Statistical analysis: The Interval Asymmetry Index and /t/ salience

Ernesto R. Gutiérrez Topete

April 1, 2025

Contents

Project Description

This project documents the statistical analysis accompanying a study of Spanish–English bilinguals' voice onset time (VOT) production in code-switching contexts. Prior research shows that bilingual speech can exhibit phonetic convergence or divergence across languages, shaped by both phonetic and social factors (e.g., Flege & Eefting, 1987; Bullock & Toribio, 2009; Podesva, 2011; Erker, 2022). Building on this work, the current study investigates VOT patterns for voiceless stops (/p/, /t/, /k/) across languages and interactional contexts, with a particular focus on unexpected patterns for /t/ in code-switching environments.

In addition to the mixed-effects modeling, this study also introduces the **Interval Asymmetry Index (IAI)**, a novel statistical measure developed to quantify deviations from the expected linear relationship among categories with gradual increases along a continuum (e.g., VOT). In this case, the IAI provides a continuous index per speaker of whether /t/ patterns align with or diverge from the typical place-of-articulation linear trend. By combining group-level modeling with this individualized metric, the analysis offers new insights into how both phonetic and social salience contribute to systematic variability in bilingual speech.

All analyses are conducted in R (R Core Team, 2025) and follow a reproducible and transparent workflow. The documentation guides readers through every step:

1. **Data import and verification:** The **Bilingual Language Profile** (*BLP*) survey data and VOT production measurements are loaded, inspected, and structured for analysis.
2. **Descriptive statistics:** Key summary measures for both dominance scores BLP and VOT distributions are calculated to provide an overview of central tendency, variability, and range.
3. **Normality checks and visualization:** The distribution of BLP dominance scores is evaluated via the Shapiro–Wilk test and visualized with histograms and Q–Q plots to assess whether parametric modeling assumptions hold.
4. **VOT distribution visualization:** Boxplots illustrate VOT patterns across stop consonants and languages, offering a visual baseline for further modeling.
5. **Mixed-effects modeling of VOT:** A maximal linear mixed-effects model is initially fit to predict VOT from stop consonant, language, and their interaction, with random intercepts and slopes for subjects and lexical items. When singular fits are encountered, principled simplification is conducted using the `buildmer` package to obtain a stable, interpretable model.
6. **Interval Asymmetry Index (IAI):** A novel metric is computed for each speaker to quantify deviations of /t/ from the expected linear VOT progression, providing a continuous measure of asymmetry along the /p/–/t/–/k/ continuum.
7. **IAI visualization:** Histograms display the distribution of IAI values per language, highlighting language-specific patterns in /t/ realization relative to the expected midpoint.
8. **Mixed-effects modeling of IAI:** A linear mixed-effects model predicts IAI using normalized linguistic background and cultural identity measures, accounting for subject- and language-level variability. Random slopes are omitted due to limited observations per subject, ensuring model convergence and accurate effect estimation.

This document combines thorough data processing, statistical modeling, and visualization steps, providing a fully reproducible workflow. The analysis leverages modern R packages for data manipulation (`dplyr`, Wickham et al., 2023), visualization (`ggplot2`, `patchwork`, and `showtext`), and mixed-effects modeling and post-hoc analysis with (`lme4`, `lmerTest`, `buildmer`, and `emeans`) to ensure reproducibility and clarity. By integrating both group-level modeling and individualized metrics, the analysis sheds light on the phonetic and social factors shaping bilingual speech.

The complete workflow is scripted to facilitate audit, replication, and extension by other researchers. All code, analyses, and visualizations were prepared by **Ernesto R. Gutiérrez Topete** and are publicly available, in conjunction with the data.

Data Processing and Inspection

Libraries

This section loads the R packages required for data manipulation, visualization, statistical modeling, and reproducible reporting. Each package is briefly described in terms of its role in the current workflow.

- `tidyverse`: A collection of R packages designed for data science, including `dplyr` (data wrangling) and `ggplot2` (visualization).
- `patchwork`: Combines multiple `ggplot2` plots into a single layout (e.g., for side-by-side comparisons).
- `showtext`: Provides font formatting functionality for graphs in R.
- `lme4`: Fits linear and generalized linear mixed-effects models via maximum likelihood.
- `lmerTest`: Provides *p*-values and significance testing for models fitted with `lme4`.
- `buildmer`: Automates stepwise simplification and selection of mixed-effects model structures.
- `emmeans`: Performs post-hoc analyses for independent variables with multiple levels and their interactions.

```r
# Load the libraries
library(tidyverse)    # Provide data manipulation (dplyr) and visualization (ggplot2)
```

```r
library(patchwork)    # Combine multiple ggplot2 plots in a single figure
library(showtext)     # Produce font format options for text in plots and graphs
library(lme4)         # Fit linear mixed-effects models (lmer)
library(lmerTest)     # Add p-values and tests for fixed effects in lme4 models
library(buildmer)     # Perform stepwise elimination and model selection for mixed-effects models
library(emmeans)      # Analyzes estimated marginal means for fitted models
```

Data

This section imports the two data sets used in the analysis: (1) the BLP survey responses, which measure participants' linguistic profiles: linguistic history, usage, proficiency, and attitudes (Birdsong, Gertken & Amengual, 2012) and (2) the *VOT production* data set, containing VOT measurements during four code-switching tasks for the 60 Spanish-English bilingual participants. After loading the files, it previews their contents and inspects their structure to confirm that the data have been imported successfully and that the variables are correctly formatted, to ensure that subsequent processing and analysis steps are based on properly loaded and verified data. This documentation file assumes a structure where the data are stored in a directory called `data/`.

```r
# Load the BLP and VOT data sets
blp_df <- read.csv("data/blp_data.csv", header = TRUE)   # Load BLP survey data
vot_df <- read.csv("data/vot_data.csv", header = TRUE)   # Load VOT production data

# Preview the first few rows of the BLP and VOT data sets (commented out for clarity)
# head(blp_df)   # Display first rows of BLP data
# head(vot_df)   # Display first rows of VOT data

# Inspect the structure of the BLP and VOT data sets (commented out for clarity)
# str(blp_df)   # Display structure of BLP data
# str(vot_df)   # Display structure of VOT data
```

Dominance Scores Normality Test

This section conducts a normality test on the dominance scores produced by the BLP. Specifically, the analysis applies the **Shapiro–Wilk test** using the R function `shapiro.test()`, which evaluates whether the data distribution seriously deviates from normality. In this context, the null hypothesis (H_0) states that the data are drawn from a normally distributed population, whereas the alternative hypothesis (H_1) posits that the data significantly depart from normality. The output of this test will provide a W statistic and p-value to assess normality; a p-value greater than 0.05 suggests that the data do not significantly deviate from normality (i.e., H_0 cannot be rejected), while a p-value less than 0.05 indicates that the normality assumption is violated (i.e., H_0 is rejected in favor of H_1).

```r
# Perform Shapiro-Wilk test on dominance scores from the BLP
shapiro.test(blp_df$dom_score)
```

```
>>
>>  Shapiro-Wilk normality test
>>
>> data:  blp_df$dom_score
>> W = 0.97878, p-value = 0.3574
```

The Shapiro–Wilk test returned a W statistic of 0.979 with a corresponding p-value of 0.36. These results indicate that the null hypothesis (H_0) cannot be rejected, meaning the dominance scores do not significantly deviate from normality. Therefore, the distribution of BLP dominance scores can reasonably be treated as normal for the purposes of subsequent statistical modeling.

Dominance Score Normality Visualization

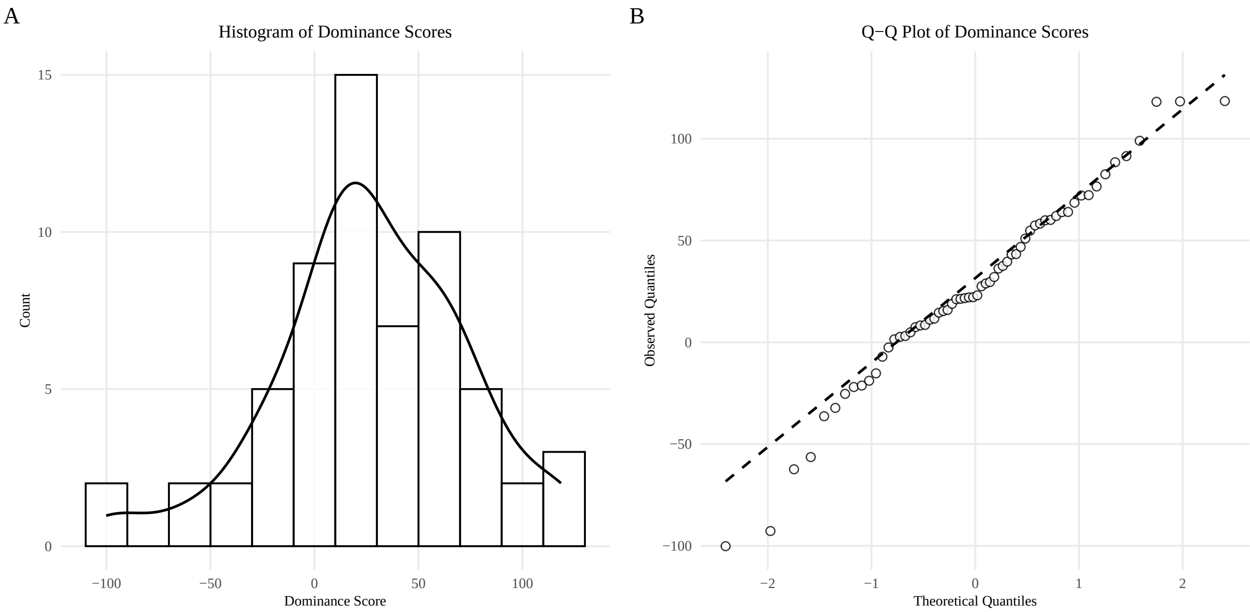
This section visualizes the distribution of BLP dominance scores to complement the Shapiro–Wilk normality test. A histogram and a Q–Q plot are generated to assess whether the data plausibly follow a normal distribution. The histogram overlays a density curve to provide a visual sense of the distribution's shape, while the Q–Q plot compares the sample quantiles to the theoretical quantiles of a normal distribution. These plots enable visual confirmation of normality alongside formal statistical testing.

The visualizations are produced using `ggplot2` (Wickham, 2016) for plotting and `patchwork` (Pedersen, 2020) for arranging the plots side by side.

Note: Figures will be saved in the folder called `images/` in the working directory. If not such folder exists, one will be created.

```r
# Set bin width for histogram
binwidth <- 20

# Create histogram of dominance scores
hist_plot <- blp_df %>%
  ggplot(aes(x = dom_score)) +
  geom_histogram(binwidth = binwidth,
                 fill = "white", color = "black", alpha = 0.8) +
  geom_density(aes(y = after_stat(count) * binwidth),
               color = "black", linetype = "solid", size = 0.7) +
  labs(title = "Histogram of Dominance Scores",
       x = "Dominance Score",
       y = "Count") +
  theme_minimal(base_family = "serif") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    panel.grid.minor = element_blank()
  )

# Create Q-Q plot of dominance scores
qq_plot <- blp_df %>%
  ggplot(aes(sample = dom_score)) +
  stat_qq(geom = "point", shape = 21, fill = "white", color = "black", size = 2, alpha = 0.8) +
  stat_qq_line(color = "black", linetype = "dashed", size = 0.7) +
  labs(title = "Q-Q Plot of Dominance Scores",
       x = "Theoretical Quantiles",
       y = "Observed Quantiles") +
  theme_minimal(base_family = "serif") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    panel.grid.minor = element_blank()
  )

# Combine histogram and Q-Q plot side by side with bold serif panel labels
final_plot <- hist_plot + qq_plot +
  plot_annotation(
    tag_levels = 'A',
    theme = theme(
```

```
    text = element_text(family = "serif"),
    plot.tag = element_text(family = "serif", face = "bold", size = 12)
  )
)

# Print combined plot
final_plot
```




```
# Save image of plots to local machine (directory: "images/")
ggsave("images/dom_score_normality.pdf",
       plot = final_plot,
       width = 12,
       height = 5,
       units = "cm",
       dpi = 600)
```

The histogram and Q–Q plot show only minor deviations from a perfectly normal distribution. Overall, the dominance scores closely approximate normality, supporting the results of the Shapiro–Wilk test.

VOT Descriptive Statistics

This section provides descriptive statistics for the VOT data prior to running inferential models. Using `dplyr` (Wickham et al., 2023), the data are grouped by **language** and **stop consonant** categories to calculate summary measures including mean VOT, standard deviation, number of tokens, standard error, and minimum and maximum values. These descriptive statistics offer a clear overview of central tendency, variability, and range, and serve as a baseline for interpreting subsequent mixed-effects regression models.

```
# Calculate descriptive statistics for VOT production data
vot_stats_df <- vot_df %>%
  group_by(language, stop) %>%
  summarise(
    mean_VOT = mean(vot_dur, na.rm = TRUE),    # Compute mean VOT
    sd       = sd(vot_dur, na.rm = TRUE),      # Compute standard deviation
    n        = n(),                            # Count number of tokens
    se       = sd / sqrt(n),                   # Compute standard error
```

```r
    min      = min(vot_dur, na.rm = TRUE),     # Compute minimum VOT
    max      = max(vot_dur, na.rm = TRUE)      # Compute maximum VOT
  ) %>%
  mutate(across(where(is.numeric), ~round(.x, 3)))

# Display descriptive statistics as a formatted table
knitr::kable(vot_stats_df)
```

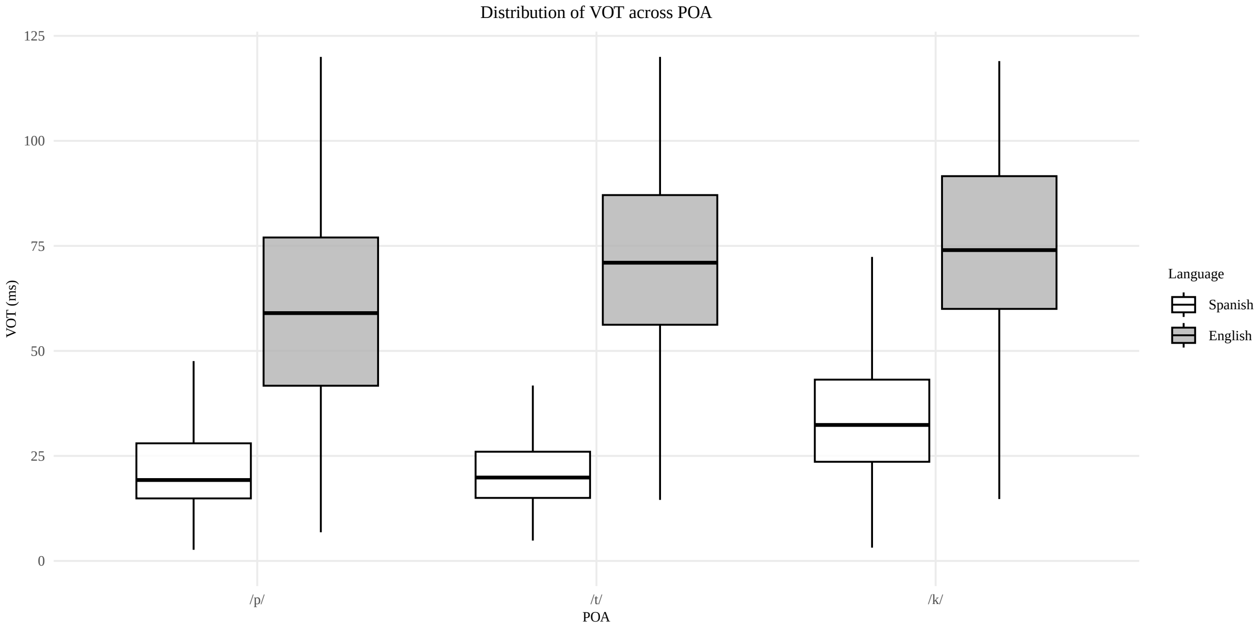
language	stop	mean_VOT	sd	n	se	min	max
english	k	80.071	26.328	1425	0.697	14.730	229.292
english	p	61.342	26.145	1150	0.771	6.824	162.000
english	t	75.983	26.837	1397	0.718	14.530	296.747
spanish	k	35.302	16.820	2284	0.352	3.158	163.000
spanish	p	23.103	13.690	1880	0.316	2.655	128.594
spanish	t	22.609	12.157	1700	0.295	4.846	100.243

VOT Distribution Visualization

This section visualizes the distribution of VOT values across place of articulation (POA) for Spanish and English. Using `ggplot2` (Wickham, 2016), boxplots are generated with POA on the x-axis and VOT in milliseconds on the y-axis, with separate boxes filled by language. The y-axis is restricted to the 0–120 ms range for clarity.

```r
# Create boxplot of VOT by stop and language
ggplot(vot_df, aes(
  x = factor(stop, levels = c("p", "t", "k")),
  y = vot_dur,
  fill = factor(language, levels = c("spanish", "english"))
  )
) +
  geom_boxplot(
    outlier.shape = NA,
    alpha = 0.8,
    color = "black"
  ) +
  labs(
    title = "Distribution of VOT across POA",
    x = "POA",
    y = "VOT (ms)",
    fill = "Language"
  ) +
  scale_y_continuous(limits = c(0, 120)) +   # Restrict y-axis range
  scale_x_discrete(labels = c("/p/", "/t/", "/k/")) +
  scale_fill_manual(
    values = c("white", "grey70"),   # Spanish = white, English = grey
    labels = c("Spanish", "English")
  ) +
  theme_minimal(base_family = "serif") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 8),
```

```r
    legend.text = element_text(size = 8),
    panel.grid.minor = element_blank()
  )
```




```r
# Save image of plots to local machine (directory: "images/")
ggsave("images/stop_vot_box.pdf",
       width = 12,
       height = 5,
       units = "cm",
       dpi = 600)
```

Statistical Analysis

Mixed-effect Maximal Model

This section fits a **maximal linear mixed-effects model** to the VOT data to evaluate whether /t/ deviates from the expected linear increase across places of articulation (/p/ → /t/ → /k/) in Spanish and English. The dependent variable is VOT duration (`vot_dur`), modeled as a function of stop consonant, language, and their interaction. To account for repeated measures and variability across speakers and lexical items, the model includes random intercepts and slopes for the interaction of stop and language by both subject and word.

The model is estimated using `lme4` (Bates et al., 2015) via the `lmer()` function, and *p*-values for fixed effects are obtained using `lmerTest` (Kuznetsova et al., 2017).

```r
# Fit maximal linear mixed-effects model to VOT data
max_mod_t <- lmer(
  vot_dur ~ stop * language +          # Include fixed effects: stop, language, and interaction
    (1 + stop * language | subject) +  # Add random intercepts and slopes by subject
    (1 + stop * language | word),      # Add random intercepts and slopes by word
  data = vot_df,
  REML = FALSE                         # Use maximum likelihood estimation
)
```

```r
# Display model summary with p-values from lmerTest
summary(max_mod_t)
```

```
>> Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
>>   method [lmerModLmerTest]
>> Formula: vot_dur ~ stop * language + (1 + stop * language | subject) +
>>     (1 + stop * language | word)
>>    Data: vot_df
>>
>>      AIC      BIC   logLik -2*log(L)  df.resid
>>   82253.5  82606.0 -41077.8  82155.5      9787
>>
>> Scaled residuals:
>>     Min      1Q  Median      3Q     Max
>> -6.8061 -0.5018 -0.1082  0.4186 12.2783
>>
>> Random effects:
>>  Groups   Name                  Variance Std.Dev. Corr
>>  word     (Intercept)           319.78   17.882
>>           stopp                 158.56   12.592   -0.59
>>           stopt                  56.55    7.520   -0.60  0.62
>>           languagespanish       139.11   11.795   -1.00  0.60  0.63
>>           stopp:languagespanish 107.27   10.357    0.07 -0.77 -0.54 -0.11
>>           stopt:languagespanish  63.83    7.990   -0.13 -0.12 -0.68  0.09  0.52
>>  subject  (Intercept)           167.55   12.944
>>           stopp                  55.06    7.421   -0.17
>>           stopt                  38.08    6.171    0.00  0.76
>>           languagespanish       142.18   11.924   -0.75 -0.14 -0.22
>>           stopp:languagespanish  39.77    6.307   -0.23 -0.81 -0.70  0.20
>>           stopt:languagespanish  30.05    5.481   -0.49 -0.43 -0.76  0.37  0.76
>>  Residual                       219.10   14.802
>> Number of obs: 9836, groups:  word, 553; subject, 60
>>
>> Fixed effects:
>>                       Estimate Std. Error       df t value Pr(>|t|)
>> (Intercept)            76.7937     2.7215 124.5114  28.217  < 2e-16 ***
>> stopp                 -13.8439     3.0431 155.1249  -4.549 1.08e-05 ***
>> stopt                   1.2417     2.9618 134.3527   0.419    0.676
>> languagespanish       -41.5026     2.8101 147.7344 -14.769  < 2e-16 ***
>> stopp:languagespanish   0.9081     3.2297 192.4849   0.281    0.779
>> stopt:languagespanish -14.1724     3.1383 165.0694  -4.516 1.19e-05 ***
>> ---
>> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>>
>> Correlation of Fixed Effects:
>>             (Intr) stopp  stopt  lnggsp stpp:l
>> stopp       -0.590
>> stopt       -0.572  0.577
>> langugspnsh -0.856  0.514  0.521
>> stpp:lnggsp  0.488 -0.912 -0.529 -0.581
>> stpt:lnggsp  0.472 -0.513 -0.921 -0.580  0.588
>> optimizer (nloptwrap) convergence code: 0 (OK)
>> unable to evaluate scaled gradient
>> Model failed to converge: degenerate  Hessian with 5 negative eigenvalues
```

Singularity Check

This section evaluates whether the maximal model suffers from singularity or convergence issues using the `isSingular()` function from `lme4` (Bates et al., 2015). A singular fit occurs when one or more random-effects variances are estimated to be zero or near zero, indicating an overparameterized model. If the function returns `TRUE`, it signals the need for simplification of the random-effects structure (i.e., removing factors or factor combinations).

```
# Check for singular fit in the maximal mixed-effects model
isSingular(max_mod_t)
```

```
>> [1] FALSE
```

The singularity check returned `TRUE`, indicating that the maximal model is overparameterized and that simplification of the random-effects structure is warranted.

Parsimonious Model Selection with Buildmer

This section addresses the singularity issue detected in the maximal model by using `buildmer` (Voeten, 2020) to perform principled stepwise simplification of the random-effects structure. The `buildmer` algorithm evaluates which components of the model meaningfully contribute to fit, systematically removing redundant terms using likelihood ratio tests (LRT). This ensures a parsimonious model specification that balances explanatory power with stability and interpretability.

```
# Define maximal model formula with fixed and random effects
form <- vot_dur ~ stop * language +
        (1 + stop * language | subject) +   # Include random intercepts and slopes by subject
        (1 + stop * language | word)        # Include random intercepts and slopes by word

# Run buildmer to simplify model using backward selection and LRT criterion
mod_build <- buildmer(
  form,
  data = vot_df,
  buildmerControl = buildmerControl(
    direction = 'backward',   # Remove terms step by step
    crit = 'LRT'              # Use likelihood ratio test for simplification
  )
)
```

```
>>   grouping          term               block Iteration           LRT
>> 1     <NA>             1             NA NA 1         1            NA
>> 2     <NA>          stop          NA NA stop         1            NA
>> 3     <NA>      language      NA NA language         1            NA
>> 4     <NA> stop:language NA NA stop:language         1  1.345170e-56
>> 5  subject             1         NA subject 1         1            NA
>> 6  subject          stop     NA subject stop         1  5.720422e-19
>> 7  subject      language NA subject language         1 3.579304e-253
```

```
# Display the formula of the simplified model
formula(mod_build)
```

```
>> vot_dur ~ 1 + stop + language + stop:language + (1 + stop + language |
>>     subject)
```

```
# Display summary of the final simplified model
summary(mod_build)
```

```
>> Linear mixed model fit by REML
```

```
>> (p-values based on Wald z-scores) ['lmerMod']
>> Formula:
>> vot_dur ~ 1 + stop + language + stop:language + (1 + stop + language |
>>     subject)
>>    Data: vot_df
>>
>> REML criterion at convergence: 84355.9
>>
>> Scaled residuals:
>>     Min      1Q  Median      3Q     Max
>> -5.1338 -0.4997 -0.1211  0.4450 11.1586
>>
>> Random effects:
>>  Groups   Name            Variance Std.Dev. Corr
>>  subject  (Intercept)     210.67   14.514
>>           stopp            24.01    4.900   -0.42
>>           stopt            17.56    4.191   -0.31  0.92
>>           languagespanish 183.78   13.556   -0.80 -0.08 -0.12
>>  Residual                 296.94   17.232
>> Number of obs: 9836, groups:  subject, 60
>>
>> Fixed effects:
>>                       Estimate Std. Error t value Pr(>|t|)
>> (Intercept)            78.4295     1.9319  40.598  < 2e-16 ***
>> stopp                 -17.8633     0.9371 -19.061  < 2e-16 ***
>> stopt                  -3.6414     0.8508  -4.280 1.87e-05 ***
>> languagespanish       -43.1148     1.8492 -23.316  < 2e-16 ***
>> stopp:languagespanish   4.7655     0.8766   5.437 5.43e-08 ***
>> stopt:languagespanish  -9.4632     0.8575 -11.035  < 2e-16 ***
>> ---
>> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>>
>> Correlation of Fixed Effects:
>>             (Intr) stopp  stopt  lnggsp stpp:l
>> stopp       -0.394
>> stopt       -0.322  0.667
>> langugspnsh -0.799  0.069  0.062
>> stpp:lnggsp  0.125 -0.575 -0.283 -0.214
>> stpt:lnggsp  0.127 -0.262 -0.582 -0.217  0.459
```

Parsimonious Model Post-hoc

Given that the parsimonious model reported a significant interaction between stop category and language, we conducted a post-hoc analysis using the `emmeans` package. Estimated marginal means (EMMs) were obtained for each stop category in both English and Spanish, and pairwise contrasts were performed to examine language effects within each stop, as well as stop effects within each language. Multiple comparisons were adjusted using the Bonferroni method. These post-hoc comparisons allow us to determine which specific stop categories drive the significant interaction and whether the cross-linguistic differences are uniform across places of articulation or concentrated in particular contrasts.

```r
# Create variable for parsimonious model
pars_mod <- lmer(vot_dur ~ stop * language + (1 + stop | subject), data = vot_df)
summary(pars_mod)
```

```
>> Linear mixed model fit by REML. t-tests use Satterthwaite's method [
```

```
>> lmerModLmerTest]
>> Formula: vot_dur ~ stop * language + (1 + stop | subject)
>>    Data: vot_df
>>
>> REML criterion at convergence: 85529.9
>>
>> Scaled residuals:
>>     Min      1Q  Median      3Q     Max
>> -5.0867 -0.5870 -0.0996  0.4727 10.1242
>>
>> Random effects:
>>  Groups   Name        Variance Std.Dev. Corr
>>  subject  (Intercept)  86.87    9.320
>>           stopp        23.59    4.857   -0.74
>>           stopt        19.94    4.465   -0.51  0.90
>>  Residual             340.66   18.457
>> Number of obs: 9836, groups:  subject, 60
>>
>> Fixed effects:
>>                         Estimate Std. Error        df t value Pr(>|t|)
>> (Intercept)              79.1328     1.3018   70.3868   60.78  < 2e-16 ***
>> stopp                   -18.0866     0.9687  136.7555  -18.67  < 2e-16 ***
>> stopt                    -3.9168     0.9088  132.6625   -4.31 3.16e-05 ***
>> languagespanish         -44.3498     0.6273 9698.0343  -70.70  < 2e-16 ***
>> stopp:languagespanish     5.4280     0.9358 9711.1343    5.80 6.83e-09 ***
>> stopt:languagespanish    -9.1961     0.9168 9717.6180  -10.03  < 2e-16 ***
>> ---
>> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>>
>> Correlation of Fixed Effects:
>>             (Intr) stopp  stopt  lnggsp stpp:l
>> stopp       -0.639
>> stopt       -0.505  0.649
>> langugspnsh -0.295  0.396  0.422
>> stpp:lnggsp  0.197 -0.594 -0.284 -0.670
>> stpt:lnggsp  0.202 -0.271 -0.582 -0.684  0.459
```

```r
# Estimate marginal means for parsimonious model
emm <- emmeans(pars_mod, ~ stop * language)

# Provide Pairwise comparisons within language (differences between stops)
pairs(emm, by = "language", adjust = "bonferroni")
```

```
>> language = english:
>>  contrast estimate    SE  df z.ratio p.value
>>  k - p      18.087 0.969 Inf  18.671  <.0001
>>  k - t       3.917 0.909 Inf   4.310  <.0001
>>  p - t     -14.170 0.788 Inf -17.980  <.0001
>>
>> language = spanish:
>>  contrast estimate    SE  df z.ratio p.value
>>  k - p      12.659 0.858 Inf  14.752  <.0001
>>  k - t      13.113 0.834 Inf  15.719  <.0001
>>  p - t       0.454 0.683 Inf   0.666  1.0000
>>
```

```
>> Degrees-of-freedom method: asymptotic
>> P value adjustment: bonferroni method for 3 tests
```

Interval Asymmetry Index (IAI)

The **Interval Asymmetry Index (IAI)** is a novel metric designed to quantify deviations of a given category along a spectrum where three labels are expected to be equally spaced. In this study, VOT is expected to increase linearly across stop categories (/p/ → /t/ → /k/).

For each speaker and language, the algorithm first calculates the mean VOT per stop category:

$$\bar{v}_{ij} = \frac{1}{n_{ij}} \sum_{k=1}^{n_{ij}} n_{ijk} \tag{1}$$

It then computes the IAI:

$$IAI = \frac{(\bar{v}_{i3} - \bar{v}_{i2}) - (\bar{v}_{i2} - \bar{v}_{i1})}{\bar{v}_{i3} - \bar{v}_{i1}} \tag{2}$$

An IAI of **0** indicates that /t/ lies at the expected midpoint between /p/ and /k/, a **positive IAI** indicates that /t/ has a shorter-than-expected VOT, and a **negative IAI** indicates a longer-than-expected VOT. This continuous measure allows for quantifying subtle deviations from the canonical place-of-articulation hierarchy per speaker, which can be further analyzed with linguistic and sociodemographic variables, shedding light on the factors that may influence speech behaviors.

```r
# Compute mean VOT per stop category per speaker and language
vij_df <- vot_df %>%
  group_by(subject, language, stop) %>%
  summarise(vij = mean(vot_dur)) %>%   # Compute mean VOT per POA
  as.data.frame()

# Restructure data to create per-speaker/per-language values for /p/, /t/, /k/ and compute IAI
iai_df <- vij_df %>%
  group_by(subject, language) %>%
  reframe(
    vi_p = vij[stop == "p"],   # Assign mean VOT for /p/
    vi_t = vij[stop == "t"],   # Assign mean VOT for /t/
    vi_k = vij[stop == "k"]    # Assign mean VOT for /k/
  ) %>%
  mutate(
    iai = ((vi_k - vi_t) - (vi_t - vi_p)) / (vi_k - vi_p)  # Compute IAI
  ) %>%
  ungroup() %>%
  as.data.frame()

# Summarize IAI distribution by language
iai_stats_df <- iai_df %>%
  group_by(language) %>%
  summarise(
    mean_iai = mean(iai),   # Compute mean IAI
    SD       = sd(iai),     # Compute standard deviation
    max      = max(iai),    # Compute maximum IAI
    min      = min(iai)     # Compute minimum IAI
  )
```

```r
# Display summary statistics as a formatted table
knitr::kable(iai_stats_df)
```

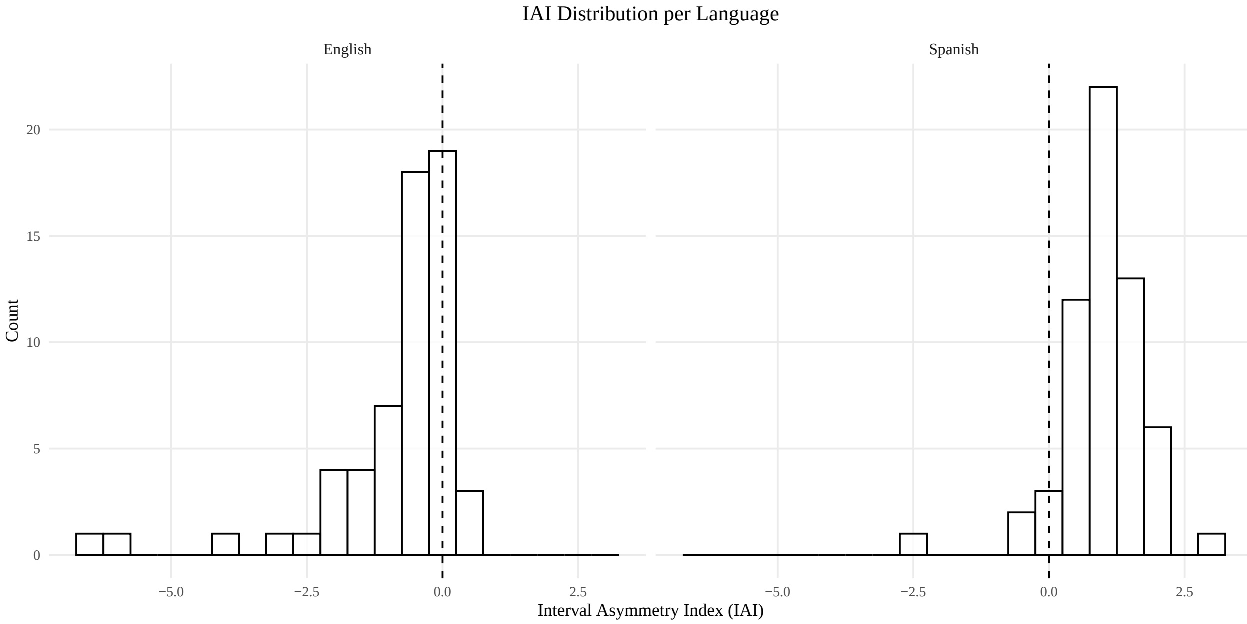
language	mean_iai	SD	max	min
english	-0.8194402	1.334297	0.6798785	-6.437863
spanish	0.9595622	0.777889	2.7694909	-2.643269

IAI Distribution Visualization

This section visualizes the distribution of **Interval Asymmetry Index (IAI)** values across languages. Histograms display how speakers align with or deviate from the expected linear VOT progression. A vertical dashed line at 0 indicates the baseline where /t/ lies exactly at the predicted midpoint between /p/ and /k/. Separate panels for Spanish and English are generated, displaying language-specific trends in /t/ realization.

The visualizations are created using `ggplot2` (Wickham, 2016) for plotting.

```r
# Plot histogram of IAI values by language and add vertical line at 0
ggplot(iai_df, aes(x = iai)) +
  geom_histogram(binwidth = 0.5, color = "black", fill = "white", alpha = 0.8) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "black") +
  facet_wrap(~ language,
             ncol = 2,
             labeller = labeller(language = c("english"="English",
                                              "spanish"="Spanish"))) +
  labs(
    title = "IAI Distribution per Language",
    x = "Interval Asymmetry Index (IAI)",
    y = "Count"
  ) +
  theme_minimal(base_family = "serif") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 8),
    legend.position = "none",
    panel.grid.minor = element_blank()
  )
```




```
# Save image of plots to local machine (directory: "images/")
ggsave("images/iai_hist.pdf",
       width = 12,
       height = 5,
       units = "cm",
       dpi = 600)
```

Mixed-Effects Model: Predicting IAI

This section fits a linear mixed-effects model to predict the **Interval Asymmetry Index (IAI)** as a function of linguistic background and cultural identity variables. All numerical predictor variables–**dominance score** and **ages of learning English and Spanish**–are normalized (z-scored) prior to model fitting. Normalization ensures that predictors are on a comparable scale, which improves model convergence, stabilizes estimates, and allows for more accurate interpretation of effect sizes, particularly when interaction terms are included.

Fixed effects include **L1**, normalized **dominance score**, normalized **ages of learning English** and **ages of learning Spanish** (and their interaction), and normalized **cultural identification with English** and **cultural identification with Spanish** (and their interaction). Random intercepts are included for **Language** and **Subject**. Due to the limited data set (120 observations total, one per language per speaker), a maximal random-effects structure was not feasible. Instead, a simplified model with random intercepts only was specified to ensure model convergence and stability.

The model is estimated using `lme4` (Bates et al., 2015) via `lmer()`, and *p*-values for fixed effects are obtained using `lmerTest` (Kuznetsova et al., 2017).

```
# Merge IAI data with BLP data by subject
merged_df <- merge(iai_df, blp_df, by = "subject")

# Create L1 variable: assign "spa" if Spanish learned first, otherwise "eng"
merged_df <- merged_df %>%
  mutate(L1 = ifelse(age_learning_spa < age_learning_eng, "spa", "eng"))

# Normalize numerical predictors: dominance score, age of learning, and cultural identification
merged_df <- merged_df %>%
```

```r
  mutate(
    dom_score_z = scale(dom_score),
    age_learning_eng_z = scale(age_learning_eng),
    age_learning_spa_z = scale(age_learning_spa),
    identify_eng_culture_z = scale(identify_eng_culture),
    identify_spa_culture_z = scale(identify_spa_culture)
  )

# Fit linear mixed-effects model predicting IAI with normalized predictors
mm_mod_iai <- lmer(
  iai ~ L1 + dom_score_z +
    age_learning_eng_z * age_learning_spa_z +
    identify_eng_culture_z * identify_spa_culture_z +
    (1 | language) +
    (1 | subject),
  data = merged_df
)

# Display model summary with p-values from lmerTest
summary(mm_mod_iai)
```

```
>> Linear mixed model fit by REML. t-tests use Satterthwaite's method [
>> lmerModLmerTest]
>> Formula: iai ~ L1 + dom_score_z + age_learning_eng_z * age_learning_spa_z +
>>     identify_eng_culture_z * identify_spa_culture_z + (1 | language) +
>>     (1 | subject)
>>    Data: merged_df
>>
>> REML criterion at convergence: 368.6
>>
>> Scaled residuals:
>>     Min      1Q  Median      3Q     Max
>> -4.2665 -0.3212  0.1002  0.5222  1.7873
>>
>> Random effects:
>>  Groups   Name        Variance Std.Dev.
>>  subject  (Intercept) 0.02562  0.1601
>>  language (Intercept) 1.56440  1.2508
>>  Residual             1.08045  1.0394
>> Number of obs: 120, groups:  subject, 60; language, 2
>>
>> Fixed effects:
>>                                               Estimate Std. Error       df
>> (Intercept)                                    0.49535    0.96687  1.39504
>> L1spa                                         -0.52851    0.45575 51.00244
>> dom_score_z                                   -0.07590    0.20244 51.00244
>> age_learning_eng_z                             0.02219    0.17865 51.00244
>> age_learning_spa_z                             0.10951    0.17115 51.00244
>> identify_eng_culture_z                        -0.21322    0.13896 51.00244
>> identify_spa_culture_z                         0.42375    0.15692 51.00244
>> age_learning_eng_z:age_learning_spa_z         -0.06968    0.08768 51.00244
>> identify_eng_culture_z:identify_spa_culture_z  0.25020    0.09971 51.00244
>>                                               t value Pr(>|t|)
>> (Intercept)                                     0.512  0.67772
```

```
>> L1spa                                       -1.160  0.25159
>> dom_score_z                                 -0.375  0.70928
>> age_learning_eng_z                           0.124  0.90164
>> age_learning_spa_z                           0.640  0.52513
>> identify_eng_culture_z                      -1.534  0.13112
>> identify_spa_culture_z                       2.700  0.00937 **
>> age_learning_eng_z:age_learning_spa_z       -0.795  0.43046
>> identify_eng_culture_z:identify_spa_culture_z   2.509  0.01531 *
>> ---
>> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>>
>> Correlation of Fixed Effects:
>>             (Intr) L1spa  dm_sc_ ag_lrnng_n_ ag_lrnng_s_ idntfy_n__ idntfy_s__
>> L1spa       -0.391
>> dom_score_z  0.117 -0.291
>> ag_lrnng_n_  0.214 -0.547  0.656
>> ag_lrnng_s_ -0.181  0.467 -0.203 -0.348
>> idntfy_ng__ -0.043  0.123 -0.497 -0.107      -0.127
>> idntfy_sp__  0.064 -0.176  0.301  0.040       0.403      -0.438
>> ag_lr__:___  0.095 -0.185  0.177  0.122      -0.120       0.206     -0.271
>> idnt___:___  0.076 -0.196  0.093  0.042       0.436      -0.212      0.550
>>             a___:_
>> L1spa
>> dom_score_z
>> ag_lrnng_n_
>> ag_lrnng_s_
>> idntfy_ng__
>> idntfy_sp__
>> ag_lr__:___
>> idnt___:___ -0.186
```

IAI Model Post-hoc

Because the model revealed a significant interaction between identification with English culture and identification with Spanish culture ($\beta = 0.25$, $p = .015$), a post-hoc analysis was conducted to clarify the nature of this effect. Specifically, we examined how varying levels of identification with one cultural group modulated the relationship between identification with the other and IAI scores. This allowed us to test whether bicultural identification produced an amplifying effect, whereby simultaneous high identification with both cultures predicted stronger IAI outcomes than would be expected from the sum of each effect in isolation.

The post-hoc analysis was performed in R using the `emmeans` package (Lenth, 2025). Estimated marginal means were generated for combinations of English and Spanish cultural identification, defined at representative values (-1 SD, mean, $+1$ SD). Pairwise contrasts were then computed to assess whether increases in IAI differed significantly across cultural identification profiles. The code below illustrates the procedure:

```r
# Generate estimated marginal means for combinations of cultural identities
emm_culture <- emmeans(mm_mod_iai,
                       ~ identify_eng_culture_z * identify_spa_culture_z,
                       at = list(identify_eng_culture_z = c(-1, 0, 1),
                                 identify_spa_culture_z = c(-1, 0, 1)))

# Display estimated means
emm_culture
```

```
>>  identify_eng_culture_z identify_spa_culture_z  emmean    SE   df lower.CL
>>                      -1                     -1  0.2708 0.917 1.13    -8.66
```

```
>>                        0                       -1 -0.1927 0.912 1.11    -9.45
>>                        1                       -1 -0.6561 0.946 1.28    -7.97
>>                       -1                        0  0.4443 0.915 1.12    -8.60
>>                        0                        0  0.2311 0.903 1.06    -9.81
>>                        1                        0  0.0179 0.911 1.10    -9.33
>>                       -1                        1  0.6179 0.932 1.21    -7.36
>>                        0                        1  0.6548 0.920 1.14    -8.08
>>                        1                        1  0.6918 0.932 1.21    -7.28
>>  upper.CL
>>      9.20
>>      9.06
>>      6.66
>>      9.49
>>     10.27
>>      9.37
>>      8.59
>>      9.39
>>      8.67
>>
>> Results are averaged over the levels of: L1
>> Degrees-of-freedom method: satterthwaite
>> Confidence level used: 0.95
```

```
# Pairwise contrasts to compare effects across cultural profiles
pairs(emm_culture, adjust = "bonferroni")
```

```
>>  contrast
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - (identify_eng_culture_z0 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - (identify_eng_culture_z1 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - (identify_eng_culture_z1 identify_spa_culture_
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture_
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_z
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture_
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z0 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_z
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture_
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_z
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - (identify_eng_culture_z-1 identify_spa_culture_
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z1 identify_spa_culture_z-1) - identify_eng_culture_z1 identify_spa_culture_z
>>  (identify_eng_culture_z-1 identify_spa_culture_z0) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z-1 identify_spa_culture_z0) - identify_eng_culture_z1 identify_spa_culture_z
>>  (identify_eng_culture_z-1 identify_spa_culture_z0) - (identify_eng_culture_z-1 identify_spa_culture_
>>  (identify_eng_culture_z-1 identify_spa_culture_z0) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z-1 identify_spa_culture_z0) - identify_eng_culture_z1 identify_spa_culture_z
>>  identify_eng_culture_z0 identify_spa_culture_z0 - identify_eng_culture_z1 identify_spa_culture_z0
>>  identify_eng_culture_z0 identify_spa_culture_z0 - (identify_eng_culture_z-1 identify_spa_culture_z1)
```

```
>>  identify_eng_culture_z0 identify_spa_culture_z0 - identify_eng_culture_z0 identify_spa_culture_z1
>>  identify_eng_culture_z0 identify_spa_culture_z0 - identify_eng_culture_z1 identify_spa_culture_z1
>>  identify_eng_culture_z1 identify_spa_culture_z0 - (identify_eng_culture_z-1 identify_spa_culture_z1)
>>  identify_eng_culture_z1 identify_spa_culture_z0 - identify_eng_culture_z0 identify_spa_culture_z1
>>  identify_eng_culture_z1 identify_spa_culture_z0 - identify_eng_culture_z1 identify_spa_culture_z1
>>  (identify_eng_culture_z-1 identify_spa_culture_z1) - identify_eng_culture_z0 identify_spa_culture_z
>>  (identify_eng_culture_z-1 identify_spa_culture_z1) - identify_eng_culture_z1 identify_spa_culture_z
>>  identify_eng_culture_z0 identify_spa_culture_z1 - identify_eng_culture_z1 identify_spa_culture_z1
>>  estimate    SE df t.ratio p.value
>>    0.4634 0.187 51   2.472  0.6049
>>    0.9268 0.375 51   2.472  0.6049
>>   -0.1736 0.132 51  -1.318  1.0000
>>    0.0397 0.153 51   0.259  1.0000
>>    0.2529 0.261 51   0.969  1.0000
>>   -0.3471 0.263 51  -1.318  1.0000
>>   -0.3841 0.247 51  -1.556  1.0000
>>   -0.4211 0.315 51  -1.336  1.0000
>>    0.4634 0.187 51   2.472  0.6049
>>   -0.6370 0.251 51  -2.537  0.5147
>>   -0.4237 0.157 51  -2.700  0.3375
>>   -0.2105 0.158 51  -1.336  1.0000
>>   -0.8105 0.355 51  -2.286  0.9511
>>   -0.8475 0.314 51  -2.700  0.3375
>>   -0.8845 0.344 51  -2.574  0.4678
>>   -1.1004 0.423 51  -2.601  0.4372
>>   -0.8872 0.310 51  -2.862  0.2194
>>   -0.6739 0.228 51  -2.962  0.1669
>>   -1.2739 0.502 51  -2.537  0.5147
>>   -1.3109 0.454 51  -2.886  0.2054
>>   -1.3479 0.455 51  -2.962  0.1669
>>    0.2132 0.139 51   1.534  1.0000
>>    0.4264 0.278 51   1.534  1.0000
>>   -0.1736 0.132 51  -1.318  1.0000
>>   -0.2105 0.158 51  -1.336  1.0000
>>   -0.2475 0.281 51  -0.880  1.0000
>>    0.2132 0.139 51   1.534  1.0000
>>   -0.3868 0.223 51  -1.732  1.0000
>>   -0.4237 0.157 51  -2.700  0.3375
>>   -0.4607 0.215 51  -2.146  1.0000
>>   -0.6000 0.348 51  -1.724  1.0000
>>   -0.6370 0.251 51  -2.537  0.5147
>>   -0.6739 0.228 51  -2.962  0.1669
>>   -0.0370 0.153 51  -0.242  1.0000
>>   -0.0739 0.306 51  -0.242  1.0000
>>   -0.0370 0.153 51  -0.242  1.0000
>>
>> Results are averaged over the levels of: L1
>> Degrees-of-freedom method: satterthwaite
>> P value adjustment: bonferroni method for 36 tests
```

Reproducibility Information

This section lists the versions of R and RStudio, the programming language and the integrated development environment used for the creation of this documentation file. It also provides the versions of all R packages used in the analysis, ensuring reproducibility of the results presented here. The data can be found in the directory called `data/`.

```
# Display working environment versions
cat("Working Environment Versions", "\n")
```

```
>> Working Environment Versions
```

```
cat("Version of R:", as.character(getRversion()), "\n")
```

```
>> Version of R: 4.5.1
```

```
cat("Version of RStudio:", as.character(rstudioapi::versionInfo()$version), "\n\n")
```

```
>> Version of RStudio: 2025.9.0.387
```

```
# Display package versions
cat("Package Versions", "\n")
```

```
>> Package Versions
```

```
cat("`tidyverse` package version:", as.character(packageVersion("tidyverse")), "\n")
```

```
>> `tidyverse` package version: 2.0.0
```

```
cat("`patchwork` package version:", as.character(packageVersion("patchwork")), "\n")
```

```
>> `patchwork` package version: 1.3.2
```

```
cat("`showtext` package version:", as.character(packageVersion("showtext")), "\n")
```

```
>> `showtext` package version: 0.9.7
```

```
cat("`lme4` package version:", as.character(packageVersion("lme4")), "\n")
```

```
>> `lme4` package version: 1.1.37
```

```
cat("`lmerTest` package version:", as.character(packageVersion("lmerTest")), "\n")
```

```
>> `lmerTest` package version: 3.1.3
```

```
cat("`buildmer` package version:", as.character(packageVersion("buildmer")), "\n")
```

```
>> `buildmer` package version: 2.12
```

References

- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language, 68*(3), 255–278. https://doi.org/10.1016/j.jml.2012.11.001
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01
- Birdsong, D., Gertken, L. M., & Amengual, M. (2012). Bilingual Language Profile: An easy-to-use instrument to assess bilingualism. *Cognitive Linguistics, 23*(3), 381–410. https://doi.org/10.1515/cogl.2012.011
- Bullock, B. E., & Toribio, A. J. (Eds.). (2009). *The Cambridge Handbook of Linguistic Code-switching*. Cambridge University Press.

- Erker, D. (2022). How social salience can illuminate the outcomes of linguistic contact: Data from Spanish in Boston. In G. R. Guy & K. Beaman (Eds.), *The coherence of linguistic communities: Orderly heterogeneity and social meaning* (pp. 145–162). Routledge.
- Flege, J. E., & Eefting, W. (1987). Cross-language switching in stop consonant perception and production by Dutch speakers of English. *Speech Communication, 6*(3), 185–202. https://doi.org/10.1016/0167-6393(87)90025-2
- Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software, 82*(13), 1–26. https://doi.org/10.18637/jss.v082.i13
- Lenth, R. V. (2025). *emmeans: Estimated Marginal Means, aka Least-Squares Means* (R package version 1.11.2). Retrieved from https://CRAN.R-project.org/package=emmeans
- Pedersen, T. L. (2020). *patchwork: The composer of ggplots* (R package version 1.1.2). https://patchwork.data-imaginist.com/
- Podesva, R. J. (2011). Salience and the social meaning of declarative contours: Three case studies of gay professionals. *Journal of English Linguistics, 39*(3), 233–264. https://doi.org/10.1177/0075424211405161
- R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
- Voeten, C. C. (2020). *buildmer: Stepwise elimination and term reordering for mixed-effects regression* (R package version 1.8). https://cran.r-project.org/package=buildmer
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer.
- Wickham, H., François, R., Henry, L., & Müller, K. (2023). *dplyr: A Grammar of Data Manipulation* (R package version 1.1.1). https://dplyr.tidyverse.org/