds: `1.5 + u0 + w0 + (0.8 + u1) * freq_c`. By-participant intercepts `u0` have SD 0.8 and slopes `u1` SD 0.3 (uncorrelated); by-word intercepts `w0` have SD 0.5.

Source

`data-raw/lexical_accuracy.R`

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::glmer(correct ~ freq_c + (1 + freq_c || participant) + (1 | word),
                   data = lexical_accuracy, family = binomial)
  summary(m)
}
```

`lexical_decision` *Lexical decision with participants and words*

Description

20 participants each respond to the same 40 words. Participants differ in speed and in the size of their frequency effect; words differ in difficulty. For crossed random effects.

Usage

```
lexical_decision
```

Format

A tibble with 800 rows and 5 columns:

participant Participant ID, `P01` to `P20`.

word Word ID, `w01` to `w40`.

log_freq log10 word frequency, 1 to 6.

freq_c `log_freq`, centred on the mean of the 40 words.

rt Reaction time in ms.

Details

True model: `log(rt) = 6.47 + u0 + w0 + (-0.045 + u1) * freq_c + e`. By-participant intercepts `u0` have SD 0.11 and slopes `u1` SD 0.025, correlated at -0.6; by-word intercepts `w0` have SD 0.045; residual SD 0.075. `languageR::lexdec` is a different data set, with real data.

Source

`data-raw/lexical_decision.R`

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(log(rt) ~ freq_c + (1 + freq_c | participant) + (1 | word),
                  data = lexical_decision)
  summary(m)
}
```

`lexical_decision_l2` *German lexical decision by L1 and L2 speakers*

Description

30 participants (15 L1 and 15 L2 speakers of German) each respond to the same 40 German words. L2 speakers are slower and more sensitive to frequency.

Usage

```
lexical_decision_l2
```

Format

A tibble with 1200 rows and 6 columns:

participant Participant ID, `P01` to `P30`.

group `"L1"` or `"L2"`.

word Word ID, `w01` to `w40`.

log_freq log10 word frequency, 1 to 6.

length Length in letters, 3 to 13; frequent words are shorter.

rt Reaction time in ms.

Details

True model:

```
log(rt) = 6.55 + 0.16 * L2 + p0 + w0 + slope * (log_freq - 3.5)
  + 0.017 * (length - 7) + e
slope = -0.035 - 0.03 * L2 + p1
```

By-participant intercepts `p0` have SD 0.085 and slopes `p1` SD 0.015; by-word intercepts `w0` have SD 0.04; residual SD 0.09.

Source

`data-raw/lexical_decision_l2.R`

Examples

```
lexical_decision_l2$freq_c <- lexical_decision_l2$log_freq - 3.5
lexical_decision_l2$length_c <- lexical_decision_l2$length - 7
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(log(rt) ~ group * freq_c + length_c + (1 + freq_c | participant) +
                    (1 | word), data = lexical_decision_l2)
  lme4::fixef(m)
}
```

`mandarin_tones` *f0 contours of the four Mandarin tones*

Description

12 speakers of Mandarin (6 female, 6 male) produce 8 syllables on each of the four lexical tones; f0 is measured at 11 equally spaced points through the vowel. The contours follow the textbook shapes: T1 high level (55), T2 rising (35), T3 low dipping (214) and T4 falling (51). For GAMMs of contour shape and for speaker normalisation.

Usage

```
mandarin_tones
```

Format

A tibble with 4224 rows and 8 columns:

speaker `S01` to `S12`.

gender `"female"` or `"male"`.

tone `"T1"` to `"T4"`.

syllable `syl1` to `syl8`.

token_id One level per token (`speaker.tone.syllable`).

duration Vowel duration in ms.

time Normalised time, 0 to 100 in steps of 10.

f0 f0 in Hz.

Details

True model: each tone follows a smooth curve through targets on Chao's 1-5 scale at 0, 25, 50, 75 and 100% of the vowel (T1: 5, 5, 5, 4.9, 4.8; T2: 3, 2.8, 3.3, 4.2, 5; T3: 2.2, 1.6, 1, 1.3, 2.5; T4: 5, 4.6, 3.7, 2.5, 1.2). One Chao level is about 2.5 semitones (by-speaker SD 0.3); level 1 is about 160 Hz for female and 90 Hz for male speakers. Token offsets have SD 0.5 semitones and residuals are AR(1) with rho 0.7. Mean durations are 250, 270, 300 and 220 ms for T1 to T4.

Source

`data-raw/mandarin_tones.R`

References

Chao, Y. R. (1930). A system of tone letters. *Le Maître Phonétique*, 45, 24-27.

Examples

```
mandarin_tones$st <- 12 * log2(mandarin_tones$f0 /
  ave(mandarin_tones$f0, mandarin_tones$speaker))
round(tapply(mandarin_tones$st, mandarin_tones[c("tone", "time")], mean), 1)
```

`megastudy` *Megastudy of 6000 words*

Description

Item-level norms for 6000 words, as in large lexical decision megastudies: mean RT and accuracy per word, with correlated lexical predictors. For multiple regression, collinearity, non-linear effects and variable importance.

Usage

```
megastudy
```

Format

A tibble with 6000 rows and 9 columns:

word Word ID, `w0001` to `w6000`.

log_freq log10 frequency per million words.

length Length in letters, 2 to 15.

syllables Number of syllables, 1 to 6.

neighbours Number of orthographic neighbours.

aoa Age of acquisition in years.

concreteness Concreteness rating, 1 to 5.

rt Mean lexical decision RT in ms.

accuracy Proportion of correct responses.

Details

True model:

```
rt = 825 - 120 * plogis(log_freq - 2.5) + 9 * (length - 7)
  + 8 * (syllables - 2.5) - 2 * neighbours + 10 * (aoa - 7)
  - 6 * (concreteness - 3) + e
accuracy = plogis(2.5 + 0.7 * (log_freq - 2.5) - 0.2 * (aoa - 7) + e)
```

The error `e` is normal with SD 35 ms for RT and SD 0.5 for accuracy. The frequency effect is S-shaped: it levels off for very rare and very frequent words. Frequent words are shorter, have more neighbours and are acquired earlier. Syllables correlate with length (r = 0.75): on its own, each syllable adds about 39 ms, but only 8 ms once length is in the model. Mean length (8 letters), mean number of syllables (2.6), mean RT (about 790 ms) and mean accuracy (0.86) are close to the English Lexicon Project words (Balota et al. 2007: 8 letters, 2.54 syllables, 784 ms, 0.84), but RTs vary less across words (SD 68 ms against 135 ms). In the English Lexicon Project, the number of syllables has an inhibitory effect of its own and the length effect is U-shaped (New et al. 2006); here the length effect is linear. The effects and data are simulated.

Source

data-raw/megastudy.R

References

Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., Neely, J. H., Nelson, D. L., Simpson, G. B. & Treiman, R. (2007). The English Lexicon Project. *Behavior Research Methods*, 39(3), 445-459. doi:10.3758/BF03193014

New, B., Ferrand, L., Pallier, C. & Brysbaert, M. (2006). Reexamining the word length effect in visual word recognition: New evidence from the English Lexicon Project. *Psychonomic Bulletin & Review*, 13(1), 45-52. doi:10.3758/BF03193811

Examples

```
round(cor(megastudy[c("log_freq", "length", "syllables", "neighbours", "aoa")]), 2)
coef(lm(rt ~ syllables, data = megastudy))
m1 <- lm(rt ~ log_freq + length + syllables + neighbours + aoa + concreteness,
         data = megastudy)
m2 <- lm(rt ~ splines::ns(log_freq, 4) + length + syllables + neighbours + aoa +
           concreteness, data = megastudy)
anova(m1, m2)
coef(m2)[c("length", "syllables", "neighbours", "aoa", "concreteness")]
```

---

messy_export *A messy experiment export*

Description

The raw export of a lexical decision experiment with 24 participants run in two labs, as it might come out of experiment software and a spreadsheet. The underlying data are clean (high- vs low-frequency words, as in rt_frequency); the export needs cleaning before analysis. All columns except `Item` are character, so this data set is the exception to the package's factor convention.

Usage

```
messy_export
```

Format

A tibble with 1086 rows and 10 columns:

Subject ID Participant ID, written in four ways (`P01`, `p1`, `P 01`, `01`).

Lab `"A"` or `"B"`.

Age Age, sometimes with `" years"`.

Block `"practice"` or `"test"`.

Item Item number, 1 to 40; practice items reuse 1 to 4.

Cond. Word frequency, spelled in eight ways (`high`, `High`, `HIGH` , `hi`, `low`, `Low`, `LOW`, `lo`), some with trailing spaces.

Response Time (ms) Reaction time, with a decimal comma in lab B, `"-999"` or `""` for missing values.

ACC Accuracy: `"1"`/`"0"` in lab A, `"correct"`/`"incorrect"` in lab B.

Timestamp Time of the trial: `2026-05-04 10:00:03` in lab A, `04.05.2026 10:00` in lab B.

Notes Always empty.

Details

Problems to fix: inconsistent IDs, condition labels and accuracy codes; a decimal comma; two codes for missing values; ages stored as text; two date formats; 96 practice trials; 30 duplicated test trials; an empty column; column names with spaces and punctuation. True model of the clean data: `log(rt) = 6.38 + 0.1 * low + p0 + w0 + e`, with participant (SD 0.12) and item (SD 0.05) effects and residual SD 0.15; accuracy is 96% for high- and 88% for low-frequency words.

Source

`data-raw/messy_export.R`

Examples

```
x <- messy_export
table(x$Cond.)
condition <- ifelse(substr(tolower(trimws(x$Cond.)), 1, 1) == "h", "high", "low")
rt <- as.numeric(sub(",", ".", x[["Response Time (ms)"]]))
rt[rt == -999] <- NA
id <- as.integer(gsub("[^0-9]", "", x[["Subject ID"]]))
test <- x$Block == "test" & !duplicated(data.frame(id, x$Item, x$Block))
tapply(rt[test], condition[test], mean, na.rm = TRUE)
```

---

n400 *Single-trial N400 amplitudes*

Description

30 participants read German sentences ending in an expected or unexpected word (*Er trinkt seinen Kaffee / Schrank*). For each trial, the mean EEG amplitude 300-500 ms after the final word at nine electrodes. The N400 is more negative the less predictable the word (lower cloze probability), and largest over centro-parietal sites. 80 items, two counterbalanced lists.

Usage

```
n400
```

Format

A tibble with 21600 rows and 9 columns:

participant Participant ID, `P01` to `P30`.

list Counterbalancing list, 1 or 2.

item Item ID, `i01` to `i80`.

condition `"expected"` or `"unexpected"`.

cloze Cloze probability of the final word.

electrode `"F3"`, `"Fz"`, `"F4"`, `"C3"`, `"Cz"`, `"C4"`, `"P3"`, `"Pz"` or `"P4"`.

anteriority `"frontal"`, `"central"` or `"parietal"`.

laterality `"left"`, `"midline"` or `"right"`.

amplitude Mean amplitude in microvolts.

Details

True model: `amplitude = 2 + p0 + i0 + e0 + weight * (-4 + p1) * (1 - cloze) + trial + e`. The electrode weight is 1 at Cz and Pz, 0.9 at the other central and parietal sites, 0.85 at Fz and 0.75 at F3 and F4; `e0` is a small electrode offset. By-participant intercepts `p0` (SD 1.5) and N400 slopes `p1` (SD 1); by-item intercepts `i0` (SD 0.5); trial noise shared across electrodes (SD 4) and electrode noise (SD 3). Cloze is 0.5 to 0.95 for expected and 0 to 0.05 for unexpected words. The midline weights follow the N400 difference waves in Kutas and Hillyard (1980, Experiment 2: -8.05, -9.62 and -9.24 microvolts at Fz, Cz and Pz); they recorded no lateral sites, so the lateral weights are simulated. In Kutas and Hillyard (1984), N400 amplitude tracked cloze probability closely at central and parietal sites (correlations of 0.90 to 0.92 for grand averages) and less at the frontal site (0.80). The sentences, participants and amplitudes here are simulated.

Source

`data-raw/n400.R`

References

Kutas, M. & Hillyard, S. A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science*, 207(4427), 203-205. doi:10.1126/science.7350657

Kutas, M. & Hillyard, S. A. (1984). Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307(5947), 161-163. doi:10.1038/307161a0

Examples

```
pz <- subset(n400, electrode == "Pz")
aggregate(amplitude ~ condition, data = pz, FUN = mean)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(amplitude ~ cloze + (1 + cloze | participant), data = pz)
  lme4::fixef(m)
}
round(tapply(n400$amplitude, n400[c("anteriority", "condition")], mean), 2)
```

`naming_pilot` *A pilot German picture-naming study for power analysis*

Description

15 younger and 15 older adults name the same 40 pictures in German. Naming is slower for pictures with less frequent names, more so for older adults. Meant as the pilot data from which to simulate a larger study.

Usage

```
naming_pilot
```

Format

A tibble with 1200 rows and 6 columns:

participant `P01` to `P30`.

age_group `"younger"` or `"older"`.

group `age_group` contrast-coded: -0.5 younger, 0.5 older.

picture `pic01` to `pic40`.

freq_c Centred log10 frequency of the picture name, about -2.5 to 2.5.

latency Naming latency in ms.

Details

True model:

```
log(latency) = 6.75 + 0.15 * group + p0 + w0
  + (-0.04 - 0.02 * group + p1) * freq_c + e
```

By-participant intercepts `p0` have SD 0.08 and slopes `p1` SD 0.015; by-picture intercepts `w0` have SD 0.04; residual SD 0.1.

Source

`data-raw/naming_pilot.R`

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(log(latency) ~ age_group * freq_c + (1 + freq_c | participant) +
                    (1 | picture), data = naming_pilot)
  lme4::fixef(m)
}
```

`overregularization` *Past-tense overregularization*

Description

Overregularization errors (*goed*, *holded*) out of all irregular past-tense contexts, for 24 children recorded every six months from 30 to 60 months. The simulation follows the account of Marcus et al. (1992): errors are rare, their rate does not change with age, and children differ in how often they make them. Maratsos (2000) disputed the low rates, arguing that averages over all verbs hide much higher rates for verbs a child uses rarely and periods in which a child overregularizes often. These data have no verb-level information. For binomial models of errors out of attempts and for testing an effect that is absent.

Usage

```
overregularization
```

Format

A tibble with 144 rows and 4 columns:

child Child ID, `C01` to `C24`.

age Age in months.

attempts Irregular past-tense contexts in the recording.

errors Overregularizations among them.

Details

True model, in log-odds: `-3.66`, with by-child intercepts of SD 0.9, so the median child makes errors in 2.5% of attempts and a few children in many more. Age has no effect. Children attempt more irregular past tenses as they get older. Marcus et al. (1992) report a median rate of 2.5% and a mean of 4.2% across 25 children, with a roughly constant rate from age 2 into the school years. Maratsos (2000) showed that pooling all tokens gives most weight to frequent verbs, whose rates are lowest. The children and counts are simulated.

Source

`data-raw/overregularization.R`

References

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T. J. & Xu, F. (1992). Overregularization in language acquisition. *Monographs of the Society for Research in Child Development*, 57(4). doi:10.2307/1166115

Maratsos, M. (2000). More overregularizations after all: New data and discussion on Marcus, Pinker, Ullman, Hollander, Rosen & Xu. *Journal of Child Language*, 27(1), 183-212. doi:10.1017/S0305000999004067

Examples

```
sum(overregularization$errors) / sum(overregularization$attempts)
aggregate(cbind(errors, attempts) ~ age, data = overregularization, FUN = sum)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::glmer(cbind(errors, attempts - errors) ~ I((age - 45) / 12) + (1 | child),
                   data = overregularization, family = binomial)
  summary(m)$coefficients
}
```

`passives` *Passives in four registers*

Description

Number of passive constructions in 400 texts, 100 each from conversation, fiction, news and academic writing. Texts differ in length, so counts need an exposure term. The counts are overdispersed.

Usage

```
passives
```

Format

A tibble with 400 rows and 4 columns:

text Text ID, `T001` to `T400`.

register `"conversation"`, `"fiction"`, `"news"` or `"academic"`.

words Text length in words.

passives Number of passives.

Details

True model: negative binomial with mean `rate * words / 1000` and size 4, where `rate` per 1000 words is 2 (conversation), 5 (fiction), 10 (news) and 20 (academic). Biber, Conrad and Leech (2002) note that passive verbs are about ten times more frequent in academic prose than in conversation. The other rates and all data are simulated.

Source

`data-raw/passives.R`

References

Biber, D., Conrad, S. & Leech, G. (2002). *Longman Student Grammar of Spoken and Written English*. Harlow: Longman.

Examples

```
if (requireNamespace("MASS", quietly = TRUE)) {
  m <- MASS::glm.nb(passives ~ 0 + register + offset(log(words / 1000)), data = passives)
  exp(coef(m))
  m$theta
}
```

`perception_training` *Perceptual training of a non-native contrast*

Description

18 learners identify a non-native consonant contrast in 40 trials before and after a week of perceptual training. A small paired design in wide format.

Usage

```
perception_training
```

Format

A tibble with 18 rows and 3 columns:

learner Learner ID, `L01` to `L18`.

pre Correct responses out of 40 before training.

post Correct responses out of 40 after training.

Details

True model: binomial out of 40, with log-odds `0.4` + `ability` before and `0.4` + `ability` + `gain` after training. `ability` has SD 0.6; `gain` has mean 0.6 and SD 0.35.

Source

`data-raw/perception_training.R`

Examples

```
t.test(perception_training$post, perception_training$pre, paired = TRUE)
t.test(perception_training$post, perception_training$pre)
```

price_f2 *F2 trajectories of the PRICE diphthong*

Description

F2 over the course of the PRICE vowel (as in *price*), 10 tokens from each of 20 speakers: 10 L1 English and 10 L1 Italian speakers of English. Italian speakers start higher, end lower and move earlier and more gradually. For GAMs and GAMMs.

Usage

```
price_f2
```

Format

A tibble with 2200 rows and 6 columns:

speaker `S01` to `S20`.

l1 `"English"` or `"Italian"`.

traj One level per token (`speaker.token`).

duration Vowel duration in ms, 120 to 300.

time Normalised time, 0 to 100 in steps of 10.

f2 F2 in Hz.

Details

True model: a logistic rise from an onset (1250 Hz English, 1320 Hz Italian) to an offset (2050 / 1900 Hz, lowered by 2 Hz per ms that the vowel is shorter than 250 ms), with its midpoint at 62 / 50 and width 8 / 14. By-speaker offsets (SD 60 Hz) and midpoints (SD 5) and by-token offsets (SD 40 Hz); residuals are AR(1) with rho 0.6 and SD 30 Hz.

Source

`data-raw/price_f2.R`

Examples

```
if (requireNamespace("mgcv", quietly = TRUE)) {
  m <- mgcv::bam(f2 ~ l1 + s(time, by = l1) + s(time, speaker, bs = "fs", m = 1),
                 data = price_f2)
  summary(m)
  new <- expand.grid(time = c(0, 50, 100), l1 = c("English", "Italian"), speaker = "S01")
  cbind(new, f2 = predict(m, new, exclude = "s(time,speaker)"))
}
```

`pupil_noise` *Pupil dilation when listening in quiet and in noise*

Description

Pupil size over 3 seconds after stimulus onset, 12 trials per condition from each of 24 participants. Listening in noise leads to a larger and later peak. For GAMMs with autocorrelated residuals.

Usage

```
pupil_noise
```

Format

A tibble with 17856 rows and 6 columns:

participant `P01` to `P24`.

condition `"quiet"` or `"noise"`.

trial Trial number within condition, 1 to 12.

trial_id One level per trial (`participant.condition.trial`).

time Time from stimulus onset in ms, 0 to 3000 in steps of 100.

pupil Baseline-corrected pupil size, arbitrary units.

Details

True model: a gamma-shaped response peaking at 180 units after 1200 ms in quiet, and 90 units higher and 250 ms later in noise. The peak shrinks by 4 units per trial (habituation). By-participant variation in peak (SD 40), latency (SD 150 ms) and noise effect (SD 30); by-trial variation in peak (SD 40) and baseline (SD 15); residuals are AR(1) with rho 0.8 and innovation SD 12, so their SD is 20 (12 / sqrt(1 - 0.8^2)).

Source

`data-raw/pupil_noise.R`

Examples

```
curve <- aggregate(pupil ~ condition + time, data = pupil_noise, FUN = mean)
plot(pupil ~ time, data = curve, col = condition)
peaks <- do.call(rbind, lapply(split(curve, curve$condition),
                               function(d) d[which.max(d$pupil), ]))
peaks
```

`reading_skipping` *Word skipping and fixation times in reading*

Description

28 readers read 54 sentences while their eye movements are recorded. Each sentence has a target word that is short (4-6 letters), medium (7-9) or long (10-12), and the context makes it either predictable or unpredictable. Each reader sees each sentence once, in one of the two conditions. Short and predictable words are skipped more often. When a word is fixated, long and unpredictable words are looked at for longer. Length and predictability do not interact. Fixation times are missing for skipped words. For mixed logistic regression of skipping, mixed models of log durations and data missing by design.

Usage

```
reading_skipping
```

Format

A tibble with 1512 rows and 9 columns:

reader Reader ID, `R01` to `R28`.

item Sentence, `I01` to `I54`.

length Length group of the target word: `"short"`, `"medium"` or `"long"`.

letters Length of the target word in letters.

predictability `"unpredictable"` or `"predictable"`.

cloze Cloze probability of the target word in this context.

skipped 1 if the target was skipped on the first pass, 0 if not.

first_fixation Duration of the first fixation on the target in ms. `NA` if skipped.

gaze_duration Sum of all first-pass fixations on the target in ms. `NA` if skipped.

Details

True model: cell means are those in Rayner et al. (2011, Table 3). Skipping rates are 36, 22 and 18% for predictable and 28, 18 and 10% for unpredictable short, medium and long words. First fixations last 193 to 211 ms and gaze durations 211 to 246 ms. Skipping is logistic, with by-reader (SD 0.6) and by-item (SD 0.4) effects. First fixations are log-normal, with by-reader (SD 0.12), by-item (SD 0.06) and residual (SD 0.25) variation. A word is refixated with probability `(gaze - first fixation) / 180` for its cell, and a refixation adds about 180 ms. Cloze values are drawn around 0.74 for predictable and 0.15 for unpredictable words, close to the norms in the same paper. Readers, items and eye movements are simulated.

Source

`data-raw/reading_skipping.R`

References

Rayner, K., Slattery, T. J., Drieghe, D. & Liversedge, S. P. (2011). Eye movements and word skipping during reading: Effects of word length and predictability. *Journal of Experimental Psychology: Human Perception and Performance*, 37(2), 514-528. doi:10.1037/a0020990

Examples

```
aggregate(skipped ~ length + predictability, data = reading_skipping, FUN = mean)
aggregate(gaze_duration ~ length + predictability, data = reading_skipping, FUN = mean)
colSums(is.na(reading_skipping))
if (requireNamespace("lme4", quietly = TRUE)) {
  m1 <- lme4::glmer(skipped ~ length * predictability + (1 | reader) + (1 | item),
                    data = reading_skipping, family = binomial)
  m0 <- update(m1, . ~ . - length:predictability)
  print(anova(m0, m1))
 m <- lme4::lmer(log(gaze_duration) ~ length + predictability + (1 | reader) + (1 | item),
                 data = reading_skipping)
  round(lme4::fixef(m), 3)
}
```

---

relativizers *Relativizer choice: that, which or zero*

Description

The relativizer in restrictive relative clauses (*the book that / which / 0 I read*), 10 tokens from each of 240 texts, half spoken, half written. A three-way outcome; zero is impossible in subject relatives.

Usage

```
relativizers
```

Format

A tibble with 2400 rows and 6 columns:

text Text ID, `T001` to `T240`.

mode `"spoken"` or `"written"`.

token Token number within the text, 1 to 10.

role Role of the relativised noun in the clause, `"subject"` or `"object"`.

antecedent_length Length of the antecedent noun phrase in words.

relativizer `"that"`, `"which"` or `"zero"`.

Details

True model: multinomial logit with "that" as reference.

```
which: -1.6 + 1.4 * written + 0.25 * (antecedent_length - 2)
zero: 0.6 - 1.2 * written - 0.4 * (antecedent_length - 2)
```

The zero relativizer occurs only in object relatives. By-text intercepts with SD 0.5 for each.

Source

`data-raw/relativizers.R`

Examples

```
table(relativizers$role, relativizers$relativizer)
if (requireNamespace("nnet", quietly = TRUE)) {
  obj <- subset(relativizers, role == "object")
  obj$length_c <- obj$antecedent_length - 2
  m <- nnet::multinom(relativizer ~ mode + length_c, data = obj, trace = FALSE)
  round(coef(m), 2)
}
```

---

rhythm_metrics *Rhythm metrics across languages, speakers and speech materials*

Description

%V, ΔC and VarcoV from 8 speakers of each of six languages: English, German, Greek, Italian, Korean and Spanish. Ramus et al. (1999) proposed %V and ΔC to sort languages into rhythm classes. Each speaker produced five samples of five kinds of material: uncontrolled sentences, sentences built from complex syllables, sentences built from simple syllables, a read story and spontaneous speech. The design follows Arvaniti (2012). The languages differ, but not in the way rhythm classes predict, and speakers, elicitation method and syllable complexity move the metrics about as much as language does. ΔC also falls as speech rate rises. For partitioning variance with mixed models.

Usage

```
rhythm_metrics
```

Format

A tibble with 1200 rows and 8 columns:

language `"English"`, `"German"`, `"Greek"`, `"Italian"`, `"Korean"` or `"Spanish"`.

speaker Speaker ID: ISO 639-3 language code and number, e.g. `eng_01`.

material Kind of speech material. The three read sentence sets are `"uncontrolled sentences"`, `"complex-syllable sentences"` and `"simple-syllable sentences"`; the other two levels are `"story"` and `"spontaneous"`.

item Sample number within speaker and material, 1 to 5.

rate Speech rate in syllables per second.

percent_v Percentage of the sample that is vocalic.

delta_c Standard deviation of consonantal intervals in ms.

varco_v Coefficient of variation of vocalic intervals.

Details

True model: each metric is the sum of a language mean, a material effect, a language-by-material deviation, a speaker effect and noise. Language means for %V, ΔC and VarcoV are 43, 51, 55 (English), 41.5, 51, 52 (German), 44.5, 46, 56 (Greek), 46, 46.5, 55 (Italian), 46, 49, 57.5 (Korean) and 46, 48.5, 54 (Spanish). They follow the order of the language differences Arvaniti (2012) found: for %V, German < English < Greek < Italian, Korean and Spanish; for ΔC, English and German > Korean and Spanish > Greek and Italian; for VarcoV, German lowest and Korean highest. Material effects, relative to the uncontrolled sentences, are -3, +7, +3 (complex-syllable sentences), +3, -6, -4 (simple-syllable sentences), -1, +2, +2 (story) and -1.5, +6, +7 (spontaneous). Language-by-material deviations have SD 1, 3 and 3, speaker effects SD 2, 4 and 4, and residuals SD 2, 5 and 6. ΔC falls by 6 ms per syllable per second, a rate effect described by Dellwo (2006). All values were chosen for the simulation to reproduce these patterns; they are not measurements from either study.

Source

`data-raw/rhythm_metrics.R`

References

Arvaniti, A. (2012). The usefulness of metrics in the quantification of speech rhythm. *Journal of Phonetics*, 40(3), 351-373. doi:10.1016/j.wocn.2012.02.003

Dellwo, V. (2006). Rhythm and speech rate: A variation coefficient for deltaC. In P. Karnowski & I. Szigeti (eds.), *Language and Language-Processing: Proceedings of the 38th Linguistics Colloquium, Piliscsaba 2003* (pp. 231-241). Frankfurt am Main: Peter Lang.

Rathcke, T. & Smith, R. (2015). Speech timing and linguistic rhythm: On the acoustic bases of rhythm typologies. *Journal of the Acoustical Society of America*, 137(5), 2834-2845. doi:10.1121/1.4919322

Ramus, F., Nespor, M. & Mehler, J. (1999). Correlates of linguistic rhythm in the speech signal. *Cognition*, 73(3), 265-292. doi:10.1016/S00100277(99)00058X

White, L. & Mattys, S. L. (2007). Calibrating rhythm: First language and second language studies. *Journal of Phonetics*, 35(4), 501-522. doi:10.1016/j.wocn.2007.02.003

Examples

```
round(tapply(rhythm_metrics$percent_v, rhythm_metrics[c("material", "language")], mean),
      1)
boxplot(percent_v ~ material, data = rhythm_metrics)
```

```
coef(lm(delta_c ~ rate + language + material, data = rhythm_metrics))["rate"]
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(percent_v ~ language + material + (1 | speaker), data = rhythm_metrics)
  round(lme4::fixef(m), 2)
  lme4::VarCorr(m)
}
```

`rt_frequency` *Reaction times to high- and low-frequency words*

Description

A small lexical decision task: 10 participants each respond to the same 20 words, 10 of high and 10 of low frequency.

Usage

```
rt_frequency
```

Format

A tibble with 200 rows and 5 columns:

participant Participant ID, `P01` to `P10`.

word Word ID, `w01` to `w20`.

frequency Word frequency, `"high"` or `"low"`.

rt Reaction time in ms.

correct 1 for a correct response, 0 for an error.

Details

True model: `log(rt) = 6.38 + 0.1 * low + p0 + w0 + e`, with by-participant intercepts `p0` (SD 0.12), by-word intercepts `w0` (SD 0.06) and residual SD 0.15, so low-frequency words take about 10% longer. Accuracy is 95% for high- and 85% for low-frequency words.

Source

`data-raw/rt_frequency.R`

Examples

```
aggregate(cbind(rt, correct) ~ frequency, data = rt_frequency, FUN = mean)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(log(rt) ~ frequency + (1 | participant) + (1 | word), data = rt_frequency)
  exp(lme4::fixef(m))
}
```

`scalar_implicature` *Scalar implicature: truth-value judgments*

Description

Participants judge *Some of the cards have stars* when all, some or none of the cards have stars. In the all-context the sentence is literally true but under-informative: a "false" response is pragmatic, a "true" response logical. Participants differ: some always answer one way, most mix the two. 60 participants, 24 items, three counterbalanced lists.

Usage

```
scalar_implicature
```

Format

A tibble with 1440 rows and 6 columns:

participant Participant ID, `P01` to `P60`.

list Counterbalancing list, 1 to 3.

item Item ID, `i01` to `i24`.

context How many cards have stars: `"some"` (true control), `"all"` (critical) or `"none"` (false control).

response `"true"` or `"false"`.

rt Response time in ms.

Details

True model: in the all-context, each participant answers "false" with their own probability, drawn from Beta(1.1, 0.7) (mean 0.61). Controls are answered correctly 96% of the time. Reaction times follow

```
log(rt) = 7.25 + 0.22 * pragmatic
```

where `pragmatic` marks a "false" response in the all-context, with by-participant (SD 0.15) and by-item (SD 0.05) intercepts and residual SD 0.25. The proportions follow Experiment 3 of Bott and Noveck (2004): 61% pragmatic responses, 9 of 32 participants who always gave the same response, 87% to 98% correct on controls, slower pragmatic responses and logical responses about as fast as controls. The effect sizes and data are simulated.

Source

`data-raw/scalar_implicature.R`

References

Bott, L. & Noveck, I. A. (2004). Some utterances are underinformative: The onset and time course of scalar inferences. *Journal of Memory and Language*, 51(3), 437-457. doi:10.1016/j.jml.2004.05.006

Examples

```
all_ctx <- subset(scalar_implicature, context == "all")
p_false <- tapply(all_ctx$response == "false", all_ctx$participant, mean)
hist(p_false, breaks = 10)
controls <- droplevels(subset(scalar_implicature, context != "all"))
tapply(controls$response == "true", controls$context, mean)
aggregate(rt ~ response, data = all_ctx, FUN = median)
```

---

simling_datasets *List the data sets in simling*

Description

Returns one row per data set, with its branch of linguistics, its language, its size, its response variable and the statistical topic it was built for. Filter it like any other tibble to find a data set for a lesson.

Usage

```
simling_datasets()
```

Value

A tibble with one row per data set and the columns:

name Name of the data set, as passed to `data()` or `?`.

branch Branch of linguistics, a factor in the order used on the package website.

language Language or languages the data simulate. `NA` for data sets without a particular language.

rows Number of rows.

columns Number of columns.

response Type of response variable.

topic Statistical topic the data set was built for.

Examples

```
simling_datasets()

sets <- simling_datasets()
sets[sets$branch == "Psycholinguistics", ]
sets[grepl("counts", sets$response), c("name", "rows", "topic")]
table(sets$branch)
```

speech_styles *Vowel duration in three speaking styles*

Description

50 vowels each in casual, clear and loud speech.

Usage

```
speech_styles
```

Format

A tibble with 150 rows and 2 columns:

style `"casual"`, `"clear"` or `"loud"`.

duration Vowel duration in ms.

Details

True model: mean 110 ms (SD 18) in casual, 150 ms (SD 20) in clear and 132 ms (SD 19) in loud speech.

Source

`data-raw/speech_styles.R`

Examples

```
m <- lm(duration ~ style, data = speech_styles)
anova(m)
coef(m)
pairwise.t.test(speech_styles$duration, speech_styles$style)
```

speech_to_song *The speech-to-song illusion*

Description

86 English listeners each hear six spoken sentences, each repeated eight times. Before the repetitions they rate how song-like the sentence sounds (1 = clearly speech, 8 = clearly song). During the repetitions they press a button as soon as the speech turns into song, and afterwards they rate the sentence again. Each sentence comes in a baseline version, a version with flattened pitch within syllables and a version with lengthened sonorous parts. Both manipulations make the illusion more likely, while musical aptitude does not. Listeners with weaker beat perception hear the song earlier, and listeners with better melody perception rate it as more song-like. For mixed logistic and ordinal models, and for a response that is missing by design.

Usage

```
speech_to_song
```

Format

A tibble with 516 rows and 9 columns:

listener Listener ID, `L01` to `L86`.

bat Beat perception score (z-score), as from the Computerised Adaptive Beat Alignment Test.

mpt Melody perception score (z-score), as from the Mistuning Perception Test.

sentence Sentence, `S1` to `S9`.

manipulation Version of the sentence: `"baseline"`, `"pitch"` (pitch movements within syllables flattened) or `"duration"` (sonorous parts lengthened by 20% and obstruents shortened by the same amount).

rating_before Rating before the repetitions, 1 (clearly speech) to 8 (clearly song).

transformed 1 if the listener reported the illusion, 0 if not.

cycle Repetition, 1 to 8, during which the listener reported the illusion. `NA` when there was no illusion.

rating_after Rating after the repetitions, on the same scale.

Details

True model: each listener hears two sentences in each condition, assigned in a Latin square. `transformed` is Bernoulli with logit `-0.9 + 1.04 * pitch + 0.78 * duration` plus listener (SD 1) and sentence (SD 0.5) effects. `cycle` and both ratings come from ordered probit models. For `cycle` the latent value is `0.61 * bat` plus a listener effect (SD 0.6). For `rating_after` it is `0.2 + 1.2 * pitch + 0.91 * duration + 0.44 * mpt + 0.8 * transformed` plus listener (SD 0.7) and sentence effects. The two aptitude scores correlate at about 0.5. The design and the effects of the manipulations and of aptitude follow Rathcke and Smit (2025); in their data the manipulated stimuli transformed about half of the time. The other intercepts, the effect of `transformed` on the final rating and all data are simulated.

Source

`data-raw/speech_to_song.R`

References

Rathcke, T. & Smit, E. A. (2025). Musical aptitude moderates ease and vividness but not frequency of the speech-to-song illusion. *Scientific Reports*, 15, 33315. doi:10.1038/s41598025186128

Falk, S., Rathcke, T. & Dalla Bella, S. (2014). When speech sounds like music. *Journal of Experimental Psychology: Human Perception and Performance*, 40(4), 1491-1506. doi:10.1037/a0036858

Rathcke, T., Falk, S. & Dalla Bella, S. (2024). Why does speech sometimes sound like song? Exploring the role of music-related priors in the "speech-to-song illusion". *Music & Science*, 7. doi:10.1177/20592043241266060

Examples

```
round(tapply(speech_to_song$transformed, speech_to_song$manipulation, mean), 2)
table(speech_to_song$cycle)
wilcox.test(speech_to_song$rating_after, speech_to_song$rating_before, paired = TRUE)
if (requireNamespace("lme4", quietly = TRUE)) {
 m <- lme4::glmer(transformed ~ manipulation + bat + mpt + (1 | listener) + (1 | sentence),
                  data = speech_to_song, family = binomial)
 round(lme4::fixef(m), 2)
}
if (requireNamespace("ordinal", quietly = TRUE)) {
 m <- ordinal::clm(factor(rating_after) ~ manipulation + bat + mpt, data = speech_to_song,
                   link = "probit")
 print(round(coef(m)[c("manipulationpitch", "manipulationduration", "bat", "mpt")], 2))
 m <- ordinal::clm(factor(cycle) ~ bat + mpt, data = speech_to_song, link = "probit")
 print(round(coef(m)[c("bat", "mpt")], 2))
}
```

---

style_voicing *Vowel duration by speaking style and voicing*

Description

A 2 x 2 design: casual or clear speech, voiceless or voiced following consonant, 40 tokens per cell. The voicing effect is larger in clear speech, which makes this a categorical-by-categorical interaction.

Usage

```
style_voicing
```

Format

A tibble with 160 rows and 3 columns:

style `"casual"` or `"clear"`.

voicing `"voiceless"` or `"voiced"`.

duration Vowel duration in ms.

Details

True model: cell means are 110 (casual, voiceless), 145 (casual, voiced), 125 (clear, voiceless) and 180 ms (clear, voiced); SD 18 ms.

Source

`data-raw/style_voicing.R`

Examples

```
m <- lm(duration ~ style * voicing, data = style_voicing)
coef(m)
with(style_voicing, interaction.plot(voicing, style, duration))
```

`td_deletion` */t, d/ deletion*

Description

Whether word-final /t, d/ is deleted (as in *wes' side*), 50 tokens from each of 40 speakers. Includes an interaction, a non-linear age effect, correlated predictors and two pure-noise predictors. For classification trees and random forests.

Usage

```
td_deletion
```

Format

A tibble with 2000 rows and 11 columns:

speaker `S01` to `S40`.

deleted `"retained"` or `"deleted"`.

following Following segment: `"vowel"`, `"pause"` or `"consonant"`.

morph `"monomorpheme"` (*mist*) or `"past"` (*missed*).

rate Speech rate in syllables per second.

log_freq log10 word frequency, 1 to 6.

age Speaker age, 18 to 80.

gender `"female"` or `"male"`; no effect.

education `"school"` or `"university"`; correlated with age but no effect of its own.

noise_cont, noise_bin Random noise, continuous and binary; no effect.

Details

True model, in log-odds of deletion: `-0.8 + s0 + 1.8 * consonant + 0.4 * pause - 1 * past + 0.9 * (rate - 5) * consonant + 1.2 * plogis((45 - age) / 5) + 0.25 * (log_freq - 3.5)`, with by-speaker intercepts `s0` of SD 0.5. Rate matters only before consonants; deletion drops sharply around age 45.

Source

`data-raw/td_deletion.R`

Examples

```
if (requireNamespace("rpart", quietly = TRUE)) {
  tree <- rpart::rpart(deleted ~ following + morph + rate + log_freq + age + gender +
                         education + noise_cont + noise_bin, data = td_deletion)
  round(tree$variable.importance)
}
if (requireNamespace("randomForest", quietly = TRUE)) {
  set.seed(1)
  rf <- randomForest::randomForest(deleted ~ following + morph + rate + log_freq + age +
                                     gender + education + noise_cont + noise_bin,
                                   data = td_deletion, importance = TRUE)
  round(randomForest::importance(rf, type = 1), 1)
}
```

---

tobi_agreement *Two coders' GToBI labels*

Description

Two trained coders label the same 456 words from 40 German utterances with GToBI pitch accents and phrase boundaries: 20 utterances of read and 20 of spontaneous speech. Coders agree on whether a word is accented for about 86% of words and on the exact accent label for about 71%, close to the agreement Grice et al. (1996) report for GToBI. For English ToBI, Pitrelli et al. (1994) report 88% and 81%. In this simulation, accent agreement is lower in spontaneous speech. For percentage agreement and Cohen's kappa.

Usage

```
tobi_agreement
```

Format

A tibble with 456 rows and 8 columns:

utterance Utterance ID, `U01` to `U40`.

style `"read"` or `"spontaneous"`.

word Word ID (`utterance_wNN`).

position Position of the word in the utterance.

accent_1, accent_2 Pitch accent labelled by coder 1 and coder 2: `"none"`, `"H*"`, `"L+H*"`, `"L*"`, `"L*+H"`, `"H+!H*"` or `"H+L*"`.

boundary_1, boundary_2 Boundary after the word labelled by coder 1 and coder 2: `"none"`, `"intermediate"` or `"intonation phrase"`.

Details

True model: each word has a true accent (55% none) and boundary (always an intonation phrase boundary at the end of the utterance, 12% intermediate boundaries elsewhere). Each coder gives the true label with probability `plogis(base - difficulty)`, where `base` is 2.6 for unaccented and 1.6 for accented words (0.2 lower for coder 2) and the word's difficulty has mean 0 in read and 0.5 in spontaneous speech (SD 0.5). Otherwise the coder picks a similar label (e.g. `"L+H*"` for `"H*"`).

Source

`data-raw/tobi_agreement.R`

References

Grice, M., Reyelt, M., Benzmüller, R., Mayer, J. & Batliner, A. (1996). Consistency in transcription and labelling of German intonation with GToBI. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP 96)* (pp. 1716-1719). Philadelphia. doi:10.1109/ICSLP.1996.607958

Grice, M., Baumann, S. & Benzmüller, R. (2005). German intonation in autosegmental-metrical phonology. In S.-A. Jun (ed.), *Prosodic Typology: The Phonology of Intonation and Phrasing* (pp. 55-83). Oxford: Oxford University Press. doi:10.1093/acprof:oso/9780199249633.003.0003

Pitrelli, J. F., Beckman, M. E. & Hirschberg, J. (1994). Evaluation of prosodic transcription labeling reliability in the ToBI framework. *Proceedings of ICSLP 1994*. https://www.isca-archive.org/icslp_1994/pitrelli94_icslp.html

Examples

```
kappa <- function(a, b) {
  tab <- table(a, b)
  po <- sum(diag(tab)) / sum(tab)
  pe <- sum(rowSums(tab) * colSums(tab)) / sum(tab)^2
  c(agreement = po, kappa = (po - pe) / (1 - pe))
}
with(tobi_agreement, kappa(accent_1 != "none", accent_2 != "none"))
both <- subset(tobi_agreement, accent_1 != "none" & accent_2 != "none")
kappa(both$accent_1, both$accent_2)
sapply(split(tobi_agreement, tobi_agreement$style),
       function(d) kappa(d$accent_1 != "none", d$accent_2 != "none"))
```

---

turn_taking *Turn-taking in 10 languages*

Description

The gap between the end of a yes/no question and the start of its answer, in 25 question-answer pairs from each of 12 conversations in each of 10 languages. Negative values are overlaps. In every language, gaps cluster just above zero and are skewed to the right; languages differ in their average gap by a few hundred milliseconds. Disconfirmations and non-answers come later; answers to a questioner who is looking at the responder come sooner.

Usage

```
turn_taking
```

Format

A tibble with 3000 rows and 6 columns:

language Language, one of `"Danish"`, `"Dutch"`, `"English"`, `"Italian"`, `"Japanese"`, `"Korean"`, `"Lao"`, `"Tzeltal"`, `"Yélî Dnye"` or `"\u01c2\u0100khoe Hai\u01c1om"`.

conversation Conversation ID: a three-letter code for the language and a number, from `akh_01` to `yel_12`; 12 per language.

response `"confirmation"` or `"disconfirmation"`.

answer `"answer"` or `"non-answer"` (e.g. *I don't know*).

gaze Whether the questioner was looking at the responder, `"no"` or `"yes"`.

offset Time from the end of the question to the start of the answer in ms.

Details

True model:

```
offset = baseline + c0 + 207 * disconfirmation + 132 * non_answer
  - 69 * gaze + e
```

`c0` is a by-conversation offset (SD 120 ms) and `e` is ex-Gaussian (a normal with SD 200 ms plus an exponential with mean 150 ms, centred). The three effects and the size of the conversation variation follow the model in Stivers et al. (2009, Table 1). Baselines are set so that each language's expected mean equals its reported mean: 7 ms for Japanese, 310 ms for Italian and 469 ms for Danish (Stivers et al. 2009), and the others from Levinson and Torreira (2015, Table 1). The conversations and answers are simulated.

Source

`data-raw/turn_taking.R`

References

Stivers, T., Enfield, N. J., Brown, P., Englert, C., Hayashi, M., Heinemann, T., Hoymann, G., Rossano, F., de Ruiter, J. P., Yoon, K.-E. & Levinson, S. C. (2009). Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26), 10587-10592. doi:10.1073/pnas.0903616106

Levinson, S. C. & Torreira, F. (2015). Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, 6, 731. doi:10.3389/fpsyg.2015.00731

Examples

```
sort(tapply(turn_taking$offset, turn_taking$language, median))
hist(turn_taking$offset, breaks = 60)
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(offset ~ response + answer + gaze + language + (1 | conversation),
                  data = turn_taking)
  lme4::fixef(m)[c("responsedisconfirmation", "answernon-answer", "gazeyes")]
}
```

`visual_world` *Visual-world eye tracking: anticipatory looks*

Description

Listeners hear *The boy will eat / move the cake* while looking at four objects: the target (a cake, the only edible object), a phonological competitor (a cage) and two unrelated distractors. After a restrictive verb (*eat*) they look at the target before the noun is heard; while the noun unfolds, the competitor briefly attracts looks too. Eye-tracker samples on each object in 50 ms bins from verb onset; the noun starts at 700 ms. 32 participants, 24 items, 12 per condition (two counterbalanced lists).

Usage

```
visual_world
```

Format

A tibble with 23040 rows and 10 columns:

participant Participant ID, `P01` to `P32`.

list Counterbalancing list, 1 or 2.

item Item ID, `i01` to `i24`.

condition `"non-restrictive"` or `"restrictive"` verb.

time Start of the 50 ms bin, ms from verb onset, 0 to 1450.

target, competitor, distractor_1, distractor_2 Eye-tracker samples on each object in this bin; they add up to `samples`.

samples Samples per bin: always 25 (500 Hz).

Details

True model: on the logit scale, looks to the target rise from chance (0.25) to about 0.88 along a logistic curve with width 90 ms, centred at 700 ms after restrictive and 930 ms after non-restrictive verbs. By-participant variation in the centre (SD 80 ms) and in the final level (SD 0.3 logits); by-item variation in the centre (SD 40 ms). Within a trial, deviations are AR(1) with rho 0.9. Of the remaining samples, the competitor gets a third plus a bump of up to 25 percentage points centred at 950 ms (SD about 100 ms); the two distractors split the rest evenly. The timing follows

Altmann and Kamide (1999, Experiment 1): the noun began about 700 ms after the verb, and the first saccade to the target came 611 ms after the onset of *eat* and 838 ms after *move*. Their scenes had no phonological competitor; it is added here, and its timing, the participants and the looks are simulated.

Source

`data-raw/visual_world.R`

References

Altmann, G. T. M. & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3), 247-264. doi:10.1016/S00100277(99)000591

Examples

```
visual_world$p_target <- visual_world$target / visual_world$samples
looks <- aggregate(p_target ~ condition + time, data = visual_world, FUN = mean)
plot(p_target ~ time, data = looks, col = condition)
abline(v = 700, lty = 2)
looks[looks$time == 700, ]
```

---

vocabulary_growth *Vocabulary growth in toddlers*

Description

Productive vocabulary on a 680-word parental checklist, every two months from 16 to 30 months, for 80 children. Each child's vocabulary follows an S-shaped (logistic) curve, and children who hear more speech reach its steep middle part earlier. Whether real vocabulary growth has a distinct spurt is debated. Ganger and Brent (2004) modelled the rate of word learning and found that a logistic curve, which implies a spurt, fitted better than a quadratic one for only 5 of 38 children. About 10% of sessions are missing.

Usage

```
vocabulary_growth
```

Format

A tibble with 573 rows and 4 columns:

child Child ID, `C01` to `C80`.

input Adult words per hour heard at home (one value per child).

age Age in months.

vocabulary Number of words produced, out of 680.

Details

True model: `vocabulary` is binomial out of 680 with probability `plogis((age - mid) / 3.2)`, where

```
mid = 25.2 - 2 * log2(input / 1200)
```

plus a by-child deviation (SD 2.6 months), and session-level noise on the logit scale (SD 0.15). Doubling the input moves the midpoint of the curve 2 months earlier; this effect is simulated. The median curve follows the CDI norms in Fenson et al. (1994): median vocabulary of 44 words at 16 months, 225 at 23 months and 573 at 30 months. They also note that a logistic function fits the shape of vocabulary growth.

Source

`data-raw/vocabulary_growth.R`

References

Fenson, L., Dale, P. S., Reznick, J. S., Bates, E., Thal, D. J. & Pethick, S. J. (1994). Variability in early communicative development. *Monographs of the Society for Research in Child Development*, 59(5), 1-185. doi:10.2307/1166093

Fenson, L., Marchman, V. A., Thal, D. J., Dale, P. S., Reznick, J. S. & Bates, E. (2007). *MacArthur-Bates Communicative Development Inventories: User's guide and technical manual* (2nd ed.). Baltimore: Brookes.

Frank, M. C., Braginsky, M., Yurovsky, D. & Marchman, V. A. (2017). Wordbank: An open repository for developmental vocabulary data. *Journal of Child Language*, 44(3), 677-694. doi:10.1017/S0305000916000209

Ganger, J. & Brent, M. R. (2004). Reexamining the vocabulary spurt. *Developmental Psychology*, 40(4), 621-632. doi:10.1037/00121649.40.4.621

Examples

```
plot(vocabulary ~ age, data = vocabulary_growth, col = grey(0.4))
if (requireNamespace("lme4", quietly = TRUE)) {
  vocabulary_growth$session <- seq_len(nrow(vocabulary_growth))
  m <- lme4::glmer(cbind(vocabulary, 680 - vocabulary) ~ I(age - 24) +
                     I(log2(input / 1200)) + (1 | child) + (1 | session),
                   data = vocabulary_growth, family = binomial)
  b <- lme4::fixef(m)
  c(midpoint_at_1200_words = 24 - b[[1]] / b[[2]], months_per_doubling = b[[3]] / b[[2]])
}
```

`vot_identification` *Identification of stops along a VOT continuum*

Description

12 English and 12 Spanish listeners label synthetic stop-vowel syllables as voiced or voiceless. Voice onset time (VOT) runs from -20 to +60 ms in 5 ms steps for labial, apical and velar stops, and each step is heard 10 times. The proportion of voiceless responses follows an S-shaped curve along the continuum. The category boundary lies at a longer VOT for apicals than for labials, and longer again for velars. Spanish listeners place it at a shorter VOT than English listeners. For logistic regression on binomial counts and for estimating category boundaries.

Usage

```
vot_identification
```

Format

A tibble with 1224 rows and 6 columns:

listener Listener ID, `En01` to `En12` and `Sp01` to `Sp12`.

language Native language, `"English"` or `"Spanish"`.

place Place of articulation: `"labial"`, `"apical"` or `"velar"`.

vot Voice onset time of the stimulus in ms.

trials Number of presentations, always 10.

voiceless Number of voiceless responses.

Details

True model: the probability of a voiceless response is `plogis(slope * (vot - boundary - b0))`. The boundaries are the 50% crossover points reported by Abramson and Lisker (1973), who give the English values from Lisker and Abramson (1970): 25, 35 and 42 ms for English and 14, 22 and 24 ms for Spanish labials, apicals and velars. Each listener's boundary is shifted by `b0` (SD 4 ms), and slopes are log-normal around 0.3 per ms (SD 0.25 on the log scale). Listeners and responses are simulated.

Source

`data-raw/vot_identification.R`

References

Abramson, A. S. & Lisker, L. (1973). Voice-timing perception in Spanish word-initial stops. *Journal of Phonetics*, 1(1), 1-8. doi:10.1016/S00954470(19)313725

Lisker, L. & Abramson, A. S. (1970). The voicing dimension: Some experiments in comparative phonetics. In *Proceedings of the 6th International Congress of Phonetic Sciences, Prague 1967* (pp. 563-567).

Examples

```
labial <- subset(vot_identification, place == "labial")
p <- aggregate(cbind(voiceless, trials) ~ language + vot, data = labial, FUN = sum)
plot(voiceless / trials ~ vot, data = p, col = as.integer(language), pch = 16,
     ylab = "Proportion voiceless")
cells <- split(vot_identification, vot_identification[c("language", "place")])
sapply(cells, function(x) {
  b <- coef(glm(cbind(voiceless, trials - voiceless) ~ vot, data = x, family = binomial))
  round(-b[[1]] / b[[2]], 1)
})
m <- glm(cbind(voiceless, trials - voiceless) ~ vot + language * place,
         data = vot_identification, family = binomial)
round(coef(m), 2)
```

---

vowel_discrimination *Same-different discrimination of English vowels*

Description

An AX discrimination task: listeners hear two vowels from different talkers and say whether they are the same or different. 20 native listeners of English and 30 L2 learners each hear six vowel contrasts, with 8 same and 8 different trials per contrast. Native listeners tell all contrasts apart; the learners only some of them. For signal detection theory: hits, false alarms, d' and response bias.

Usage

```
vowel_discrimination
```

Format

A tibble with 4800 rows and 7 columns:

listener Listener ID, `E01` to `E20` (L1) and `G01` to `G30` (L2).

group `"L1"` or `"L2"`.

trial Trial number in the listener's random order, 1 to 96.

contrast Vowel contrast, named by lexical sets: `"FLEECE-KIT"`, `"FLEECE-DRESS"`, `"START-STRUT"`, `"TRAP-START"`, `"THOUGHT-LOT"` or `"GOOSE-FOOT"`.

pair Order of the two vowels: `"AA"`, `"BB"`, `"AB"` or `"BA"`.

trial_type `"same"` (AA, BB) or `"different"` (AB, BA).

response `"same"` or `"different"`.

Details

True model: the differencing model for same-different designs (Macmillan and Creelman 2005). Each vowel gives a normal observation with SD 1; the two observations differ in mean by `d` on different trials and not at all on same trials. The listener says "different" when the absolute difference between them exceeds their criterion (mean 2, SD 0.35 across listeners). `d` is 3.9 to 4.6 for native listeners and 0.7 to 2.6 for learners, multiplied by a listener factor `exp(skill)` with `skill` normal, SD 0.2. With 8 trials per cell, hit rates of 1 and false-alarm rates of 0 are common, so the example adds 0.5 to each count and 1 to each number of trials; this correction pulls the native listeners' d' below their true values. The design follows the AX task of Georgiou and Giannakou (2024), whose learners discriminated FLEECE-DRESS and THOUGHT-LOT above chance and the other contrasts not; the listeners, d' values and responses here are simulated.

Source

`data-raw/vowel_discrimination.R`

References

Georgiou, G. P. & Giannakou, A. (2024). Discrimination of second language vowel contrasts and the role of phonological short-term memory and nonverbal intelligence. *Journal of Psycholinguistic Research*, 53(1), 9. doi:10.1007/s1093602410038z

Macmillan, N. A. & Creelman, C. D. (2005). *Detection Theory: A User's Guide* (2nd ed.). Mahwah, NJ: Erlbaum.

Examples

```
vd <- vowel_discrimination
vd$said_different <- vd$response == "different"
rates <- aggregate(said_different ~ listener + group + contrast + trial_type, data = vd,
                   FUN = function(x) (sum(x) + 0.5) / (length(x) + 1))
rates <- reshape(rates, idvar = c("listener", "group", "contrast"),
                 timevar = "trial_type", direction = "wide")
dprime_sd <- function(h, f) {
  if (h <= f) return(0)
  k <- sqrt(2) * qnorm(f / 2)
  uniroot(function(d) h - pnorm((k + d) / sqrt(2)) - pnorm((k - d) / sqrt(2)),
          c(0, 10))$root
}
rates$dprime <- mapply(dprime_sd, rates$said_different.different,
                       rates$said_different.same)
round(tapply(rates$dprime, rates[c("contrast", "group")], mean), 2)
```

---

vowel_duration *Vowel duration with speakers and words*

Description

16 speakers each produce the same 24 words, 12 with a voiceless and 12 with a voiced final consonant. Speakers differ in speed and in the size of their voicing effect; words differ in duration.

Usage

```
vowel_duration
```

Format

A tibble with 384 rows and 4 columns:

speaker Speaker ID, `S01` to `S16`.

word Word ID, `w01` to `w24`.

voicing `"voiceless"` or `"voiced"`.

duration Vowel duration in ms.

Details

True model: `duration = 150 + s0 + w0 + (20 + s1) * voiced + e`. By-speaker intercepts `s0` have SD 25 ms and slopes `s1` SD 12 ms (uncorrelated); by-word intercepts `w0` have SD 15 ms; residual SD 20 ms.

Source

`data-raw/vowel_duration.R`

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {
  m <- lme4::lmer(duration ~ voicing + (1 + voicing | speaker) + (1 | word),
                  data = vowel_duration)
  summary(m)
}
```

`vowel_voicing` *Vowel duration before voiced and voiceless consonants*

Description

A production study: 60 vowels before voiceless consonants (*beat*) and 60 before voiced ones (*bead*).

Usage

```
vowel_voicing
```

Format

A tibble with 120 rows and 2 columns:

voicing `"voiceless"` or `"voiced"`.

duration Vowel duration in ms.

Details

True model: mean 120 ms (SD 20) before voiceless and 160 ms (SD 22) before voiced consonants.

Source

`data-raw/vowel_voicing.R`

Examples

```
t.test(duration ~ voicing, data = vowel_voicing)
boxplot(duration ~ voicing, data = vowel_voicing)
```

---

web_registers *Linguistic features of six web registers*

Description

Frequencies of 16 lexico-grammatical features in 480 texts from six registers of the web, as used in multi-dimensional analysis (Biber, 1988). Its parameters are based on the web register data of Ehret and Taboada (2021). Three dimensions underlie the features, and one column is the sum of three others. For exploratory factor analysis and, on the dimension scores, cluster analysis.

Usage

```
web_registers
```

Format

A tibble with 480 rows and 20 columns. Except for `word_length`, the features are rates per 1,000 words.

text Text ID, `T001` to `T480`.

register `"forum discussion"`, `"personal blog"`, `"news report"`, `"encyclopedia article"`, `"research article"` or `"recipe"`.

words Text length in words.

adverbs Adverbs.

attributive_adjectives Attributive adjectives (*a new house*).

conjuncts Conjuncts (*however*, *therefore*).

contractions Contractions (*don't*, *it's*).

first_person First person pronouns.

negation Analytic negation (*not*).

nominalisations Nominalisations (*situation*, *development*).

nouns Nouns.

past_tense Past tense verbs.

perfect_aspect Perfect aspect (*has done*).

prepositions Prepositions.

present_tense Present tense verbs.

private_verbs Private verbs (*think*, *feel*, *know*).

pronouns All personal pronouns: the sum of `first_person`, `second_person` and `third_person`.

second_person Second person pronouns.

third_person Third person pronouns.

word_length Mean word length in letters.

Details

True model: each text has scores on three dimensions, involved (vs informational) production, narrative and informational elaboration. They are the register mean plus normal noise with SD 0.6 (involved, elaborated) or 0.9 (narrative):

register	texts	involved	narrative	elaborated
forum discussion	90	1.3	0.0	-1.0
personal blog	90	1.1	0.5	-0.9
news report	110	-0.2	0.5	0.3
encyclopedia article	60	-1.1	0.0	0.5
research article	70	-1.0	-0.5	1.8
recipe	60	0.0	-1.0	-1.6

Each feature has a latent score: the dimension scores times its loadings, plus normal noise with variance 1 minus the sum of its squared loadings. Loadings are 0.75 (`contractions`, `first_person`), 0.7 (`second_person`), 0.6 (`negation`), 0.55 (`adverbs`), 0.5 (`private_verbs`), -0.55 (`prepositions`) and -0.8 (`nouns`) on the involved dimension; 0.85 (`past_tense`), 0.55 (`third_person`) and 0.5 (`perfect_aspect`) on the narrative dimension; 0.75 (`nominalisations`), 0.5 (`attributive_adjectives`) and 0.45 (`conjuncts`) on informational elaboration. `present_tense` loads 0.7 on involved and -0.4 on narrative; `word_length` loads -0.45 on involved and 0.7 on elaboration. The rate per 1,000 words is `base_rate * exp(spread * latent)`, with the base rates and spreads in `data-raw/web_registers.R`, and the number of occurrences in a text is Poisson with mean `rate * words / 1000`; the columns give occurrences per 1,000 words. `word_length` is `4.6 + 0.32 * latent`. Text length is log-normal with a median of 900 words, limited to 400 to 4,000. Because of the log scale and the Poisson counts, a factor analysis of the rates gives smaller loadings than these; log rates come closer. Since `pronouns` is the sum of three other columns, the correlation matrix of all features is singular.

The base rates and spreads were chosen so that the register means approach those of the same six registers in the data of Ehret and Taboada (2021), whose texts come from the Corpus of Online Registers of English (Egbert et al., 2015). Most simulated register means are within 25% of theirs. The largest differences are in pronouns and contractions, most of all in recipes, which in their data use more first and second person pronouns than three dimensions can reproduce. The differences between registers on the dimensions are larger than in their data, so that the registers separate more clearly.

Source

`data-raw/web_registers.R`

References

Biber, D. (1988). *Variation across Speech and Writing*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511621024

Egbert, J., Biber, D. & Davies, M. (2015). Developing a bottom-up, user-based method of web register classification. *Journal of the Association for Information Science and Technology*, 66(9), 1817-1831. doi:10.1002/asi.23308

Ehret, K. & Taboada, M. (2021). Characterising online news comments: A multi-dimensional cruise through online registers. *Frontiers in Artificial Intelligence*, 4, 643770. doi:10.3389/frai.2021.643770

Ehret, K. & Taboada, M. (2021). *MDA-OnlineRegisters* (version 1.0) [Data set and code]. Zenodo. doi:10.5281/zenodo.4885180

Examples

```
features <- setdiff(names(web_registers), c("text", "register", "words", "pronouns"))
fa <- factanal(web_registers[, features], factors = 3, rotation = "promax")
print(fa$loadings, cutoff = 0.3)
aggregate(cbind(contractions, past_tense, nominalisations) ~ register,
          data = web_registers, FUN = median)
round(min(eigen(cor(web_registers[, c(features, "pronouns")]))$values), 10)
```

---

word_order *Word order in a typological sample*

Description

400 invented languages from 100 invented families in six invented areas. Languages mostly inherit their family's word order, so languages from the same family are similar. Adposition order and case marking correlate with word order in the same way in every family, as in Greenberg's (1963) universals. Dunn et al. (2011) argue that such dependencies are mostly specific to lineages, so real samples need not share one pattern. Family sizes are skewed: a few large families and many small ones.

Usage

```
word_order
```

Format

A tibble with 400 rows and 8 columns:

language Language ID, `L001` to `L400`.

family `"Family 001"` to `"Family 100"`.

area `"Area A"` to `"Area F"`.

order Basic word order: `"SOV"`, `"SVO"`, `"VSO"` or `"no dominant order"`.

adpositions `"postpositions"` or `"prepositions"`.

case_marking Whether nouns are case-marked, "no" or "yes".

consonants Size of the consonant inventory.

vowels Size of the vowel inventory.

Details

True model: each family draws a word order from its area's preferences; each language keeps it with probability 0.8 and otherwise draws again. Postpositions have probability 0.92 in SOV, 0.5 in no-dominant-order and 0.12 in VO languages. Case marking has probability 0.7 in SOV and 0.3 in other languages. Consonant inventories are 6 plus a negative binomial count (mean 16, by-family variation); vowel inventories are 3 plus a Poisson count (mean 2.5). Overall, about 44% of the languages are SOV, 36% SVO, 6% VSO and 14% have no dominant order, close to the world-wide proportions in WALS (feature 81A). All language, family and area names and all figures are invented.

Source

data-raw/word_order.R

References

Dryer, M. S. (2013). Order of subject, object and verb. In M. S. Dryer & M. Haspelmath (eds.), *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://wals.info/chapter/81

Greenberg, J. H. (1963). Some universals of grammar with particular reference to the order of meaningful elements. In J. H. Greenberg (ed.), *Universals of Language* (pp. 58-90). Cambridge, MA: MIT Press.

Dunn, M., Greenhill, S. J., Levinson, S. C. & Gray, R. D. (2011). Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, 473(7345), 79-82. doi:10.1038/nature09923

Examples

```
round(prop.table(table(word_order$order)), 2)
round(prop.table(table(word_order$order, word_order$adpositions), 1), 2)
head(sort(table(word_order$family), decreasing = TRUE))
if (requireNamespace("lme4", quietly = TRUE)) {
  word_order$sov <- word_order$order == "SOV"
  m1 <- glm(adpositions ~ sov, data = word_order, family = binomial)
 m2 <- lme4::glmer(adpositions ~ sov + (1 | family), data = word_order, family = binomial)
  rbind(glm = summary(m1)$coefficients["sovTRUE", 1:2],
        glmer = summary(m2)$coefficients["sovTRUE", 1:2])
}
```

zipf_rt *Reaction times and Zipfian word frequency*

Description

2000 words sampled at random from the 50,000 most frequent word types of a corpus whose frequencies follow Zipf's law (frequency proportional to 1 / rank). As in real corpora, most words are rare and a few are very frequent, so frequency is extremely skewed. For log transformations.

Usage

```
zipf_rt
```

Format

A tibble with 2000 rows and 4 columns:

word Word ID, `w0001` to `w2000`.

rank Frequency rank in the corpus, 1 to 50,000.

freq Frequency per million words, `50000 / rank`.

rt Reaction time in ms.

Details

True model: `log(rt) = 6.62 - 0.08 * log10(freq) + e`, with `e` normal, SD 0.15.

Source

`data-raw/zipf_rt.R`

Examples

```
hist(zipf_rt$freq, breaks = 50)
hist(log10(zipf_rt$freq), breaks = 50)
coef(lm(log(rt) ~ log10(freq), data = zipf_rt))
```

Index

∗ **datasets**
accent, 3
acceptability, 4
address_forms, 6
agreement_attraction, 7
aphasia_naming, 8
bouba_kiki_meta, 9
cantonese_tone_merger, 11
code_switching, 12
dgs_iconicity, 13
dialect_survey, 15
disfluencies, 16
do_support, 17
first_words, 19
forensic_comparison, 20
formants, 21
freq_length, 22
german_plurals, 23
german_relatives, 24
hungarian_harmony, 25
iterated_learning, 26
korean_stops, 28
l2_frequency, 29
language_attitudes, 30
lexical_accuracy, 31
lexical_decision, 32
lexical_decision_l2, 33
mandarin_tones, 34
megastudy, 35
messy_export, 36
n400, 37
naming_pilot, 39
overregularization, 40
passives, 41
perception_training, 42
price_f2, 43
pupil_noise, 44
reading_skipping, 45
relativizers, 46
rhythm_metrics, 47
rt_frequency, 49
scalar_implicature, 50
speech_styles, 52
speech_to_song, 52
style_voicing, 54
td_deletion, 55
tobi_agreement, 56
turn_taking, 57
visual_world, 59
vocabulary_growth, 60
vot_identification, 62
vowel_discrimination, 63
vowel_duration, 64
vowel_voicing, 65
web_registers, 66
word_order, 68
zipf_rt, 70

accent, 3
acceptability, 4
address_forms, 6
agreement_attraction, 7
aphasia_naming, 8

bouba_kiki_meta, 9

cantonese_tone_merger, 11
code_switching, 12

dgs_iconicity, 13
dialect_survey, 15
disfluencies, 16
do_support, 17

first_words, 19
forensic_comparison, 20
formants, 21
freq_length, 22

german_plurals, 23

german_relatives, 24

hungarian_harmony, 25

iterated_learning, 26

korean_stops, 28

l2_frequency, 29
language_attitudes, 30
lexical_accuracy, 31
lexical_decision, 32
lexical_decision_l2, 33

mandarin_tones, 34
megastudy, 35
messy_export, 36

n400, 37
naming_pilot, 39

overregularization, 40

passives, 41
perception_training, 42
price_f2, 43
pupil_noise, 44

reading_skipping, 45
relativizers, 46
rhythm_metrics, 47
rt_frequency, *36*, 49

scalar_implicature, 50
simling_datasets, 51
speech_styles, 52
speech_to_song, 52
style_voicing, 54

td_deletion, 55
tobi_agreement, 56
turn_taking, 57

visual_world, 59
vocabulary_growth, 60
vot_identification, 62
vowel_discrimination, 63
vowel_duration, 64
vowel_voicing, 65

web_registers, 66
word_order, 68

zipf_rt, 70