Bayesian multivariate linear mixed effects models for speech research: a tutorial using `brms`

Na Hu[1], Paul-Christian Bürkner[2], & Amalia Arvaniti[1]

[1] Radboud University

[2] TU Dortmund University

Author note

Correspondence concerning this article should be addressed to Na Hu, Postbus 9103, 6500 HD Nijmegen, the Netherlands. E-mail: huna-2010@hotmail.com

Abstract

Purpose: When studying how factors influence multiple outcomes (e.g., acoustic measures in phonetics), researchers often analyze each outcome separately using a univariate approach. However, this approach ignores relationships between outcomes, which can reduce estimate accuracy and make it difficult to examine how effects are related across outcomes. A multivariate approach addresses these issues by modelling all outcomes jointly. This tutorial illustrates how to fit Bayesian multivariate linear mixed effects models using the R package `brms`.

Method: We present three example applications in phonetic research, using a corpus of Greek utterances. We focus on a rising accent, that is, a deliberate f_0 movement temporally aligned with a word's stressed syllable and used to highlight that word in speech. Specifically, we examine whether the phonetic characteristics of the rising accent depend on two factors: (a) the presence of a preceding accent within the same utterance, and (b) the location of the stress in the accented word relative to its final syllable.

Results: The multivariate approach reduced uncertainty in population-level effect estimates compared to univariate models and provided a convenient way to examine correlations among effects across outcomes.

Conclusions: This tutorial provides guidance on implementing Bayesian multivariate linear mixed effects models and demonstrates their potential.

Keywords: Bayesian, multivariate, linear mixed effects models, correlations, phonetics

Word count: 12,426

Bayesian multivariate linear mixed effects models for speech research: a tutorial using `brms`

1. Introduction

Many entities and phenomena in the world are inherently multidimensional. For example, speech signals comprise a variety of acoustic properties, including temporal characteristics (e.g., duration and speech rate), spectral characteristics (e.g., intensity, formants, and spectral tilt), and measures related to fundamental frequency (f_0). A central goal of speech production research is to understand whether and how these acoustic features encode different types of information. These include segmental categories (e.g., /b/ vs. /p/), prosodic categories (e.g., word stress and pitch accents), higher-level linguistic properties, such as information structure and discourse pragmatics, and speaker-related factors (e.g., intentions, emotions, or social identity) (see Cole, 2015 for a review).

Studies addressing these questions often use univariate linear regression models, modelling each acoustic feature separately as a function of predictors (see Sonderegger & Sóskuthy, 2025 for a review). However, this approach fails to account for covariance among acoustic features, which can reduce the accuracy of parameter estimation and uncertainty quantification. Furthermore, univariate models make it difficult, and sometimes, impossible, to address certain research questions, such as relationships between effects across acoustic dimensions.

To address these limitations, a more effective approach would be to model multiple acoustic features simultaneously using a multivariate model. Multivariate models can be fitted within either a frequentist (e.g., Honda, Clayards, & Baum, 2024) or a Bayesian framework (e.g., Foster & Cole, 2024; Hu & Arvaniti, 2024; Smith, Sonderegger, & Spade Consortium, 2024; Tanner, Sonderegger, & Stuart-Smith, 2020; see Sonderegger & Sóskuthy, 2025 for a review).

The distinctions between Bayesian and frequentist approaches have been thoroughly discussed in previous work, for example, in phonetics-focused tutorials (Nalborczyk, Batailler, Lœvenbruck, Vilain, & Bürkner, 2019; Vasishth, Nicenboim, Beckman, Li, & Kong, 2018) and in general introductions to Bayesian statistics (Kruschke, 2015; McElreath, 2020). A central difference lies in how each framework conceptualizes probability: the Bayesian approach assesses the probability of a hypothesis given the data ($P(Hypothesis \mid Data)$), whereas the frequentist approach tests the probability of observing the data given the hypothesis ($P(Data \mid Hypothesis)$). These two conceptualizations of probability are fundamentally different. For example, consider patients who show the symptom of muscle aches ($Data$). We would like to know the probability that they have kuru ($Hypothesis$), a rare disease observed among the Fore tribe in New Guinea (example adopted from Pinker, 2021). The Bayesian approach calculates this probability directly, concluding that the probability of having kuru ($Hypothesis$) given muscle aches

($Data$) is low, since kuru is rare. In contrast, the frequentist approach does not compute this probability directly; instead, it calculates the probability of having muscle aches ($Data$) given that a patient has kuru ($Hypothesis$), concluding that this probability is high because muscle aches are a common symptom of the disease. The frequentist answer is not incorrect; it is true that patients with kuru are likely to show muscle aches, but it does not answer the question we asked: how probable it is that a patient has kuru if they show muscle aches. This conceptual mismatch between the question of interest and the frequentist answer can lead to inappropriate treatment decisions in a medical context. In linguistic research, the consequences are not life-threatening but can nonetheless lead to a false understanding of the research hypothesis, which, in some fields at least, can have real-life repercussions, such as decisions about language teaching or the treatment of language-related disorders.

In terms of formal statistical inference, hypotheses are typically expressed in terms of model parameters. Consequently, within the Bayesian framework, evaluating $P(Hypothesis \mid Data)$ amounts to evaluating $P(Parameter \mid Data)$, the posterior distribution of the relevant parameters. According to Bayes' theorem, the posterior probability is conditional on two key components: first, the likelihood of the data, $P(Data \mid Parameter)$, which reflects how likely the observed data are given the parameters; and second, the prior probability of the parameters, $P(Parameter)$, representing our knowledge about the parameters before seeing the data (for a more detailed explanation, see Nalborczyk et al., 2019).

In Bayesian modelling using `Stan` or its R wrapper `brms`, the posterior distribution of model parameters is estimated using Markov Chain Monte Carlo (MCMC) methods. The result is a full posterior distribution of all model parameters, represented as a collection of samples (commonly referred to as "draws" in `Stan` and `brms`) from the posterior. Each sample corresponds to one possible joint configuration of all parameters in the model, each parameter represented by a range of plausible values rather than a single "best" estimate. The posterior distribution offers two practical advantages: (1) it provides a straightforward way to quantify uncertainty, which is central to statistical inference; (2) it enables a wide range of follow-up analyses, such as correlations between model parameters, since the parameter values are drawn from the same joint posterior distribution.

Although conceptually a straightforward extension of univariate models, the use of multivariate models remains relatively limited, especially within the Bayesian framework. This is likely due to challenges associated with model specification and interpretation, and the high computational demands involved.

A wide range of software and R packages (collections of R functions) are now available for fitting Bayesian models, including `JAGS` (Depaoli, Clifton, & Cobb, 2016), `WinBUGS` (Lunn, Thomas, Best, & Spiegelhalter, 2000),

`mcmcGLMM` (Hadfield, 2010), and `brms` (Bürkner, 2018). In particular, the R package `brms` (Bayesian regression models using Stan) provides `lme4`-like syntax for fitting Bayesian regression models via the probabilistic programming language `Stan` (Carpenter et al., 2017), allowing users to easily specify complex model structures. Moreover, using cloud computing services can greatly speed up model estimation.

In this tutorial, we focus on Bayesian multivariate linear mixed effects models, also known as hierarchical or multilevel linear models (McElreath, 2020). These models jointly analyze multiple continuous response variables as a function of predictor variables while accounting for both population-level (fixed) effects (the average effect of predictors across the population, i.e., the population or grand mean) and group-level (random) effects (how individual groups within the population, such as participants, deviate from the population mean). We illustrate three example applications of these models using phonetic data: estimation of population-level effects, that is, whether acoustic characteristics of utterances differ across experimental conditions (Application 1), analysis of correlations among population-level effects across outcomes (Application 2), and analysis of correlations among speaker-level effects across outcomes (Application 3). Although our examples are drawn from phonetic research, Bayesian multivariate linear mixed effects models are broadly applicable to data with a similar structure in other domains, such as treatment studies that evaluate the effects of an intervention on participants' responses (e.g., response times and accuracy scores).

The remainder of the paper is structured as follows. Section 2 outlines key differences between Bayesian multivariate and univariate linear mixed effects models. Section 3 describes the technical details of fitting Bayesian models using the `brms` package (Bürkner, 2018) in combination with cloud computing on Amazon EC2. Section 4 introduces the data and acoustic measurements used in this tutorial. Section 5 presents three example applications of Bayesian multivariate models to address research questions in phonetics.

2. Bayesian multivariate linear mixed effect models

We call a model a *multivariate* linear regression if it includes multiple continuous response variables. By contrast, a *univariate* linear regression model contains only a single continuous response variable. To illustrate their differences, we compare their mathematical expressions, which provide an unambiguous way to highlight the distinction between the two approaches.

Consider a speech production experiment with a repeated-measures design, where target items are produced in two conditions (e.g., focused or unfocused) by multiple speakers, with multiple items per condition, yielding a collection of utterances. For each utterance, two continuous measurements are taken, such as f_0 maximum (denoted

as $y_{1,i}$) and f_0 minimum (denoted as $y_{2,i}$). A researcher may be interested in whether, and how, these measures differ between the two experimental conditions, represented by a binary predictor x_i (coded as 1 for the focused condition). Mathematically, this requires modelling $y_{1,i}$ and $y_{2,i}$ as functions of x_i, with both intercepts and slopes allowed to vary by speaker and by item, following a maximally complex random-effects structure (Barr, Levy, Scheepers, & Tily, 2013).

In the phonetics literature, this is typically analyzed by fitting two univariate linear regression models, for $y_{1,i}$ and $y_{2,i}$ separately, as shown in the following mathematical expressions:

$$
\begin{aligned}
y_{1,i} &\sim Normal(\mu_{1,i}, \sigma_1) && (1)\\
\mu_{1,i} &= \alpha_1 + \alpha_{1,\text{speaker}[i]} + \alpha_{1,\text{item}[i]} + (\beta_1 + \beta_{1,\text{speaker}[i]} + \beta_{1,\text{item}[i]})x_i && (2)\\
\begin{bmatrix}\alpha_{1,\text{speaker}[j]}\\ \beta_{1,\text{speaker}[j]}\end{bmatrix} &\sim \text{MVN}(\mathbf{0}, \Sigma_{1,\text{speaker}}) && (3)\\
\begin{bmatrix}\alpha_{1,\text{item}[k]}\\ \beta_{1,\text{item}[k]}\end{bmatrix} &\sim \text{MVN}(\mathbf{0}, \Sigma_{1,\text{item}}) && (4)
\end{aligned}
$$

$$
\begin{aligned}
y_{2,i} &\sim Normal(\mu_{2,i}, \sigma_2) && (5)\\
\mu_{2,i} &= \alpha_2 + \alpha_{2,\text{speaker}[i]} + \alpha_{2,\text{item}[i]} + (\beta_2 + \beta_{2,\text{speaker}[i]} + \beta_{2,\text{item}[i]})x_i && (6)\\
\begin{bmatrix}\alpha_{2,\text{speaker}[j]}\\ \beta_{2,\text{speaker}[j]}\end{bmatrix} &\sim \text{MVN}(\mathbf{0}, \Sigma_{2,\text{speaker}}) && (7)\\
\begin{bmatrix}\alpha_{2,\text{item}[k]}\\ \beta_{2,\text{item}[k]}\end{bmatrix} &\sim \text{MVN}(\mathbf{0}, \Sigma_{2,\text{item}}) && (8)
\end{aligned}
$$

where $y_{1,i}$ and $y_{2,i}$ are modelled as independent normally distributed variables (Equations [1] and [5]), each with its own mean, $\mu_{1,i}$ and $\mu_{2,i}$, and standard deviation σ_1 and σ_2. The means $\mu_{1,i}$ and $\mu_{2,i}$ are modelled as linear functions of the dummy variable x_i (Equations [2] and [6]). The population-level (fixed) effects are the intercepts α_1 and α_2 and the slopes β_1 and β_2. The intercepts correspond to the expected values for $y_{1,i}$ and $y_{2,i}$ in the baseline condition, while the slopes capture the expected difference between focused and unfocused conditions in y_{1i} and y_{2i}. Additionally, the models include group-level (random) effects that allow intercepts and slopes to vary across speakers and items: for $y_{1,i}$, these are $\alpha_{1,\text{speaker}[i]}$, $\alpha_{1,\text{item}[i]}$, $\beta_{1,\text{speaker}[i]}$, and $\beta_{1,\text{item}[i]}$; for $y_{2,i}$, they are $\alpha_{2,\text{speaker}[i]}$, $\alpha_{2,\text{item}[i]}$, $\beta_{2,\text{speaker}[i]}$, and $\beta_{2,\text{item}[i]}$. These random effects represent speaker- and item-specific deviations from the corresponding fixed effects for each response. Within each univariate model and within each grouping factor, the random effects are modelled with a multivariate normal (`MVN`) distribution with a zero mean vector and a covariance matrix, as shown in Equations (3), (4), (7), and (8). For example, Equation (3) expresses that, for the response variable $y_{1,i}$, the speaker-level random intercepts ($\alpha_{1,\text{speaker}[i]}$) and slopes ($\beta_{1,\text{speaker}[i]}$) follow an `MVN` distribution with a zero mean vector and covariance matrix $\Sigma_{1,\text{speaker}}$, which represents the variance-covariance structure of the

speaker-level random effects for $y_{1,i}$. Similarly, $\Sigma_{1,\text{item}}$ in Equation (4) denotes the item-level variance-covariance matrix for the response variable $y_{1,i}$.

Instead of fitting two separate univariate models, we can model both outcome variables jointly with a joint error structure using a multivariate linear regression model. This multivariate approach also allows group-level effects to be modelled as correlated across responses (Bürkner, 2018), rather than only within each response as in the univariate case. These structures can be easily specified using the `brms` package; the code is provided in Section 5.1.2. Here, we focus only on the mathematical expression for the multivariate model, which is given below:

$$\begin{pmatrix} y_{1,i} \\ y_{2,i} \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} \mu_{1,i} \\ \mu_{2,i} \end{pmatrix}, \Sigma\right) \tag{9}$$

$$\mu_{1,i} = \alpha_1 + \alpha_{1,\text{speaker}[i]} + \alpha_{1,\text{item}[i]} + \left(\beta_1 + \beta_{1,\text{speaker}[i]} + \beta_{1,\text{item}[i]}\right)x_i \tag{10}$$

$$\mu_{2,i} = \alpha_2 + \alpha_{2,\text{speaker}[i]} + \alpha_{2,\text{item}[i]} + \left(\beta_2 + \beta_{2,\text{speaker}[i]} + \beta_{2,\text{item}[i]}\right)x_i \tag{11}$$

$$\Sigma = \begin{bmatrix} \sigma_{y1} & 0 \\ 0 & \sigma_{y2} \end{bmatrix} \mathbf{R} \begin{bmatrix} \sigma_{y1} & 0 \\ 0 & \sigma_{y2} \end{bmatrix} \tag{12}$$

$$\mathbf{R} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} \alpha_{1,\text{speaker}[j]} \\ \beta_{1,\text{speaker}[j]} \\ \alpha_{2,\text{speaker}[j]} \\ \beta_{2,\text{speaker}[j]} \end{bmatrix} \sim \text{MVN}\left(\mathbf{0}, \Sigma_{\text{speaker}}\right) \tag{14}$$

$$\begin{bmatrix} \alpha_{1,\text{item}[k]} \\ \beta_{1,\text{item}[k]} \\ \alpha_{2,\text{item}[k]} \\ \beta_{2,\text{item}[k]} \end{bmatrix} \sim \text{MVN}(\mathbf{0}, \Sigma_{\text{item}}) \tag{15}$$

where $y_{1,i}$ and $y_{2,i}$ are simultaneously modelled with a joint error structure, rather than separately as in the univariate models. Specifically, the two outcome variables follow a multivariate normal (`MVN`) distribution with a vector of means ($\mu_{1,i}$ and $\mu_{2,i}$) and a shared error covariance matrix Σ, as shown in Equation (9). Unlike in the univariate case, where each outcome has its own standard deviation (σ_1 and σ_2), the matrix Σ represents the full data-level residual variance-covariance structure. Importantly, Σ is not the descriptive covariance between y_{1i} and y_{2i}, but rather the residual covariance matrix estimated from the data (McElreath, 2020). Σ can be decomposed into the standard deviations σ_{y1} and σ_{y2} and a residual correlation matrix $\mathbf{R}$ as shown in (12), with $\mathbf{R}$ defined in (13), where ρ represents the residual correlation between $y_{1,i}$ and $y_{2,i}$ (i.e., f_0 maximum and f_0 minimum), indicating how strongly the residuals of the two response variables correlated after accounting for the predictors (Bürkner, 2025).

As in the univariate models, $\mu_{1,i}$ and $\mu_{2,i}$ get their own linear models, yielding two μ definitions containing fixed and random effects (Equations [10] and [11]). However, unlike the univariate approach, the random effects across outcome dimensions within each grouping factor are assumed to have a multivariate prior distribution.

Specifically, the random effects for speakers—$\alpha_{1,\text{speaker}[j]}$, $\beta_{1,\text{speaker}[j]}$, $\alpha_{2,\text{speaker}[j]}$, and $\beta_{2,\text{speaker}[j]}$—are modelled jointly with a multivariate normal distribution with a zero mean vector and covariance matrix Σ_{speaker}, which captures speaker-level variance components, as shown in Equation (14). The same structure applies to the random effects for items, with Σ_{item} capturing item-level variance components, as in (15).

To summarize, the univariate and multivariate models differ in the following ways:

1. **Residual structure**. The univariate approach models each outcome ($y_{1,i}$ and $y_{2,i}$) separately with independent normal distributions. As a result, each outcome has its own residual variance parameter. By contrast, the multivariate approach models the outcomes jointly as a multivariate normal distribution with a shared residual covariance matrix, which captures the unexplained correlation among outcomes. It has been shown that multivariate Bayesian methods can improve parameter estimation and prediction accuracy compared to univariate methods (Brown, Fearn, & Vannucci, 1999; Li, Ghosh, & Villarini, 2023; Wu, Goldfeld, Petkova, & Park, 2024).
2. **Correlations among random effects across outcomes.** The multivariate model also estimates correlations among random effects (e.g., by-speaker intercepts and slopes) across response variables. This includes correlations between by-speaker random intercepts for y_1 and y_2, between random intercepts and random slopes of x_i for y_1 and y_2, and between random slopes of x_i for y_1 and y_2. Such correlations are especially informative for examining speaker-specific strategies in responding to a given effect, such as the presence of focus.
3. **Posterior distributions.** In the two univariate models, the parameters are estimated under separate posterior distributions, so the samples drawn from the two posteriors are independent from each other. In contrast, in the multivariate model, the model parameters from the individual sub-models are estimated jointly under a single posterior distribution. This makes it possible to examine correlations among parameters across sub-models, such as β_1, which represents the effect of predictor x_i on y_1, and β_2, which represents the effect of the same predictor on y_2. This feature is particularly useful in phonetic research as it enables the analysis of cue-weighting (trading or enhancing) across different acoustic dimensions at the population level (across speakers).

Taken together, these features make multivariate models particularly suitable for investigating research questions involving multiple, potentially correlated outcomes. These models improve parameter estimation and enable analyses that are not possible with univariate approaches.

3. Fitting `brms` models on Amazon EC2

In this tutorial, we demonstrate how to fit Bayesian multivariate linear mixed effects models in `R` using the `brms` package (Bürkner, 2018), the wrapper package for the probabilistic programming language `Stan` (Carpenter et al., 2017). Fitting Bayesian models can be computationally intensive, and using computers with powerful hardware (e.g., a large number of CPUs, large memory) can significantly reduce model fitting time. However, such machines are often expensive and impractical to maintain locally. Cloud computing offers a more cost-efficient alternative. It allows users to easily scale computing power as needed (up to hundreds of CPUs, while most personal computers have four) and pay only for what they use. Many companies provide cloud services, and many universities maintain their own high-performance computing infrastructure. Users can select the option that complies with any data-sharing restrictions. For this tutorial, we used Amazon Elastic Compute Cloud (Amazon EC2) (https://aws.amazon.com/ec2/), a service provided by Amazon Web Services. EC2 offers a wide range of virtual machines (called "instances") with varying CPU and memory configurations.

For our analyses, we used the `c6a.8xlarge` instance, which has 32 virtual CPUs, 64 GiB of memory, and an on-demand hourly rate of $1.224 (https://aws.amazon.com/ec2/pricing/on-demand/). On this instance, we ran the RStudio Server Amazon Machine Image (AMI) provided at https://www.louisaslett.com/RStudio_AMI/, which deploys RStudio version 1.3.1073 and R version 4.0.2. Thorough instructions for setting up this AMI are available: a video guide can be found on the website and written instructions are provided by Schröder (2019). Here, we note two important points. First, once an instance is no longer in use, it must be "stopped". This can be done by opening the Amazon EC2 console by choosing "Stop instance". Otherwise, costs will continue to accumulate. Second, once an instance is stopped, the data on any instance store volumes is erased; thus, it is important to ensure all data is saved to persistent storage before stopping an instance (for more information on these two points, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Stop_Start.html).

Once the RStudio server AMI was ready, we installed the required R packages and ran the analysis in the same way as in RStudio Desktop. For Bayesian modelling, we used the `brms` package (version 2.22.0). To enable within-chain parallelization, we used `CmdStanR` (version 0.9.0, Gabry, Češnovar, Johnson, & Bronder, 2025) as the backend. The `CmdStanR` package is an R interface to `cmdstan` (Stan Development Team, 2025), the command line interface to `Stan`. As such, `CmdStanR` requires a working installation of `cmdstan`, and provides a convenient way to install it (for instructions see https://mc-stan.org/cmdstanr/articles/cmdstanr.html). In our setup, `cmdstan` version 2.36

was used. Details on how to set up within-chain parallelization and the resulting improvement on model runtime are provided in later sections.

To reiterate, cloud computing (in our case, Amazon EC2) and within-chain parallelization (via `CmdStanR` and `cmdstan`) were used solely to reduce model runtime. Neither is required for fitting Bayesian multivariate models, nor do they affect results. If long model runtimes are not a concern, the models can be fitted using only RStudio Desktop and the `brms` package, without within-chain parallelization.

4. Introducing the data

We demonstrate three applications of Bayesian multivariate linear mixed effects models using data on the Greek rising accent from Arvaniti, Katsika, and Hu (2024) (298 instances, produced by 13 native speakers of Standard Greek; 10 females; mean age = 34; publicly available at https://osf.io/sjxuq/). An accent is a deliberate pitch movement synchronized with the stressed syllable of the word it highlights. The Greek rising accent under investigation[1] is used to highlight an item that corrects or contrasts with something previously said or implied. For example, the accent would occur on the first syllable of *yellow* in the phrase *it is YELLOW*, signalling either a correction (not blue, as previously said) or a contrast (not some other colour). In Arvaniti et al. (2024), these accents were elicited using a question-answer task, where the questions were designed to elicit corrective responses, typically produced with the rising accent under investigation (e.g., – "Did you say their son's eyes are brown?" – "Blue.").

The research question addressed in Arvaniti et al. (2024) was whether the phonetic characteristics of the rising accent depended on two factors. The first factor was the presence of a preceding accent within the same utterance, coded as a categorical variable called accentual context (`AC`). This factor was embedded in the dialogue task by manipulating the number of content words in the responses, which consisted of one or two words. In one-word responses, the target word was the only content word, which carried the target accent; a preceding accent was not possible; thus, `AC` was absent. In two-word responses, the target word was the second content word and as such it was preceded by another rising accent on the first content word; thus, `AC` was present. The second factor was the location of word stress in the accented word relative to its final syllable (word stress location, `STR`). This factor was

[1] The rising accent is autosegmentally represented as L+H* (e.g., Arvaniti et al., 2024). In this dataset, the "accent" actually refers to the accent on the highlighted word together with the following L-L% edge tones. For convenience, we refer to this combination as "accent".

implemented in the task by including target words with varying stress patterns: stress on the third-to-last syllable (antepenultimate, `ante`, as in [laðoˈ**le**mono]), the second-to-last syllable (penultimate, `penult`, as in [lemoˈ**na**ða]), or the final syllable (`final`, as in [ɣalaˈ**na**]).

Two phonetic properties of the target accents were examined: the f_0 contour shape and segmental duration. Both properties were of interest because they interacted under the conditions represented in the task (see Roettger & Grice, 2019). Moreover, there was reason to believe that f_0 and duration were not independent of one another, as duration was used to enhance accents (Arvaniti et al., 2024), although strictly speaking, duration did not pertain to the accent itself.

In the analysis, segmental duration was represented as the duration of the target word (`durWord`), a continuous variable. The f_0 contour shape was represented using Functional Principal Component Analysis (FPCA) (Gubian, Torreira, & Boves, 2015). FPCA represents each input contour as a mathematical function (see Equation [16]), which consists of the overall mean contour ($\mu(t)$) and a set of principal component curves (e.g., $PC1(t)$ and $PC2(t)$) that capture the dominant modes of variation across curves, such as differences in slope or scaling. Each PC is associated with a score (e.g., s_1 and s_2) that reflects its contribution to reconstructing the original curve. While the mean curve and PC curves are shared across all input curves, the PC scores are specific to each input curve and thus characterize its shape.

$$f(t) \approx \mu(t) + s_1 \times PC1(t) + s_2 \times PC2(t) + \cdots \qquad (16)$$

The FPCA was performed by Arvaniti et al. (2024). They reported that, for the Greek rising accent, PC1 captured curve scaling, with higher scores indicating higher scaling and lower scores indicating lower scaling, while PC2 reflected the magnitude of the final drop. In this tutorial, we directly adopted their FPCA output and used the PC1 and PC2 scores (i.e., s_1 and s_2 in Equation [16], coded as `PC1` and `PC2` for clarity) in the analysis.

In summary, the data included two categorical predictor variables, word stress location (`STR`) and accentual context (`AC`), and three continuous response variables: word duration (`durWord`), scaling of the f_0 contour (`PC1`), and magnitude of the final f_0 drop (`PC2`).

5. Example applications

5.1. Application 1: population-level effects

In this application, we demonstrate that multivariate models improve parameter estimation compared to univariate models, focusing on population-level slopes, that is, the expected difference between experimental conditions at the population level.

5.1.1. Research question.

The research question we aim to address is, at the population level, whether, and to what extent, the shape and duration of the rising accent are influenced by the presence or absence of a preceding accent and by word stress location.

This involves testing the population-level effects of `AC` and `STR` on the three acoustic features: PC1, PC2, and duration. A common statistical approach would be to analyze each feature separately using univariate linear mixed effects models, each with only one dependent variable. Instead, we propose fitting a multivariate model that includes all three features simultaneously, which can yield more precise parameter estimates. In what follows, we compare univariate and multivariate models to show that the latter provide better accuracy.

5.1.2. Model specification.

We first describe the specification of the univariate models, followed by the multivariate model.

Three separate univariate models were fitted, one for each response variable: `durWord` (duration of the target words), and `PC1` and `PC2` (the PC1 and PC2 scores of the f_0 curves), all on the original scale.

```
Greek_uni_PC1_m1 <- brm(PC1 ~ STR + AC + (STR + AC | Speaker), data = Greek_data,
    family = gaussian(), chains = 4, cores = 4, warmup = 1000, iter = 3000, control = list(adapt_delta = 0
.99,
        max_treedepth = 10), save_pars = save_pars(all = TRUE), backend = "cmdstanr",
    threads = threading(8))
Greek_uni_PC2_m1 <- brm(PC2 ~ STR + AC + (STR + AC | Speaker), ...)

Greek_uni_durWord_m1 <- brm(durWord ~ STR + AC + (STR + AC | Speaker), ...)
```

The predictor variables in each model were stress location (`STR`) and accentual context (`AC`). `STR` had three levels: antepenultimate (`ante`, reference level), penultimate (`penult`), and final (`final`). `AC` had two level: absent (`abs`, reference level) and present (`pres`). We did not include an interaction between `STR` and `AC`, as they reflect distinct sources of pressure at different time points and are therefore unlikely to interact. The grouping factor was speaker

(`Speaker`), with by-speaker random intercepts and slopes for both predictors. We did not include `Item` as a grouping factor, as its variation was fully accounted for by the combination of `STR` and `AC`.

We did not specify custom priors for these models (hence the absence of a prior parameter in the model specification), as we did not have clear prior knowledge of the direction or magnitude of effects (but see Vasishth et al., 2018 for discussion of the influence of priors on posterior estimates). Instead, we used the default priors provided by `brms`, which can be inspected using the `get_prior()` function. Population-level (fixed) slopes were assigned `flat` (uninformative) priors. Intercepts followed weakly informative Student's *t* distributions: `student_t(3, 0.5, 2.5)` for `durWord`, `student_t(3, -1.2, 15.9)` for `PC1`, and `student_t(3, 0.4, 9.8)` for `PC2`. Group-level standard deviations (`sd`) and residual standard deviations (`sigma`) were also given Student's *t* distributions. Correlation structures among group-level effects were assigned an `LKJ(1)` prior.

Each model was fitted using 4 Markov chains (`chains`) with 3,000 iterations (`iter`) each, including 1000 warm-up iterations (`warmup`) to let the chains stabilize, and 4 cores; for more detailed explanations of these parameters, see https://www.rdocumentation.org/packages/brms/versions/2.22.0/topics/brm. Since we had access to 32 vCPUs, we enabled within-chain parallelization by allocating 8 vCPUs per chain using `threads = threading(8)` with `cmdstanr` as the backend (`backend = "cmdstanr"`) (these two parameters can be omitted if within-chain parallelization is not used). As a rule of thumb, the `cores` argument specifies the number of MCMC chains to run in parallel (in our case, 4), and the `threads` argument specifies the number of threads (CPUs) to use within each chain (in our case, 8, 32 total CPUs divided by 4 chains) (Weber & Bürkner, 2025). Therefore, the total number of CPUs used is `cores`×`threads` (Gabry et al., 2025).

We next describe the specification of the multivariate model, as shown in the chunk below.

```
PC1_m <- bf(PC1 ~ STR + AC + (STR + AC | p | Speaker))

PC2_m <- bf(PC2 ~ STR + AC + (STR + AC | p | Speaker))

durWord_m <- bf(durWord ~ STR + AC + (STR + AC | p | Speaker))

Greek_mv_m1 <- brm(PC1_m + PC2_m + durWord_m + set_rescor(TRUE), data = Greek_data,
    family = gaussian(), chains = 4, cores = 4, warmup = 1000, iter = 3000, control = list(adapt_delta = 0
.99,
        max_treedepth = 10), save_pars = save_pars(all = TRUE), backend = "cmdstanr",
    threads = threading(8))
```

The multivariate model, `Greek_mv_m1`, included all three response variables: `PC1`, `PC2`, and `durWord`. Essentially, we have three sub-models, each corresponding to one response variable and using the same linear function as in the univariate models. As a first step, we specified each sub-model separately in a `bf()` function and saved them as objects. Next we combined the three `bf()` objects with the `+` operator within the `brm()` function. We also specified `set_rescor(TRUE)` to instruct `brms` to estimate the residual correlations among `PC1`, `PC2`, and `durWord`. Note that if `set_rescor(FALSE)` is used instead, `brms` will completely ignore the residual correlation structure. Currently, it is not possible to estimate only a partial residual correlation structure (e.g., allowing only correlations between the residuals of `PC1` and `PC2`, but excluding the other pairs). The term `(STR + AC|p|Speaker)` indicates varying intercepts and varying slopes for predictors `STR` and `AC` across levels of `Speaker`. By writing `|p|` in between, we indicate that all random effects for `Speaker` should be modeled as correlated. The indicator `p` is arbitrary and can be replaced by any other symbols (e.g., `a`, `b`, `x`, `y`, etc.). Alternatively, grouping terms in `brms` can also be set up using the `gr()` function, so the expression `(STR + AC|p|Speaker)` is equivalent to `(STR + AC|gr(Speaker, id = "p"))`, which states that all group-level terms across the sub-models with the same `id` will be modeled as correlated. For more details on the multilevel syntax of `brms`, see `help("brmsformula")` and `vignette("brms_multilevel")`. If additional random effects (e.g., `Item`) are modeled as correlated, a different symbol than the one used for `Speaker` must be used. Otherwise, `brms` would throw an error, as it does not make sense to "correlate" items and speakers—they are different entities.

Because all three sub-models share the same formula (`~ STR + AC + (STR + AC|p|Speaker)`), a shorthand specification is also possible with the `mvbind` notation, which instructs `brms` to jointly model `PC1`, `PC2`, and `durWord`.

```
Greek_mv_m1 <- brm(bf(mvbind(PC1, PC2, durWord) ~ STR + AC + (STR + AC | p | Speaker)) +
    set_rescor(TRUE), ...)
```

Note that `mvbind` only works when all sub-models share the same formula. If any sub-model has a different formula from the others, then we need to use the previous approach to specify each formula separately.

The model was fitted using default priors and the same settings as the univariate models (4 chains of 3,000 iterations each, including 1,000 warm-ups, and 4 cores). As before, we allocated the total 32 vCPUs across the 4 parallel chains by assigning 8 CPUs per chain using `threads = threading(8)` with `cmdstanr` as the backend. With this setup, the total execution time for this model was 150 seconds (2.5 minutes). For comparison, running the same model without within-chain parallelization using only 4 CPUs required 1,345 seconds (22.4 minutes). Thus, within-chain parallelization reduced execution time by 89%, corresponding to a savings of 1,195 seconds (19.9 minutes).

While this amount of time reduction may seem modest, for models with more complex structure and longer runtime, the benefit is expected to be more substantial. For instance, for a model including five dependent variables and one categorical independent variable with varying intercepts and slopes by speaker and item (Kim, Hu, & Arvaniti, in preparetion), the runtime without within-chain parallelization was 25,072 seconds (approximately 7 hours); by enabling within-chain parallelization, it was reduced to 1,898 seconds (0.5 hours), a 92% reduction.

5.1.3. Model summary structure.

The estimates from the multivariate model can be retrieved using the following command, with the output provided in Appendix 1.

```
summary(Greek_mv_m1)
```

The model summary consists of five sections. The first section provides general model information (`Formula`, `Data`, `Draws`, etc.), similar to that of a univariate model. The following sections contain parameter estimates, which are all summarized by their posterior mean (`Estimate`) along with the lower (2.5%) and upper (97.5%) bounds of the 95% credible interval (CrI). The second section, `Multilevel Hyperparameters`, contains the group-level effect estimates. In our case, it includes only one sub-section: `~Speaker`, as `Speaker` is our only grouping variable in the model. If additional grouping variables had been included in the formula (e.g., `Item`), each would have its own subsection (e.g., `~Item`). Note that all parameter names now have the name of the dependent variable as a prefix (`PC1_`, `PC2_`, or `durWord_`) to indicate the sub-model to which they belong. Since we instructed `brms` to estimate correlations among all random effects for `Speaker`, not only within the same acoustic measure, but also across measures, these correlation estimates also appear in this section. For example, the first row starting with `cor`, `cor(PC1_Intercept,PC1_STRfinal)`, represents the correlation between by-speaker intercepts for `PC1` and by-speaker slopes of `STRfinal` for `PC1`. In other words, it captures how the intercepts for `PC1` and the slopes of `STRfinal` for `PC1` co-vary at the speaker level. We will return to this in Application 3 (Section 5.3). Following the `Multilevel Hyperparameters` section is the `Regression Coefficients` section, which reports the population-level (fixed) effect estimates. Their interpretation is consistent with those from the univariate models. Consider the effects beginning with `PC1_`. `PC1_Intercept` (the intercept term α) represents the mean PC1 score when both `STR` and `AC` are at their reference levels (`STR` = `antep`, `AC` = `Absent`); `PC1_STRfinal` (the coefficient β_1) indicates the difference in PC1 scores between `STRfinal` and the reference level, when `AC` is at the reference level (`Absent`); `PC1_STRpenult` (the coefficient β_2) is the difference in PC1 scores between `STRpenult` and the reference level, when `AC` is `Absent`; and `PC1_ACpres` (the coefficient β_3) is the difference in PC1 scores between `ACpres` and the reference level (`ACabs`) when `STR` is at its

reference level (`antepenultimate`). The `Further Distributional Parameters` section contains estimates of the standard deviations of the three dependent variables. Finally, the `Residual Correlations` section reports the residual correlations among them. These correlations are included because we set `set_rescor(TRUE)`.

5.1.4. Model estimates comparison.

To address the research question, that is, whether, and to what extent, `AC` and `Stress` affect the curve shape and duration of the rising accent at the population level, we focused on the population-level effect estimates and compared these estimates between the multivariate and univariate models.

We first extracted the posterior summaries of the population-level (fixed) effects from each model using the `fixef()` function from the `brms` package .

```
# Multivariate model
Greek_mv_m1_fixef <- fixef(Greek_mv_m1)
# Univariate models
Greek_uni_PC1_m1_fixef <- fixef(Greek_uni_PC1_m1)
Greek_uni_PC2_m1_fixef <- fixef(Greek_uni_PC2_m1)
Greek_uni_durWord_m1_fixef <- fixef(Greek_uni_durWord_m1)
```

This resulted in four separate data frames. To compare estimates between the multivariate and univariate models, we combined and wrangled these data frames using functions from the `dpylr` package (Wickham, François, Henry, Müller, & Vaughan, 2025) (code not shown, as it does not involve any Bayesian-specific operations). The first few rows of the resulting combined data frame are shown below.

```
##               Estimate Est.Error   Q2.5  Q97.5 Feature    Effect        model CrI_len
## PC1_STRfinal      1.30      1.87  -2.32   4.96     PC1  STRfinal Multivariate    7.28
## PC1_STRpenult    -0.28      2.09  -4.50   3.83     PC1 STRpenult Multivariate    8.33
## PC1_ACpres       16.30      2.11  12.11  20.55     PC1    ACpres Multivariate    8.43
## PC2_STRfinal    -17.48      1.51 -20.48 -14.50     PC2  STRfinal Multivariate    5.98
```

We then computed the width of the 95% CrI (`CrI width`) for each effect by subtracting the lower bound (Q2.5) from the upper bound (Q97.5). The amount by which the 95% CrI narrowed, referred to as the shrinkage, when moving from a univariate to a multivariate model was computed using Equation (17). Positive values indicate that the univariate model's CrI is wider compared to the multivariate model (in other words, the CrI shrinks in the multivariate model), whereas negative values indicate the opposite.

$$\text{Shrinkage} = \frac{\text{CrI width}_{\text{uni}} - \text{CrI width}_{\text{multi}}}{\text{CrI width}_{\text{uni}}} \times 100 \qquad (17)$$

We then used the `ggplot2` package (Wickham, 2016) to visualize the mean estimates, their 95% CrIs, as well as the shrinkage (Figure 1). The full code is available on OSF.

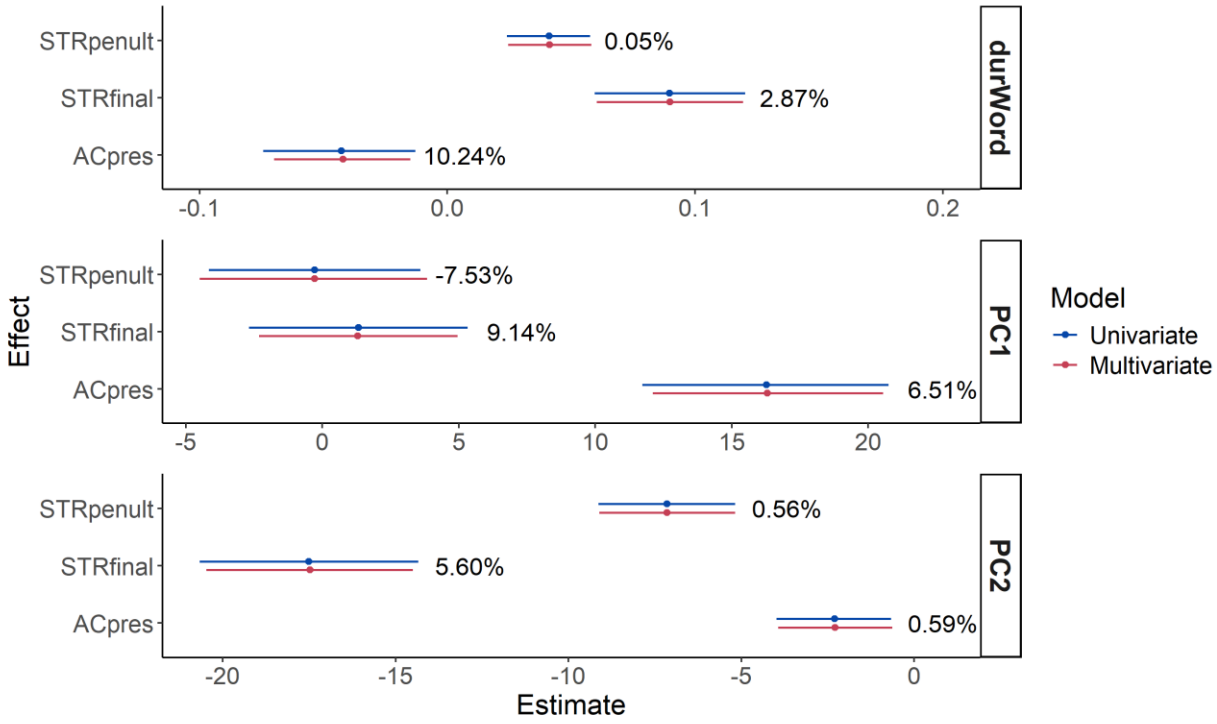

Figure 1: Comparison of population-level effect estimates from univariate (blue) and multivariate (red) models across three outcome variables: durWord, PC1, and PC2.

In Figure 1, each panel corresponds to one of the dependent variables, with estimated effects (`ACpres`, `STRpenult`, and `STRfinal`) shown on the y-axis and their estimates on the x-axis. Horizontal lines indicate the 95% CrIs, and points represent the posterior means. Percentages next to the estimates show the shrinkage in 95% CrI width when moving from the univariate to the multivariate model.

Across all dependent variables, the multivariate and univariate models yielded similar mean estimates. While this is the case for the present data set, previous simulation-based studies (with pre-specified effect sizes) have shown that the two approaches can produce different posterior means, with estimates from multivariate models closer to the true effect sizes (Bürkner, 2025; McElreath, 2020).

While the mean estimates are similar across the multivariate and univariate models, the results show a general shrinkage in the 95% CrIs when moving from the univariate to the multivariate model. For example, CrIs shrink by 10.24% for `ACpres` and 2.87% for `STRfinal` in `durWord`, and by 9.14% for `STRfinal` and 6.51% for `ACpres` in `PC1`. The only exception is `STRpenult` in `PC1`, where the CrI widens slightly. The overall shrinkage in the 95% CrIs reflects a reduction in uncertainty, indicating that jointly modelling the outcomes using the multivariate approach improves the precision of population-level estimates compared to fitting separate univariate models. Some readers might consider this magnitude of shrinkage modest, potentially questioning the value of a multivariate approach. However, even a small reduction in CrI width is meaningful, as it directly reflects reduced uncertainty in the estimates. Importantly, achieving the same reduction in CrI width with univariate models would require increasing the sample size, which entails more cost and time for data collection and annotation. For example, reducing the 95% CrI for `ACpres` on `durWord` by 10.24% through increasing sample size would require approximately 72 more observations, a 24.2% increase over the current sample size (N = 298).

Regarding our research question—whether, and to what extent, the shape and duration of the rising accent were influenced by the presence or absence of a preceding accent and by word stress location—the results of the multivariate model showed clear effects of word stress location (`STR`) on both word duration and the magnitude of the final f_0 drop (captured by `PC2`). Compared to target accents with antepenultimate stress (stress on the third-to-last syllable, `STRante`), accents with penultimate stress (stress on the second-to-last syllable, `STRpenult`) had longer duration (estimate = 0.04 seconds, 95% CrI [0.02, 0.06]) and lower `PC2` values (estimate = –7.15, 95% CrI [–9.11, –5.18]). `STRfinal` (stress on the last syllable) showed effects in the same direction on both measures, but with larger magnitudes (word duration: estimate = 0.09 seconds, 95% CrI [0.06, 0.12]; `PC2`: estimate = –17.48, 95% CrI [–20.48, –14.50]). Together, these results suggest that as word stress shifts rightward, the accent becomes longer as a result of segment elongation and shows lower f_0 scaling and a steeper final f_0 fall. In addition, the results showed that the presence of a preceding accent (`AC`) affected the phonetic realization of the target accent. When the target accent was preceded by another accent (`ACpres`), it had a shorter duration (estimated = –0.04 s, 95% CrI [–0.07, –0.01]), higher scaling (estimated = 16.30, 95% CrI [12.11, 20.55]), and a less salient final drop (estimated = –2.29, 95% CrI [–3.93, –0.63]), compared to the target accent without a preceding accent.

While our primary focus in this application is on population-level effects, the model's residual correlations also provide useful insight. They reflect how strongly the dependent variables are correlated after accounting for the predictors. In our case, the residual correlations among the three outcomes are clearly non-zero, justifying the use of

a multivariate model. If the residual correlations among outcomes had been close to zero, a multivariate model and separate univariate ones would yield similar estimates. However, considering the magnitude of residual correlations can only be assessed by running a multivariate model, it is advisable to directly adopt a multivariate approach when multiple outcomes are involved.

5.2 Application 2: correlation among population-level effects across outcomes

In addition to assessing the effect of a predictor on individual outcomes at the population level as demonstrated in Application 1, researchers are often interested in how effects interact across multiple outcomes. Two main types of interplay are typically investigated. The first concerns the relative contribution of multiple outcomes to category differentiation. This is often tested using methods such as Linear discriminant analysis (LDA) and logistic regression; see Schertz and Clare (2020) for a comprehensive review. LDA identifies the optimal linear combination of features for separating categories. The resulting standardized coefficients indicate the relative weight of each feature compared to the others. The second type of interplay concerns the correlations among outcomes in differentiating categories, that is, how outcomes co-vary in their realization of a categorical difference. Three possible scenarios can be hypothesized:

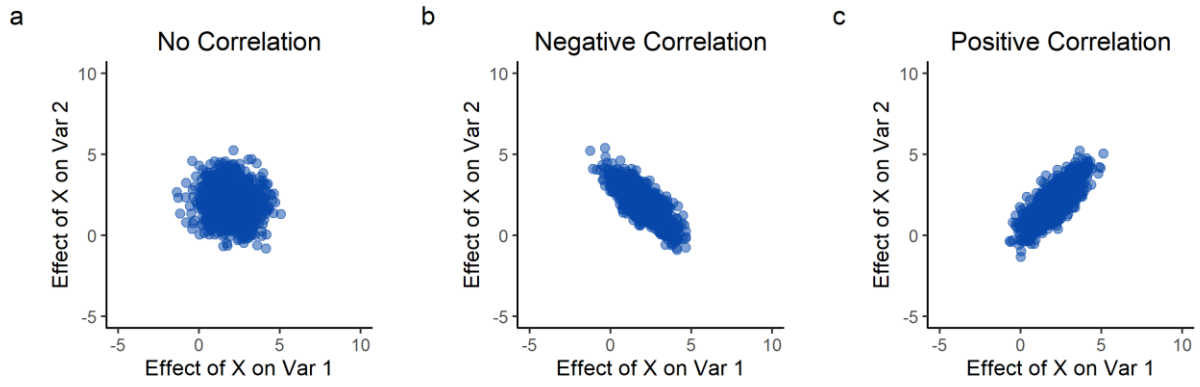

Figure 2: Illustrative examples of the relationships between the effects of a predictor variable X on two outcome variables (Var 1 and Var 2): (a) no correlation, (b) negative correlation, and (c) positive correlation.

1. **No correlation** (Fig. 2a). Changes in the two outcome variables in response to the predictor are independent of one another.
2. **Negative correlation: cue-trading** (Fig. 2b). The effect of the predictor on one outcome variable is negatively correlated with its effect on the other, indicating a trade-off relationship between the two

outcome dimensions in response to the predictor, as commonly observed in phonetics (e.g., Repp, 1982). For example, an increase in f_0 scaling in response to the predictor is accompanied by a decrease in duration in response to the same predictor.

3. **Positive correlation: cue-enhancing** (Fig. 2c). The effect of the predictor on one outcome variable is positively correlated with its effect on the other. For example, a predictor that leads to an increase in f_0 also results in longer duration across speakers.

Cue-trading and cue-enhancing can be assessed using FPCA (briefly introduced in Section 4) followed by linear regression modelling (Arvaniti et al., 2024; Asano & Gubian, 2018; Gubian et al., 2015). In this approach, FPCA acts on curves from multiple dimensions (e.g., F1 and F2 in Gubian et al., 2015; f_0 and time-warped duration curves in Arvaniti et al., 2024), transforming each pair of curves into a pair of mathematical functions, as shown in Equation (16). The curves in each dimension share the same mean curve and PC curves, while the two curves in each pair share the same PC score. This sharing of PC scores allows the PC curves across dimensions to act jointly on their respective mean curves, capturing covariation in the multidimensional input curves. By regressing the PC scores onto the predictor of interest, researchers can assess whether variation in the PC scores is associated with the predictor. Additionally, by visualizing the mean curve for each condition and dimension by inserting the model's estimated mean PC scores in Equation (16), researcher can determine whether the correlation is negative or positive. However, one limitation of this approach is that it does not provide a coefficient quantifying how strong the correlation is.

An alternative approach to examining cue-trading and cue-enhancing is to work with posteriors from a Bayesian multivariate model. However, in contexts with multiple predictor variables (as in our example), the notion of cue-trading and cue-enhancing for a single predictor can be framed in at least two different ways, depending on how the other predictors are treated. One approach is to focus on the predictor of interest while averaging over the levels of other predictors. This tests cue-trading and cue-enhancing for the predictor of interest across all levels of the other predictors. For example, Tanner et al. (2020) focused on the relationship between pVOT (the duration of "burst plus aspiration" following the release of the closure) and VDC (the presence of voicing during closure) in realizing stop voicing contrast while integrating over all levels of other factors such as speaking style and break index. Alternatively, cue-trading and cue-enhancing can be examined for the predictor of interest while holding all other predictors constant (Gelman & Hill, 2006). For the data in Tanner et al. (2020), this would mean testing the relationship between pVOT and VDC in realizing stop voicing contrast while holding the other factors, such as

speaking style, constant, for example, at their reference levels. While the choice of approach depends on the specific research question, the analyses for these two approaches differ slightly. For this first approach, Tanner et al. (2020) already presents an excellent example (but note that the `fitted_draws` function used in their paper was deprecated in `tidybayes` version 3.0; instead, one should use `add_epred_draws()` or `add_linpred_draws()`, depending on the type of prediction needed; see Smith et al. (2024) for an example. In this application, we demonstrate the second approach.

5.2.1. Research question.

The research question we aim to address concerns correlations among population-level effects of accentual context present (`ACpres`) across acoustic dimensions. Specifically, we ask whether the effect of `ACpres` on one acoustic dimension is systematically related to its impact on another at the population level (i.e., across speakers). The same question was investigated in Arvaniti et al. (2024) using FPCA, which showed a trade-off between f_0 scaling and the duration of the accented word. However, as noted above, the analysis did not quantify the strength of this correlation.

5.2.2. Analysis and results.

Essentially, addressing this question involves testing the correlations among the estimated `ACpres` effects on the three response variables, that is, the β coefficients for `ACpres` (`b_x_ACpres` with `x` being `PC1`, `PC2`, or `durWord`). Such parameter correlations can only be meaningfully tested if the coefficients were estimated by a multivariate model, in which the response variables are modelled jointly and the parameters are drawn from a joint posterior distribution. In contrast, testing correlations between posterior samples of parameter estimates drawn from separate univariate models is not statistically meaningful, as these samples come from independent posterior distributions and are therefore statistically independent. Below, we outline the steps for the analysis using samples drawn from the multivariate model.

First, we drew posterior samples of the relevant β parameters from the multivariate model using the `spread_draws()` function from the `tidybayes` package (Kay, 2024):

```
Greek_mv_m1_b_AC <- Greek_mv_m1 %>%
    spread_draws(b_PC1_ACpres, b_PC2_ACpres, b_durWord_ACpres)
```

Next, we tested the correlations among the three β coefficients. This can be done with the `cor()` function, which by default computes Pearson correlations:

```
Greek_mv_m1_b_AC_Pearson <- cor(Greek_mv_m1_b_AC[, c("b_PC1_ACpres", "b_PC2_ACpres",
    "b_durWord_ACpres")]) %>%
    round(digits = 2)
```

The resulting correlation matrix is shown below:

```
##                 b_PC1_ACpres b_PC2_ACpres b_durWord_ACpres
## b_PC1_ACpres           1.000        0.139          -0.211
## b_PC2_ACpres           0.139        1.000          -0.124
## b_durWord_ACpres      -0.211       -0.124           1.000
```

This approach, however, yields only point estimates of the correlation coefficients. To quantify uncertainty around the point estimates, we can instead fit a Bayesian multivariate model with the three coefficients as dependent variables, with only intercepts (i.e., `~ 1`) and no predictor. Since the model contains no predictors, the estimated residual correlations directly reflect the correlations among the three coefficients.

```
Greek_mv_m1_b_AC_cor <- brm(bf(mvbind(b_PC1_ACpres, b_PC2_ACpres, b_durWord_ACpres) ~
    1) + set_rescor(TRUE), data = Greek_mv_m1_b_AC, family = student, iter = 2000,
    warmup = 500, chains = 4, cores = 4, backend = "cmdstanr", threads = threading(8))
```

We listed all three coefficients within `mvbind()` on the left-hand side of the model formula. On the right-hand side, we specified only intercepts (i.e., `~ 1`). Then we wrapped the entire formula in `bf()` and use the `+` operator to append `set_rescor(TRUE)`, which instructs `brms` to estimate residual correlations among the variables. We opted for the `student` family over the `gaussian` family, following Kurz (2019), who demonstrated that the Student's *t* distribution yields more robust estimates than the Gaussian distribution in the presence of outliers. When, on the other hand, outliers are absent, the Student's *t* distribution and the Gaussian distribution produce comparable results. The model summary is provided in Appendix 2, and the `Residual Correlations` section at the bottom reports the estimated correlations among the three coefficients, summarized as the mean and 95% credible intervals representing the range of the 95% most plausible values, as follows:

```
##                                  Estimate Est.Error l-95% CI u-95% CI
## rescor(bPC1ACpres,bPC2ACpres)        0.14      0.01     0.12     0.16
## rescor(bPC1ACpres,bdurWordACpres)   -0.21      0.01    -0.24    -0.19
## rescor(bPC2ACpres,bdurWordACpres)   -0.12      0.01    -0.14    -0.10
```

As shown in the results, the mean estimates are close to the Pearson correlations. The results indicate that the effect of `ACpres` on the three response variables (`PC1`, `PC2`, and `durWord`) are correlated. Specifically, the effect on `PC1` is positively correlated with the effect on `PC2` (Estimate = 0.14, 95% CrI [0.12, 0.16]), and negatively correlated

with the effect on `durWord` (Estimate = –0.21, 95% CrI [–0.24, –0.19]). Similarly, the effect on `PC2` is negatively correlated with the effect on `durWord` (Estimate = –0.12, 95% CrI [–0.14, –0.10]).

To understand what these correlations entail, we visualized them using a paired scatterplot matrix (Figure 3) generated with the `ggpairs()` function from the `GGally` package (Schloerke et al., 2025). This matrix displays posterior sample points for each parameter pair in the lower triangular panels, along with a regression line fitted using the `lm` method and a confidence band. The upper triangular panels show the posterior summaries of the correlation coefficients, obtained using the `posterior_summary()` function from the `brms` package. Full code for generating the plot is available on OSF.

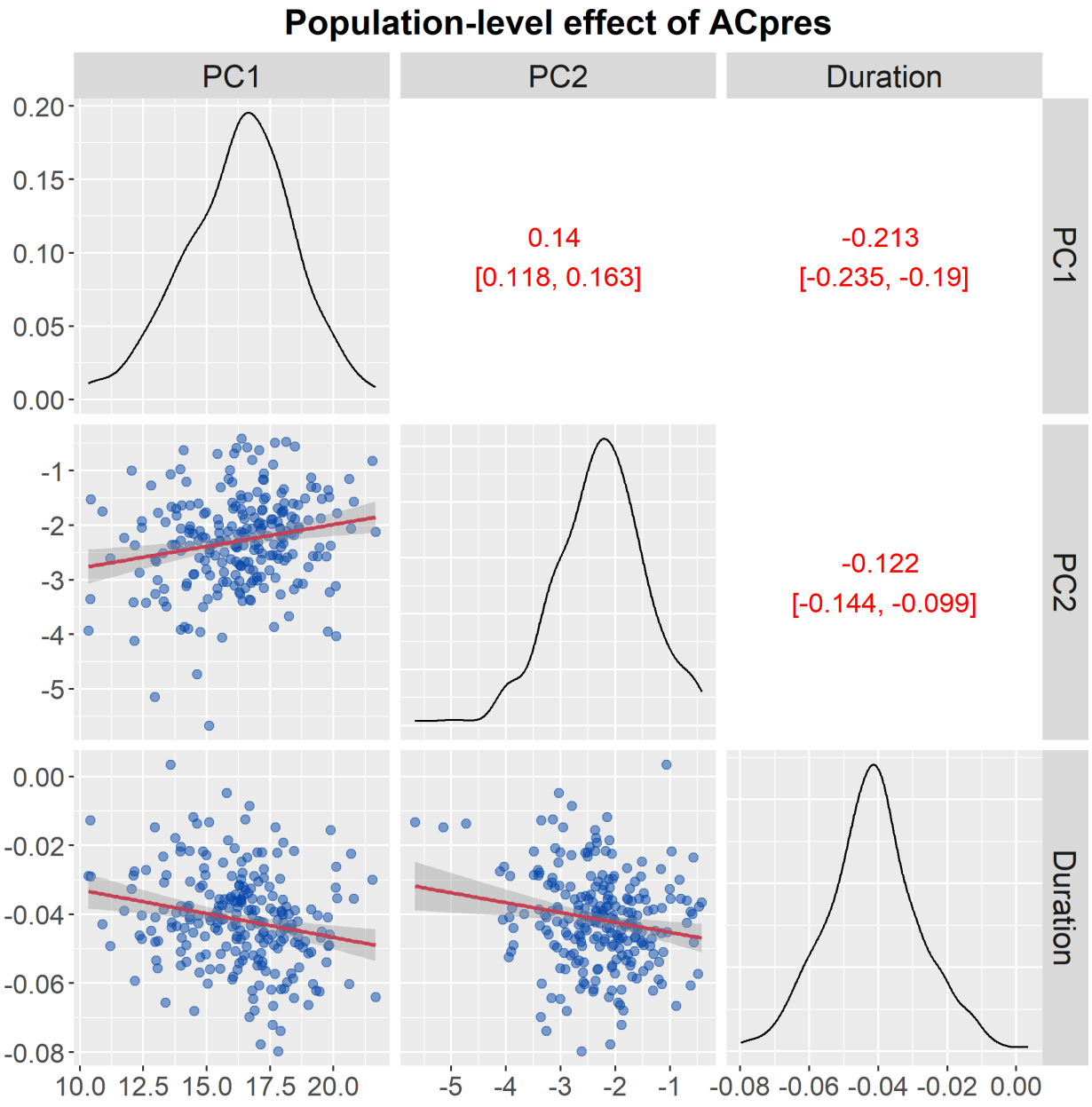

Figure 3: Paired scatterplot matrix showing the relationships between the effects of accentual context present on PC1, PC2, and durWord as predicted by the multivariate model.

Figure 3 clearly shows that the effect of `ACpres` (accentual context present) on one dependent variable is associated with its effect on another. For example, its effect on PC1 (`b_PC1_ACpres`) is negatively correlated with its effect on word duration (`b_durWord_ACpres`): larger values of `b_PC1_ACpres` correspond to more negative values of `b_durWord_ACpres`, indicating a trade-off between PC1 and word duration in realizing the accentual context contrast.

Another way to understand correlations among effects is by computing condition means derived from varying effect sizes. Specifically, one can hold the mean of the reference condition constant, manipulate one coefficient (β), compute the expected values of the other β based on their correlation, and then derive the mean of the comparison level from the resulting values. For example, to illustrate the negative correlation between the `ACpres` effect on `PC1` and on `durWord`, we first computed the means and standard deviations (SDs) of the posterior draws for the two coefficients of interest (`b_PC1_ACpres` and `b_durWord_ACpres`). We then calculated three representative values of `b_PC1_ACpres`: the mean, the mean plus two SDs (mean + 2 SDs), and the mean minus two SDs (mean – 2 SDs), and standardized them, as the correlation coefficient between two variables is equivalent to their regression coefficient only when both are on a standardized scale. Third, these standardized values of `b_PC1_ACpres` were multiplied by its correlation coefficient with `b_durWord_ACpres` (–0.21) to obtain the corresponding standardized values for `b_durWord_ACpres`, which were then back-transformed to the original scale using the mean and SD from Step 1, yielding three target values of `b_durWord_ACpres`. Finally, each target value of `b_durWord_ACpres` was added to the estimated intercept (i.e., the estimated mean of the baseline condition) to obtain the mean word duration for the AC present condition. The same procedure was used to calculate the mean `PC1` score for that condition. The resulting values were plotted in Figure 4, with word durations as bars and PC1 scores incorporated into the FPCA formula as shown in Equation (16).

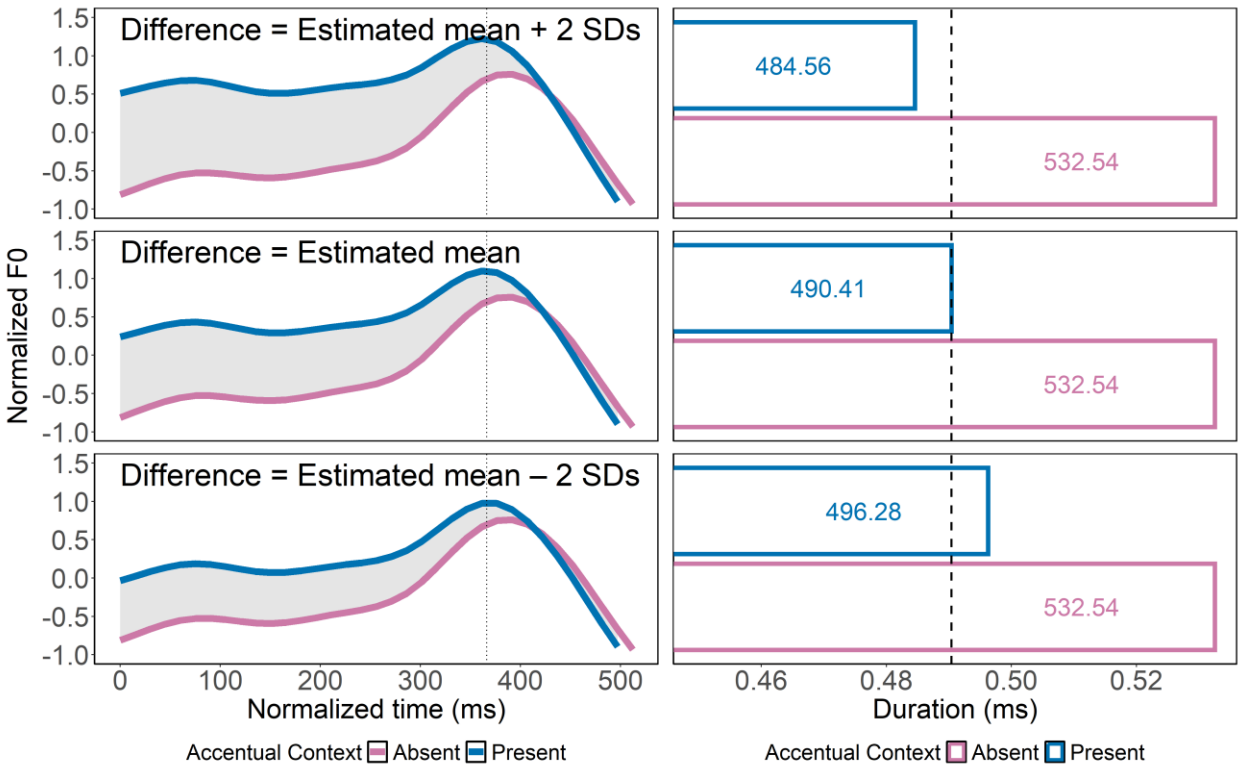

Figure 4: Illustration of the negative correlation between the effects of AccentualContext on PC1 and durWord. The left panels show normalized f_0 trajectories reconstructed using FPCA, with the dashed line indicating the onset of the accented vowel. The right panels show the corresponding word durations, with the dashed line indicating the estimated mean when accentual context is present.

The figure illustrates the negative correlation between the effects of `ACpres` (i.e., accentual context present) on f_0 curve scaling (`PC1`) and word duration. The left panels show normalized f_0 contours for the two levels of `AC`—absent (purple) and present (blue)—across three different scenarios: the estimated mean difference (middle panel) and the estimated mean plus or minus two standard deviations (top and bottom panels). The f_0 contours reveal that the presence of a preceding accent (`AC` present) is associated with higher f_0 scaling. The right panels show the corresponding effects on word duration, with blue and purple bars representing estimated durations for the present and absent conditions, respectively. The presence of a preceding accent (blue) results in shorter duration estimates than the absence of a preceding accent (purple). Together, the panels show that as f_0 scaling increases under the present condition, the duration of the word decreases, revealing a cue-trading relationship between the two dimensions under the effect of accentual context.

As previously mentioned, testing correlations among posterior samples extracted from separate univariate models is not statistically meaningful, as the samples are drawn from separate (independent) posterior distributions rather than from a joint posterior. Consequently, the correlations among those samples will always be zero, with minor deviations solely caused by sampling error. Nevertheless, we conducted this analysis, which indeed yielded near-zero correlations among the parameters of interest; details are provided in Appendix 3.

5.3 Application 3: correlation among speaker-level effects across outcomes

While cue-weighting in realizing categorical differences is well established at the population level, a growing focus in phonetic research is whether such patterns also emerge at the level of individual speakers (e.g., Chodroff, Golden, & Wilson, 2019; Chodroff & Wilson, 2017; Clayards, 2018).

Statistically, addressing this issue involves testing for correlations among speaker-specific effects across acoustic dimensions. Much of the existing literature has approached this indirectly, by extracting by-speaker estimates from separate univariate models or coefficients from separate linear discriminant analysis for each cue, and then testing for correlations post hoc (e.g., Clayards, 2018). A more direct approach is to use a Bayesian multivariate model, which jointly estimates speaker-specific effects across measures and includes explicit correlation parameters for their interrelationships. This approach enables a direct test of whether individual speakers systematically co-vary in how they respond to a given effect across different measures, and also allows for shrinkage of random effects across univariate models—an advantage that cannot be achieved when the models are estimated separately.

5.3.1. Research question.

The research question we aimed to address was whether the random effects of accentual context present and stress location correlated across different acoustic dimensions at the level of individual speakers.

5.3.2. Analysis and results.

Addressing this question requires testing correlations among by-speaker random effects of `ACpres` (accentual context present), `STRfinal` (stress on the final syllable), and `STRpenult` (stress on the penultimate syllable) across acoustic measures. These correlations, as well as correlations among these effects within each acoustic dimension, were directly estimated by the multivariate model fitted in Section 5.1.2 (`Greek_mv_m1`) and reported in the speaker-level `Multilevel Hyperparameters` section of the model summary (see Appendix 1). This section includes 66 hyperparameters, representing all possible correlations among the 12 speaker-level random effects (4 per sub-model: one intercept and three slopes for `STRfinal`, `STRpenult` and `ACpres`), including correlations between slopes, between intercepts, and between slopes and intercepts, within and across sub-models. The hyperparameter names are

formatted as `cor(parameter1, parameter2)`, where the two parameters indicate the speaker-level random effects whose correlation is being estimated. Each parameter name consists of the dependent variable, followed by an underscore, and then the type of random effect (intercept or slope for a given predictor). For example, `PC1_ACpres` refers to the random slope for the predictor `ACpres` on the response variable `PC1`.

Among the 66 correlation pairs, 48 pairs involve parameters across sub-models (i.e., acoustic dimensions). For example, `cor(PC1_ACpres,durWord_ACpres)` represents the correlation between by-speaker slopes of `ACpres` for `PC1` and `durWord`. It captures how the effect of `ACpres` co-varies across these acoustic measures at the speaker level. These cross-dimension correlations can only be estimated using multivariate models, as univariate models model each acoustic feature separately and only estimate correlations among parameters within each dimension. The remaining 18 pairs involve parameters within each sub-model. For instance, `cor(f0PC1_STRfinal,f0PC1_STRpenult)` represents the correlation between by-speaker random slopes of `STRfinal` and `STRpenult` for `f0PC1`.

Given the large number of the `cor()` hyperparameters, examining them directly in the model summary is difficult. We thus visualized them using a plot, with the 95% CrIs computed using the `mcmc_intervals_data()` function from the `bayesplot` package (Gabry & Mahr, 2025). The final plot was generated using the `ggplot2` package (Wickham, 2016), with across-dimension correlations marked in bold in the upper part of the plot and within-dimension correlations in the lower part. 95% CrIs were colored blue if they contained zero (suggesting the associated parameter could be zero) and red if they did not (suggesting the parameters are credibly non-zero). The plot is shown in Figure 5 (see OSF for the full code).

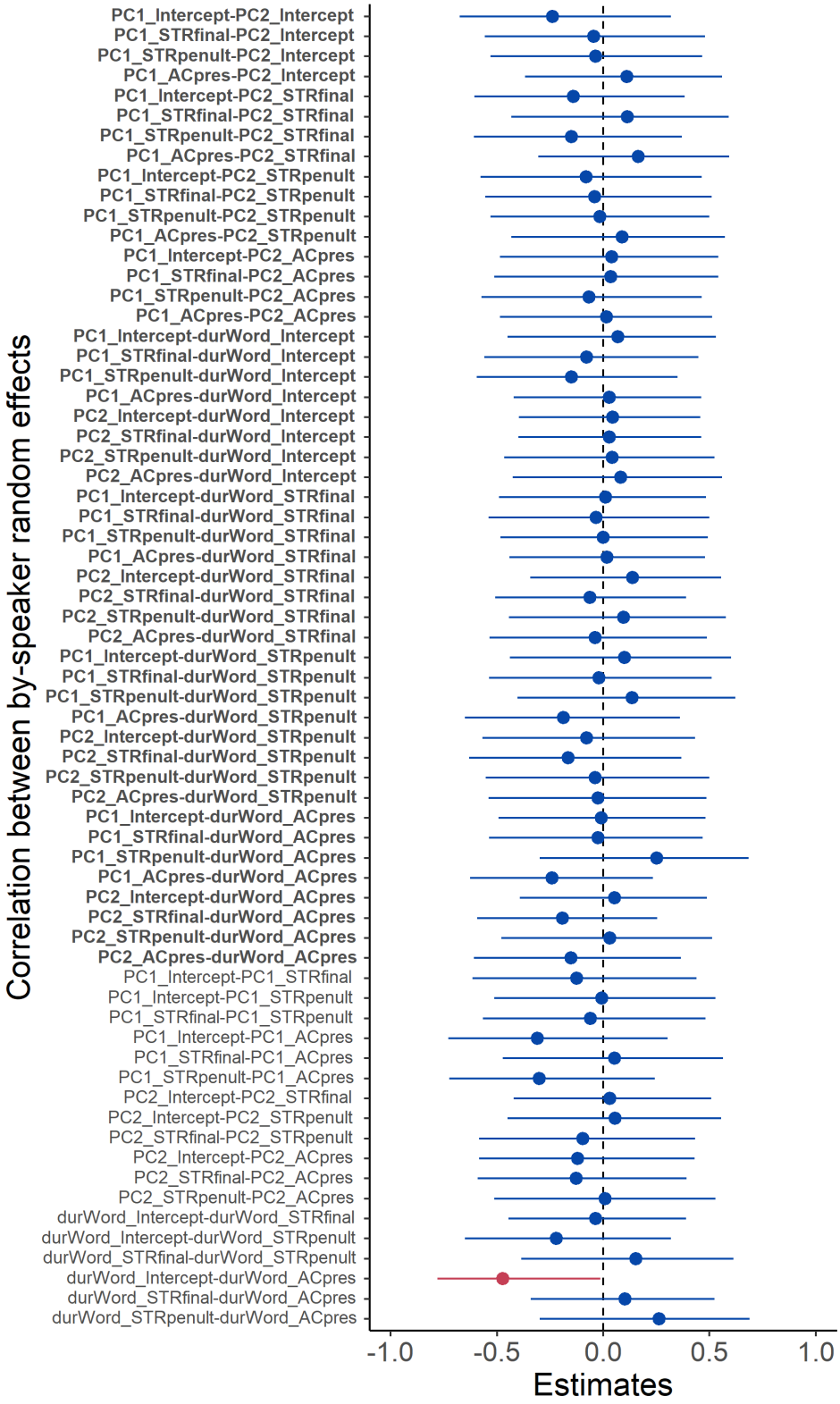

Figure 5: Pairwise correlations between speaker-level random effects.

Since our primary interest was in correlations among parameters across sub-models, we examined these first, marked in bold in the upper part of Figure 5. The figure shows that, for this particular data set, however, none of these parameters have 95% CrIs that exclude zero. This suggests that, at the speaker level, the effect of one predictor on one acoustic measure does not reliably co-vary with its effect on another measure. If we want to compute the posterior probability that a correlation is present, we can use the `hypothesis()` function in `brms`. As an example, consider the hyperparameter `cor_Speaker__PC1_ACpres__durWord_ACpres`, which represents the correlation between the speaker-specific slopes of the effect of `ACpres` on `PC1` and `durWord`. This parameter reveals whether speakers jointly modulate f_0 scaling (captured by PC1) and word duration to signal the presence of accentual context. The estimated correlation has a mean of –0.23 with a 95% CrI of [–0.63, 0.23]. To compute the posterior probability that this correlation is below zero, we use:

```
hypothesis(Greek_mv_m1, class = NULL, "cor_Speaker__PC1_ACpres__durWord_ACpres < 0")
```

This yields a posterior probability of 84% that the correlation is negative. This means that while the correlation is not credible in the strict 95% sense, the posterior distribution leans toward negative values, suggesting a modest tendency for speakers who increase f_0 scaling in the presence of accentual context to shorten word duration though the uncertainty is substantial.

Although none of the across-sub-model parameters are credibly correlated, two parameters within the same sub-model are. As shown in the lower part of Figure 5, there is a negative correlation between by-speaker intercepts for `durWord` and by-speaker slopes of `ACpres` for `durWord` at the speaker level (Estimate = –0.45, 95% CrI [–0.78, –0.01]). This suggests that speakers with longer baseline word duration tend to show weaker effects of `ACpres` on word duration, and vice versa. Since the 95% CrI does not include zero, there is a 95% probably that the negative correlation is present between the two parameters. Although such a correlation could also be estimated in a univariate model, we use it here as an example to illustrate how to visualize parameter correlations. The same procedure can be applied to across-sub-model parameter correlations.

To this end, we need to extract the relevant coefficients, which can be done in several ways. For example, the `coef()` function from the `brms` package internally sums up population-level effects and corresponding group-level effects, either as summaries or as raw values, but its output requires wrangling before plotting. Alternatively, speaker-level effects can be computed directly using the `tidybayes` package (Kay, 2024), which does not require additional wrangling. Here, we illustrate the `tidybayes` approach.

We first drew posterior samples of the relevant model parameters, namely, population-level intercept of `durWord` (`b_durWord_Intercept`), the population-level slope of `ACpres` on `durWord` (`b_durWord_ACpres`), and by-speaker random effects (i.e., deviations from the population-level effects for each speaker).

```
Greek_mv_m1_draws <- Greek_mv_m1 %>%
    spread_draws(b_durWord_Intercept, b_durWord_ACpres, r_Speaker__durWord[Speaker,
        term])
```

The structure of the resulting object is shown below:

```
## # A tibble: 4 x 8
## # Groups:   Speaker, term [4]
##   .chain .iteration .draw b_durWord_Intercept b_durWord_ACpres Speaker term      r_Speaker__durWord
##    <dbl>      <dbl> <dbl>               <dbl>            <dbl> <chr>   <chr>                  <dbl>
## 1      1          1     1                0.49            -0.05 F1      ACpres                  0.05
## 2      1          1     1                0.49            -0.05 F1      Intercept              -0.05
## 3      1          1     1                0.49            -0.05 F1      STRfinal               -0.01
## 4      1          1     1                0.49            -0.05 F1      STRpenult               0.02
```

With `r_Speaker__durWord[Speaker,term]`, we obtained posterior samples of the random effects on the response variable `durWord` for each speaker (shown in the `Speaker` column) and for each effect term (shown in the `term` column). The effect terms included not only `Intercept` and `ACpres`, but also other effects (i.e., `STRfinal` and `STRpenult`). Since we were only interested in `Intercept` and `ACpres`, we filtered the `term` column to retain only rows corresponding to the two effects. We then calculated by-speaker intercepts for `durWord` (`durWord_Intercept`) by summing the population-level intercept (`b_durWord_Intercept`) and the speaker-specific deviation (`r_Speaker__durWord`); the same operation was performed to obtain `coef_durWord_ACpres` with corresponding parameters, representing the speaker-specific effects of `ACpres` for `durWord`.

```
Greek_mv_m1_durWord_Intercept <- Greek_mv_m1_draws %>%
    filter(term == "Intercept") %>%
    ungroup() %>%
    mutate(durWord_Intercept = b_durWord_Intercept + r_Speaker__durWord, id = paste(.chain,
        .iteration, .draw, Speaker, sep = "_")) %>%
    dplyr::select(Speaker, id, durWord_Intercept)

Greek_mv_m1_coef_durWord_ACpres <- Greek_mv_m1_draws %>%
    filter(term == "ACpres") %>%
```

```
    ungroup() %>%
    mutate(coef_durWord_ACpres = b_durWord_ACpres + r_Speaker__durWord, id = paste(.chain,
        .iteration, .draw, Speaker, sep = "_")) %>%
    dplyr::select(Speaker, id, coef_durWord_ACpres)
```

We then merged the two data frames by `id` and plotted `durWord_Intercept` against `coef_durWord_ACpres` as a scatter plot using the `ggplot2` package (Figure 6). The points represent posterior samples of speaker-specific intercepts (x-axis) and speaker-specific effects for `ACpres` on `durWord` (y-axis), color-coded by speaker. Ellipses and lines reflect the 95% confidence regions and local regressions for individual speakers, respectively. The bold black line represents the overall trend across speakers. Again, the full code is available on OSF.

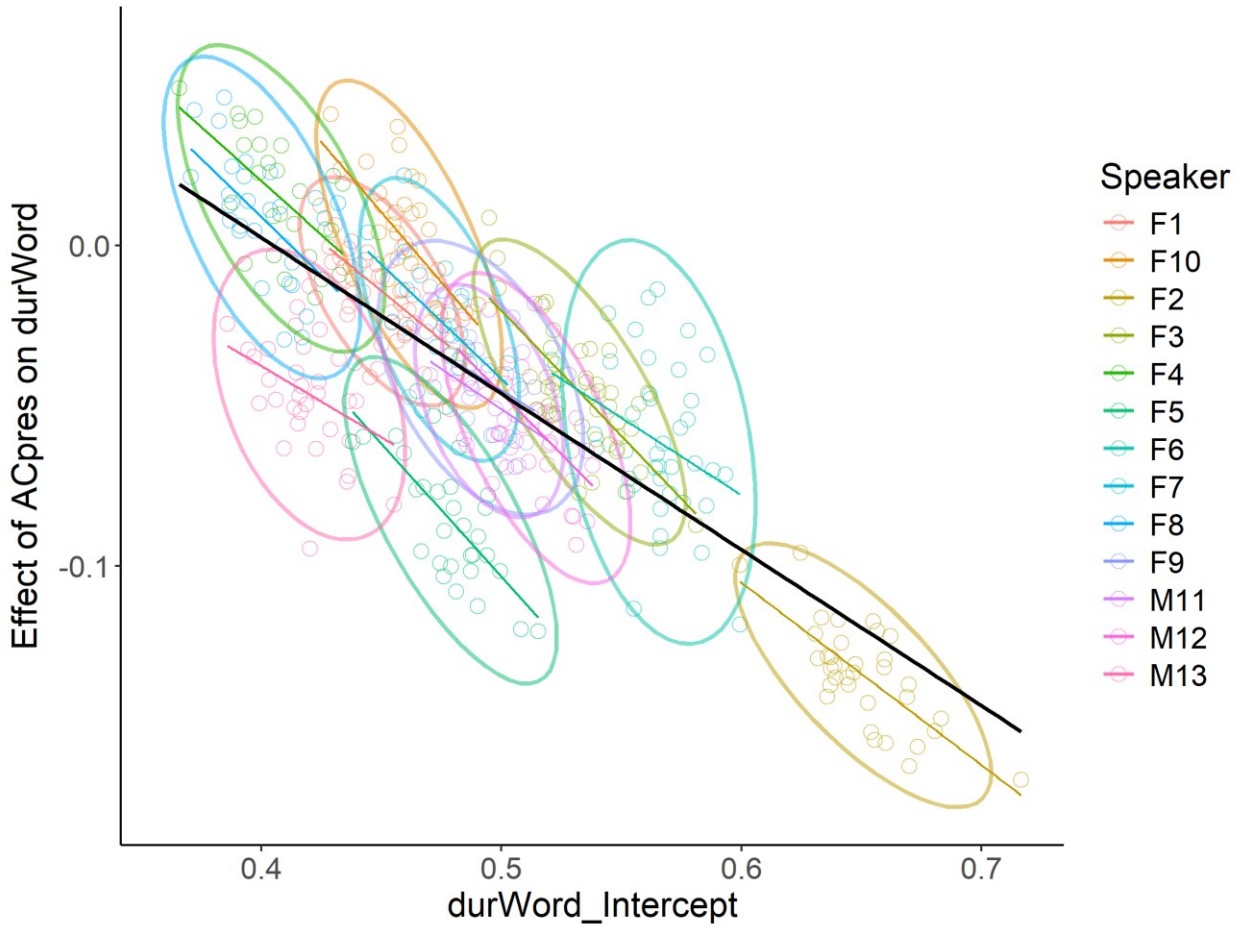

Figure 6: Speaker-specific intercepts for `durWord` and speaker-specific effects of `ACpres` on `durWord`.

The plot shows a clear negative relationship between speakers' baseline word duration and the effect of `ACpres` on word duration: as speakers' baseline word duration increases, the effect of `ACpres` becomes more

negative—that is, speakers with longer baseline word duration reduce duration more when accentual context is present.

5. Discussion and conclusions

The aim of this tutorial was to demonstrate how Bayesian multivariate linear mixed effects models, fitted using the `brms` package, can be applied to address common questions in phonetic research. Such models are particularly useful when multiple outcome variables are involved, as they allow estimation of residual correlations among outcome variables, potentially reducing estimation uncertainty compared to separate univariate models. Additionally, they enable direct examination of correlations among effects across outcomes. To demonstrate these, we presented three applications: (1) assessing the effect of a predictor on multiple acoustic features at the population level, (2) examining correlations among population-level effects across acoustic features, and (3) examining correlations among speaker-level effects across features.

The results of Application 1 showed that the multivariate approach reduced uncertainty in population-level effect estimates compared to the univariate models. Applications 2 and 3 demonstrated that the multivariate approach provides a convenient way to examine correlations among model parameters across outcomes, at both the population and speaker levels. Overall, this tutorial illustrates that the `brms` package, along with many supporting packages, makes it easy to fit Bayesian multivariate models and draw inferences from them. Moreover, although such models can be computationally demanding, their runtime can be substantially reduced through the use of cloud computing services at an affordable cost.

In addition, Bayesian multivariate linear mixed effects models address concerns related to multiple comparisons across conditions. In the frequentist framework, multiple comparisons need to be corrected because each additional test introduces an independent chance of falsely rejecting a true null hypothesis (Type 1 error). In contrast, Bayesian inference does not reply on null hypothesis significance testing and therefore does not incur Type 1 error inflation in the same way. Instead, Bayesian approaches mitigate concerns about multiple comparisons by producing more reliable parameter estimates, for example through the use of multilevel model structures or by incorporating informative priors when available. These mechanisms induce shrinkage in posterior distributions, as demonstrated in Application 1, rendering comparisons more conservative and more likely to be valid (Gelman, Hill, & Yajima, 2012).

While we demonstrated three applications of Bayesian multivariate linear models, these models can also be used to address other questions. For example, Tanner et al. (2020) examined speaker variability within a single

acoustic cue in realizing stop voicing contrast, and correlations among cues within each voicing category. Hu and Arvaniti (2024) performed a clustering analysis on the characteristics of 95% CrIs of pitch accent effects across acoustic features, aiming to reveal individual variability in cue selection for realizing pitch accent contrasts. Smith et al. (2024) used simulated distributions to compute a measure of vowel overlap and found a substantial improvement compared to the measure computed from empirical distributions.

Due to space limitations, we did not include several common practices in Bayesian model fitting, such as model fit checking and model comparison, because these procedures for multivariate models are the same as for univariate models and have already been sufficiently described elsewhere (e.g., Bürkner, 2025, for model fit checking of multivariate models and model comparison based on information criteria).

This tutorial focuses on multivariate linear models, in which all response variables are continuous and follow a Gaussian distribution. Multivariate models can also accommodate responses with different distributions, for example, one response being a count variable with a Poisson distribution while the other a positive continuous variable with an exponential distribution. To specify such models in `brms`, instead of specifying a global family as shown in our example, families can be assigned to individual sub-models (for example code, see Bürkner, 2025). The only constraint is that for families other than `gaussian` or `student`, `brms` does not yet directly support modelling residual correlation (specified via the `set_rescor()` argument). This is because, neither the Poisson distribution ($Poisson(\lambda)$) nor the exponential distribution ($Exponential(\lambda)$) has the residual parameter σ needed to estimate residual correlations, as in the Gaussian distribution ($Normal(\mu, \sigma)$) (Bürkner, 2025; McElreath, 2020). To obtain the residual parameters needed for estimating residual correlations, one approach is to add residuals to the model via observation-level varying coefficients, that is, by including one coefficient per observation. A detailed discussion of this approach is beyond the scope of this tutorial, but see Bürkner (2025) for a comprehensive guide.

To sum up, we hope this tutorial has demonstrated the potential of Bayesian multivariate linear models and resolved some of the practical hurdles in fitting them. We also hope it serves as the starting point for researchers to further explore the possibilities these models offer to gain deeper insights from their data.

Acknowledgments

This work was financially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. ERC-ADG-835263 to A.A.).

Data Availability Statement

The data and scripts are available at the OSF repository:

https://osf.io/2kcnu/overview?view_only=d16293b898f54f228f1dbc75877bbbfe.

References

Arvaniti, A., Katsika, A., & Hu, N. (2024). Variability, overlap, and cue trading in intonation. *Language*, *100*, 265–307.

Asano, Y., & Gubian, M. (2018). "Excuse meeee!!": (Mis)coordination of lexical and paralinguistic prosody in L2 hyperarticulation. *Speech Communication*, *99*, 183–200. https://doi.org/10.1016/j.specom.2017.12.011

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*, 255–278.

Brown, P. J., Fearn, T., & Vannucci, M. (1999). The choice of variables in multivariate regression: A non-conjugate Bayesian decision theory approach. *Biometrika*, *86*(3), 635–648. https://doi.org/10.1093/biomet/86.3.635

Bürkner, P. C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *R Journal*, *10*, 395–411. https://doi.org/10.32614/rj-2018-017

Bürkner, P. C. (2025). *The brms book: Applied Bayesian regression modelling using R and stan (early draft)*. Retrieved from https://paulbuerkner.com/software/brms-book

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*, 1–32. https://doi.org/10.18637/jss.v076.i01

Chodroff, E., Golden, A., & Wilson, C. (2019). Covariation of stop voice onset time across languages: Evidence for a universal constraint on phonetic realization. *The Journal of the Acoustical Society of America*, *145*(1), EL109–EL115. https://doi.org/10.1121/1.5088035

Chodroff, E., & Wilson, C. (2017). Structure in talker-specific phonetic realization: Covariation of stop consonant VOT in American English. *Journal of Phonetics*, *61*, 30–47. https://doi.org/https://doi.org/10.1016/j.wocn.2017.01.001

Clayards, M. (2018). Individual talker and token covariation in the production of multiple cues to stop voicing. *Phonetica*, *75*, 1–23. https://doi.org/https://doi.org/10.1159/000448809

Cole, J. (2015). Prosody in context: A review. *Language, Cognition and Neuroscience*, *30*, 1–31. https://doi.org/10.1080/23273798.2014.963130

Depaoli, S., Clifton, J. P., & Cobb, P. R. (2016). Just another gibbs sampler (JAGS): Flexible software for MCMC implementation. *Journal of Educational and Behavioral Statistics*, *41*(6), 628–649. https://doi.org/https://doi.org/10.3102/1076998616664876

Foster, S., & Cole, J. (2024). Trading relations in segmental cues to prosodic prominence. *Speech Prosody 2024*, 577–581. https://doi.org/10.21437/SpeechProsody.2024-117

790 Gabry, J., Češnovar, R., Johnson, A., & Bronder, S. (2025). *Cmdstanr: R interface to 'CmdStan'*. Retrieved from https://mc-
791 stan.org/cmdstanr/
792 Gabry, J., & Mahr, T. (2025). *Bayesplot: Plotting for Bayesian models*. Retrieved from https://mc-stan.org/bayesplot/
793 Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.
794 Gelman, A., Hill, J., & Yajima, M. (2012). Why we (usually) don't have to worry about multiple comparisons. *Journal of*
795 *Research on Educational Effectiveness*, *5*(2), 189–211. https://doi.org/10.1080/19345747.2011.618213
796 Gubian, M., Torreira, F., & Boves, L. (2015). Using functional data analysis for investigating multidimensional dynamic phonetic
797 contrasts. *Journal of Phonetics*, *49*, 16–40. https://doi.org/10.1016/j.wocn.2014.10.001
798 Hadfield, J. D. (2010). MCMC methods for multi-response generalized linear mixed models: The MCMCglmm R package.
799 *Journal of Statistical Software*, *33*(2), 1–22. Retrieved from https://www.jstatsoft.org/v33/i02/
800 Honda, C. T., Clayards, M., & Baum, S. R. (2024). Exploring individual differences in native phonetic perception and their link to
801 nonnative phonetic perception. *Journal of Experimental Psychology: Human Perception and Performance*, *50*(4), 370–
802 394. https://doi.org/https://doi.org/10.1037/xhp0001191
803 Hu, N., & Arvaniti, A. (2024). Individual variability in the use of tonal and non-tonal cues in intonation. *JASA Express Letters*, *4*,
804 095203. https://doi.org/https://doi.org/10.1121/10.0028613
805 Kay, M. (2024). *tidybayes: Tidy data and geoms for Bayesian models*. https://doi.org/10.5281/zenodo.1308151
806 Kim, J., Hu, N., & Arvaniti, A. (in preparetion)...
807 Kruschke, J. K. (2015). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan* (Second Edi). Academic Press /
808 Elsevier. https://doi.org/10.1016/B978-0-12-405888-0.09999-2
809 Kurz, A. S. (2019). *Bayesian robust correlations with brms (and why you should love student's t)*. Retrieved from
810 https://solomonkurz.netlify.app/blog/2019-02-10-bayesian-robust-correlations-with-brms-and-why-you-should-love-
811 student-s-t/
812 Li, X., Ghosh, J., & Villarini, G. (2023). A comparison of Bayesian multivariate versus univariate normal regression models for
813 prediction. *The American Statistician*, *77*(3), 304–312. https://doi.org/10.1080/00031305.2022.2087735
814 Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: Concepts,
815 structure, and extensibility. *Statistics and Computing*, *10*(4), 325–337.
816 https://doi.org/http://dx.doi.org/10.1023/A:1008929526011
817 McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and STAN (2nd ed.)*. Chapman; Hall/CRC
818 Press. https://doi.org/https://doi.org/10.1201/9780429029608
819 Nalborczyk, L., Batailler, C., Lœvenbruck, H., Vilain, A., & Bürkner, P. C. (2019). An introduction to Bayesian multilevel
820 models using brms: A case study of gender effects on vowel variability in standard Indonesian. *Journal of Speech,*
821 *Language, and Hearing Research : JSLHR*, *62*, 1225–1242. https://doi.org/10.1044/2018_JSLHR-S-18-0006

Pinker, S. (2021). *Rationality: What it is, why it seems scarce, why it matters*. New York, NY: Penguin.

Repp, B. H. (1982). Phonetic trading relations and context effects: New experimental evidence for a speech mode of perception. *Psychological Bulletin*, *92*, 81–110. https://doi.org/https://doi.org/10.1037/0033-2909.92.1.81

Roettger, T. B., & Grice, M. (2019). The tune drives the text - competing information channels of speech shape phonological systems. *Language Dynamics and Change*, *9*(2), 265–298. https://doi.org/10.1163/22105832-00902006

Schertz, J., & Clare, E. J. (2020). Phonetic cue weighting in perception and production. *WIREs Cognitive Science*, *11*(2), e1521. https://doi.org/https://doi.org/10.1002/wcs.1521

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., … Crowley, J. (2025). *GGally: Extension to 'ggplot2'*. Retrieved from https://ggobi.github.io/ggally/

Schröder, J. (2019). *How to run RStudio on AWS in under 3 minutes for free*. Retrieved from https://towardsdatascience.com/how-to-run-rstudio-on-aws-in-under-3-minutes-for-free-65f8d0b6ccda/

Smith, I., Sonderegger, M., & Spade Consortium, T. (2024). Modelled Multivariate Overlap: A method for measuring vowel merger. *Interspeech 2024*, 457–461. https://doi.org/10.21437/Interspeech.2024-2260

Sonderegger, M., & Sóskuthy, M. (2025). Advancements of phonetics in the 21st century: Quantitative data analysis. *Journal of Phonetics*, *111*, 101415. https://doi.org/https://doi.org/10.1016/j.wocn.2025.101415

Stan Development Team. (2025). *CmdStan user's guide*. Retrieved from https://mc-stan.org/docs/cmdstan-guide/

Tanner, J., Sonderegger, M., & Stuart-Smith, J. (2020). Structured speaker variability in Japanese stops: Relationships within versus across cues to stop voicing. *The Journal of the Acoustical Society of America*, *148*, 793–804. https://doi.org/https://doi.org/10.1121/10.0001734

Vasishth, S., Nicenboim, B., Beckman, M. E., Li, F., & Kong, E. J. (2018). Bayesian data analysis in the phonetic sciences: A tutorial introduction. *Journal of Phonetics*, *71*, 147–161. https://doi.org/10.1016/j.wocn.2018.07.008

Weber, S., & Bürkner, P. (2025). *Running brms models with within-chain parallelization*. Retrieved from https://cran.r-project.org/web/packages/brms/vignettes/brms_threading.html

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2025). *Dplyr: A grammar of data manipulation*. Retrieved from https://dplyr.tidyverse.org

Wu, D., Goldfeld, K. S., Petkova, E., & Park, H. G. (2024). A Bayesian multivariate hierarchical model for developing a treatment benefit index using mixed types of outcomes. *BMC Medical Research Methodology*, *24*, 218. https://doi.org/https://doi.org/10.1186/s12874-024-02333-z

Appendix 1: Summary of the multivariate model used in Application 1

The summary of the multivariate model, `Greek_mv_m1`, is shown below:

```
##  Family: MV(gaussian, gaussian, gaussian)
##   Links: mu = identity; sigma = identity
##          mu = identity; sigma = identity
##          mu = identity; sigma = identity
## Formula: PC1 ~ STR + AC + (STR + AC | p | Speaker)
##          PC2 ~ STR + AC + (STR + AC | p | Speaker)
##          durWord ~ STR + AC + (STR + AC | p | Speaker)
##    Data: Greek_data (Number of observations: 298)
##   Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup draws = 8000
##
## Multilevel Hyperparameters:
## ~Speaker (Number of levels: 13)
##                                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(PC1_Intercept)                        2.68      1.52     0.19     5.90 1.00     2271     3086
## sd(PC1_STRfinal)                         2.85      2.00     0.11     7.44 1.00     3161     4131
## sd(PC1_STRpenult)                        4.09      2.01     0.40     8.32 1.00     2450     1857
## sd(PC1_ACpres)                           5.46      2.01     1.73     9.87 1.00     2809     2757
## sd(PC2_Intercept)                        3.30      1.07     1.61     5.79 1.00     4647     5230
## sd(PC2_STRfinal)                         4.42      1.39     2.23     7.61 1.00     5092     6030
## sd(PC2_STRpenult)                        1.82      1.20     0.09     4.53 1.00     3451     3783
## sd(PC2_ACpres)                           1.66      1.02     0.11     3.90 1.00     3424     3732
## sd(durWord_Intercept)                    0.08      0.02     0.05     0.12 1.00     4459     5884
## sd(durWord_STRfinal)                     0.05      0.01     0.02     0.08 1.00     5964     6178
## sd(durWord_STRpenult)                    0.02      0.01     0.00     0.04 1.00     3967     3254
## sd(durWord_ACpres)                       0.04      0.01     0.03     0.07 1.00     6045     5690
## cor(PC1_Intercept,PC1_STRfinal)         -0.11      0.28    -0.62     0.44 1.00     8406     6257
## cor(PC1_Intercept,PC1_STRpenult)        -0.00      0.27    -0.51     0.53 1.00     7862     6493
## cor(PC1_STRfinal,PC1_STRpenult)         -0.06      0.27    -0.57     0.48 1.00     6656     6413
## cor(PC1_Intercept,PC1_ACpres)           -0.28      0.27    -0.73     0.30 1.00     3351     4846
## cor(PC1_STRfinal,PC1_ACpres)             0.05      0.27    -0.47     0.56 1.00     5115     6407
## cor(PC1_STRpenult,PC1_ACpres)           -0.28      0.25    -0.72     0.24 1.00     4887     6537
## cor(PC1_Intercept,PC2_Intercept)        -0.22      0.26    -0.67     0.32 1.00     3696     4782
```

```
## cor(PC1_STRfinal,PC2_Intercept)            -0.05      0.27    -0.56     0.48 1.00     4708     5709
## cor(PC1_STRpenult,PC2_Intercept)           -0.04      0.26    -0.53     0.47 1.00     5394     5614
## cor(PC1_ACpres,PC2_Intercept)               0.11      0.24    -0.37     0.56 1.00     6836     6825
## cor(PC1_Intercept,PC2_STRfinal)            -0.13      0.26    -0.61     0.38 1.00     3712     4760
## cor(PC1_STRfinal,PC2_STRfinal)              0.10      0.27    -0.43     0.59 1.00     4494     6176
## cor(PC1_STRpenult,PC2_STRfinal)            -0.14      0.25    -0.61     0.37 1.00     4944     5875
## cor(PC1_ACpres,PC2_STRfinal)                0.16      0.23    -0.30     0.59 1.00     7209     7370
## cor(PC2_Intercept,PC2_STRfinal)             0.03      0.24    -0.42     0.51 1.00     7187     6863
## cor(PC1_Intercept,PC2_STRpenult)           -0.07      0.27    -0.58     0.46 1.00     8772     6403
## cor(PC1_STRfinal,PC2_STRpenult)            -0.03      0.28    -0.55     0.51 1.00     7792     6114
## cor(PC1_STRpenult,PC2_STRpenult)           -0.02      0.27    -0.53     0.50 1.00    10235     6578
## cor(PC1_ACpres,PC2_STRpenult)               0.09      0.26    -0.43     0.57 1.00     9500     6872
## cor(PC2_Intercept,PC2_STRpenult)            0.05      0.26    -0.45     0.55 1.00    10033     6205
## cor(PC2_STRfinal,PC2_STRpenult)            -0.09      0.26    -0.58     0.43 1.00     8379     6507
## cor(PC1_Intercept,PC2_ACpres)               0.04      0.27    -0.49     0.54 1.00    10448     6644
## cor(PC1_STRfinal,PC2_ACpres)                0.03      0.28    -0.51     0.54 1.00     9820     6369
## cor(PC1_STRpenult,PC2_ACpres)              -0.06      0.27    -0.57     0.46 1.00     8422     6244
## cor(PC1_ACpres,PC2_ACpres)                  0.01      0.26    -0.49     0.51 1.00     9650     6738
## cor(PC2_Intercept,PC2_ACpres)              -0.11      0.27    -0.58     0.43 1.00     9104     7052
## cor(PC2_STRfinal,PC2_ACpres)               -0.12      0.26    -0.59     0.39 1.00     7872     7099
## cor(PC2_STRpenult,PC2_ACpres)               0.01      0.27    -0.51     0.53 1.00     6687     6914
## cor(PC1_Intercept,durWord_Intercept)        0.07      0.25    -0.45     0.53 1.00     3597     4856
## cor(PC1_STRfinal,durWord_Intercept)        -0.07      0.26    -0.56     0.45 1.00     3938     5456
## cor(PC1_STRpenult,durWord_Intercept)       -0.14      0.25    -0.59     0.35 1.00     4060     4826
## cor(PC1_ACpres,durWord_Intercept)           0.03      0.23    -0.42     0.46 1.00     6113     6376
## cor(PC2_Intercept,durWord_Intercept)        0.04      0.22    -0.40     0.46 1.00     5710     6270
## cor(PC2_STRfinal,durWord_Intercept)         0.03      0.22    -0.40     0.46 1.00     6381     6519
## cor(PC2_STRpenult,durWord_Intercept)        0.04      0.26    -0.47     0.52 1.00     4883     6380
## cor(PC2_ACpres,durWord_Intercept)           0.08      0.25    -0.43     0.56 1.00     5485     6454
## cor(PC1_Intercept,durWord_STRfinal)         0.01      0.25    -0.49     0.48 1.00     5874     6655
## cor(PC1_STRfinal,durWord_STRfinal)         -0.03      0.27    -0.54     0.50 1.00     4774     5794
## cor(PC1_STRpenult,durWord_STRfinal)        -0.00      0.25    -0.48     0.49 1.00     5494     6056
## cor(PC1_ACpres,durWord_STRfinal)            0.02      0.24    -0.44     0.48 1.00     7299     7098
## cor(PC2_Intercept,durWord_STRfinal)         0.13      0.23    -0.34     0.56 1.00     7640     7115
## cor(PC2_STRfinal,durWord_STRfinal)         -0.06      0.23    -0.51     0.39 1.00     6797     6885
```

```
## cor(PC2_STRpenult,durWord_STRfinal)          0.09      0.26    -0.44     0.58 1.00     4932     6116
## cor(PC2_ACpres,durWord_STRfinal)            -0.03      0.26    -0.53     0.49 1.00     6208     7164
## cor(durWord_Intercept,durWord_STRfinal)     -0.03      0.22    -0.45     0.39 1.00     7789     7790
## cor(PC1_Intercept,durWord_STRpenult)         0.10      0.27    -0.44     0.60 1.00     7165     6142
## cor(PC1_STRfinal,durWord_STRpenult)         -0.02      0.27    -0.54     0.51 1.00     7135     5990
## cor(PC1_STRpenult,durWord_STRpenult)         0.13      0.27    -0.40     0.62 1.00     8666     5994
## cor(PC1_ACpres,durWord_STRpenult)           -0.18      0.26    -0.65     0.36 1.00     9182     6975
## cor(PC2_Intercept,durWord_STRpenult)        -0.08      0.26    -0.57     0.43 1.00     9691     6543
## cor(PC2_STRfinal,durWord_STRpenult)         -0.15      0.26    -0.63     0.37 1.00     8656     6596
## cor(PC2_STRpenult,durWord_STRpenult)        -0.04      0.27    -0.55     0.50 1.00     5898     6894
## cor(PC2_ACpres,durWord_STRpenult)           -0.03      0.27    -0.54     0.49 1.00     6242     7479
## cor(durWord_Intercept,durWord_STRpenult)    -0.21      0.25    -0.65     0.32 1.00     7103     6718
## cor(durWord_STRfinal,durWord_STRpenult)      0.14      0.26    -0.38     0.61 1.00     7273     6264
## cor(PC1_Intercept,durWord_ACpres)           -0.01      0.25    -0.49     0.48 1.00     4232     5588
## cor(PC1_STRfinal,durWord_ACpres)            -0.03      0.26    -0.54     0.47 1.00     3971     5204
## cor(PC1_STRpenult,durWord_ACpres)            0.24      0.25    -0.30     0.68 1.00     3819     4580
## cor(PC1_ACpres,durWord_ACpres)              -0.23      0.22    -0.63     0.23 1.00     6277     6547
## cor(PC2_Intercept,durWord_ACpres)            0.05      0.23    -0.39     0.49 1.00     7726     6783
## cor(PC2_STRfinal,durWord_ACpres)            -0.19      0.22    -0.59     0.25 1.00     7251     6900
## cor(PC2_STRpenult,durWord_ACpres)            0.03      0.26    -0.48     0.51 1.00     5914     6670
## cor(PC2_ACpres,durWord_ACpres)              -0.14      0.25    -0.61     0.36 1.00     5764     5957
## cor(durWord_Intercept,durWord_ACpres)       -0.45      0.20    -0.78    -0.01 1.00     7126     6907
## cor(durWord_STRfinal,durWord_ACpres)         0.10      0.22    -0.34     0.52 1.00     7666     7010
## cor(durWord_STRpenult,durWord_ACpres)        0.24      0.26    -0.30     0.69 1.00     5063     6998
##
## Regression Coefficients:
##                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## PC1_Intercept        -8.15      1.55   -11.15    -5.03 1.00     9676     6652
## PC2_Intercept         9.39      1.15     7.14    11.68 1.00     6054     6397
## durWord_Intercept     0.49      0.02     0.44     0.53 1.00     5572     5614
## PC1_STRfinal          1.30      1.87    -2.32     4.96 1.00     9652     6127
## PC1_STRpenult        -0.28      2.09    -4.50     3.83 1.00     7525     6127
## PC1_ACpres           16.30      2.11    12.11    20.55 1.00     5096     5682
## PC2_STRfinal        -17.48      1.51   -20.48   -14.50 1.00     5360     5109
## PC2_STRpenult        -7.15      1.00    -9.11    -5.18 1.00     8706     6081
```

```
## PC2_ACpres           -2.29      0.83    -3.93    -0.63 1.00     8936     6357
## durWord_STRfinal      0.09      0.01     0.06     0.12 1.00     6135     6078
## durWord_STRpenult     0.04      0.01     0.02     0.06 1.00     9620     6343
## durWord_ACpres       -0.04      0.01    -0.07    -0.01 1.00     5871     6122
##
## Further Distributional Parameters:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma_PC1        11.40      0.51    10.46    12.43 1.00     7994     5905
## sigma_PC2         5.50      0.25     5.03     6.02 1.00    12306     5501
## sigma_durWord     0.05      0.00     0.04     0.05 1.00    11240     6250
##
## Residual Correlations:
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## rescor(PC1,PC2)         0.26      0.06     0.15     0.38 1.00    15122     6261
## rescor(PC1,durWord)    -0.33      0.06    -0.43    -0.21 1.00    12676     5817
## rescor(PC2,durWord)    -0.15      0.06    -0.27    -0.02 1.00    14379     5677
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Appendix 2: Summary of the multivariate model used in Application 2

```
##  Family: MV(student, student, student)
##   Links: mu = identity; sigma = identity; nu = identity
##          mu = identity; sigma = identity; nu = identity
##          mu = identity; sigma = identity; nu = identity
## Formula: b_PC1_ACpres ~ 1
##          b_PC2_ACpres ~ 1
##          b_durWord_ACpres ~ 1
##    Data: Greek_mv_m1_b_AC (Number of observations: 8000)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Regression Coefficients:
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## bPC1ACpres_Intercept        16.30      0.02    16.26    16.35 1.00     6237     3991
```

```
## bPC2ACpres_Intercept        -2.29      0.01    -2.30    -2.27 1.00     5197     3620
## bdurWordACpres_Intercept    -0.04      0.00    -0.04    -0.04 1.00     5945     4050
##
## Further Distributional Parameters:
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma_bPC1ACpres        1.98      0.02     1.95     2.02 1.00     5618     4409
## sigma_bPC2ACpres        0.77      0.01     0.76     0.79 1.00     4147     3523
## sigma_bdurWordACpres    0.01      0.00     0.01     0.01 1.00     5146     4553
## nu                     16.11      1.29    13.87    18.90 1.00     4276     3906
## nu_bPC1ACpres           1.00      0.00     1.00     1.00   NA       NA       NA
## nu_bPC2ACpres           1.00      0.00     1.00     1.00   NA       NA       NA
## nu_bdurWordACpres       1.00      0.00     1.00     1.00   NA       NA       NA
##
## Residual Correlations:
##                                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## rescor(bPC1ACpres,bPC2ACpres)        0.14      0.01     0.12     0.16 1.00     6706     4263
## rescor(bPC1ACpres,bdurWordACpres)   -0.21      0.01    -0.24    -0.19 1.00     7489     4033
## rescor(bPC2ACpres,bdurWordACpres)   -0.12      0.01    -0.14    -0.10 1.00     1371     1332
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Appendix 3: Testing correlations among posterior samples extracted from separate univariate models

First, we extracted posterior samples for the `AccentualContext` effect from each univariate model. Since there was only one dependent variable per model, the effect name was consistently named `b_AccentualContextpresent`. To distinguish between them, we renamed each effect by appending the name of the corresponding dependent variable.

```
# f0_PC1
Greek_uni_f0_PC1_m1_b_AC <- Greek_uni_f0_PC1_m1 %>%
    spread_draws(b_AccentualContextpresent) %>%
    rename(b_f0PC1_AccentualContextpresent = b_AccentualContextpresent)
# f0_PC2
Greek_uni_f0_PC2_m1_b_AC <- Greek_uni_f0_PC2_m1 %>%
```

```
    spread_draws(b_AccentualContextpresent) %>%
    rename(b_f0PC2_AccentualContextpresent = b_AccentualContextpresent)
# durationWord
Greek_uni_durationWord_m1_b_AC <- Greek_uni_durationWord_m1 %>%
    spread_draws(b_AccentualContextpresent) %>%
    rename(b_durationWord_AccentualContextpresent = b_AccentualContextpresent)


Greek_uni_m1_b_AC <- cbind(Greek_uni_f0_PC1_m1_b_AC, Greek_uni_f0_PC2_m1_b_AC, Greek_uni_durationWord_m1_b
_AC)
```

Next, we fitted a multivariate model with all three effect estimates as dependent variables with the `student` family.

```
Greek_uni_m1_b_AC_cor <- brm(data = Greek_uni_m1_b_AC, family = student, bf(mvbind(b_f0PC1_AccentualContex
tpresent,
    b_f0PC2_AccentualContextpresent, b_durationWord_AccentualContextpresent) ~ 1) +
    set_rescor(TRUE), iter = 2000, warmup = 500, chains = 4, cores = 4)
```

Then, we extracted the posterior summaries of the residual correlation estimates, following the same procedure as in the multivariate analysis, and generated a paired scatterplot matrix to visualize the results.

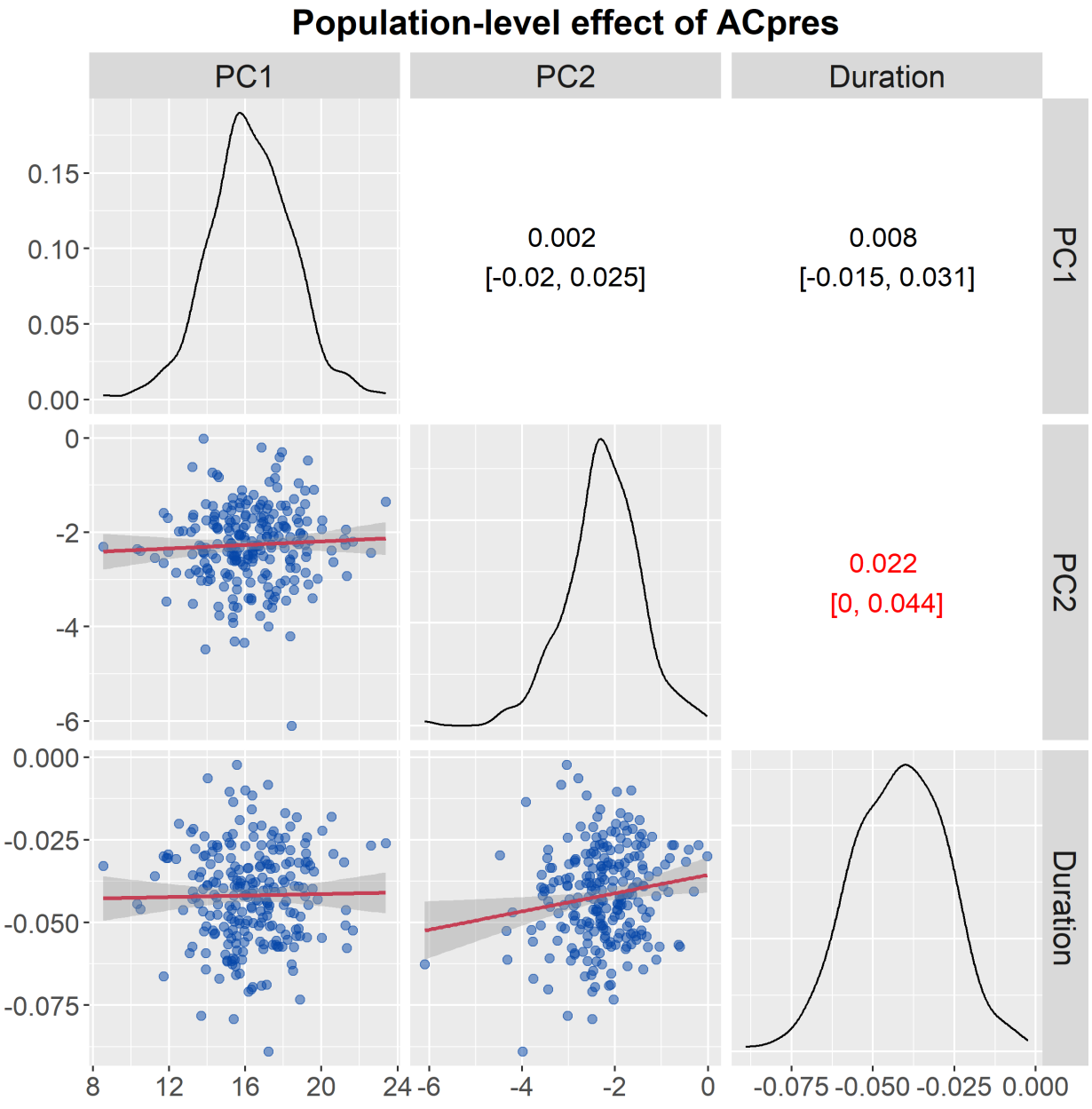

Figure 7: Paired scatterplot matrix showing the relationships between the effects of AccentualContext on PC1, PC2, and durationWord as predicted by the univariate models.

As expected, the results show no meaningful correlation between the parameters, in contrast with the results based on estimates from the multivariate model.