# An introduction to the generalized additive model

R. Harald Baayen and Maja Linke

**Abstract**

In this chapter we introduce the Generalized Additive Model (GAM). GAMs enable the analyst to investigate non-linear functional relations between a response variable and one or more predictors. Furthermore, GAMs provide a principled framework for studying interactions involving two or more numeric predictors. GAMs are an extension of the generalized linear model, and can therefore be used not only for Gaussian response variables, but also for binomial and Poisson response variables (and many others). Corpus linguists will find GAMs useful for coming to a detailed understanding of nonlinear patterns in their data, which can range from historical change (see, e.g., Ellegård, 1953) to the effects of corpus-based measures on acceptability ratings (e.g., Baayen and Divjak, 2017).

---

Department of Linguistics
University of Tübingen
harald.baayen@uni-tuebingen.de, maja.linke@uni-tuebingen.de

# 1 Introduction

The generalized additive model (GAM) offers the analyst an outstanding regression tool for understanding the quantitative structure of language data. An early monograph on generalized additive models is Hastie and Tibshirani (1990). The book by Wood (2006), a revised and expanded version of which appeared in 2017, provides the reader with both a thorough mathematical treatment, and a large number of detailed examples. Many of these come from biology, where the analyst faces challenges very similar to those faced by the corpus linguist. If one is interested in the density of mackerel eggs in the Atlantic east of France and the British Isles, one is faced with data that are unevenly spread out over a large area, where the ocean varies in depth, the gulf stream is of variable strength, and average temperature changes as one moves from Brittany to Scotland. A linguist interested in language use as it evolved in North America, as attested in the Corpus of Historical American English Davies (2010), similarly faces naturalistic data with a bewildering variety of properties. How the language changes over time varies with register, the education level of the writer, with the gender of the writer, with time, with social changes that come with immigration, and with technological developments. Crucially, one can hardly expect that effects of numerical predictors (henceforth covariates) will be strictly linear. Furthermore, covariates may interact nonlinearly with factorial predictors and with other covariates in ways that are difficult or even impossible to predict before initiation of data analysis.

Whereas a decade ago, bringing random effect factors into generalized additive models was not straightforward, recent versions of the **mgcv** package for `R` offer the analyst an excellent toolkit for dealing with multiple sources of noise relating to speakers and linguistic units (Wood, 2017). Working with the **mgcv** package is also substantially facilitated thanks to the **itsadug** package (van Rij et al., 2017).

Within linguistics, GAMs have been found useful in dialectometry and sociolinguistics (Wieling et al., 2011, 2014), phonetics (Wieling et al., 2016; Tomaschek et al., 2018), psycholinguistics (Linke et al., 2017; Milin et al., 2017), cognitive linguistics (Divjak et al., 2017; Baayen and Divjak, 2017) and historical linguistics (Baayen et al., 2017a). The goal of this chapter is to provide the reader with sufficient background to be able to understand the GAMs presented in these studies, and to start working with GAMs oneself. To this end, this chapter has three main parts, first a general introduction into common use cases that benefit from the application of generalized additive models, followed by a practical introduction to working with GAMs and a non-technical introduction to how GAMs work.

# 2 Fundamentals

In an ordinary least squares regression model, a response $y_i$ is modeled as a weighted sum of $p$ predictors and an error term that follows a normal distribution with zero mean.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma).$$

Although the linear predictor $\eta_i$,

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip},$$

may provide an adequate model for the functional relation between a response and its predictors, there are many cases in which the assumption of linearity is inadequate. Reaction times in lexical decision task, for instance, tend to decrease in a non-linear way as a function of words' frequency of

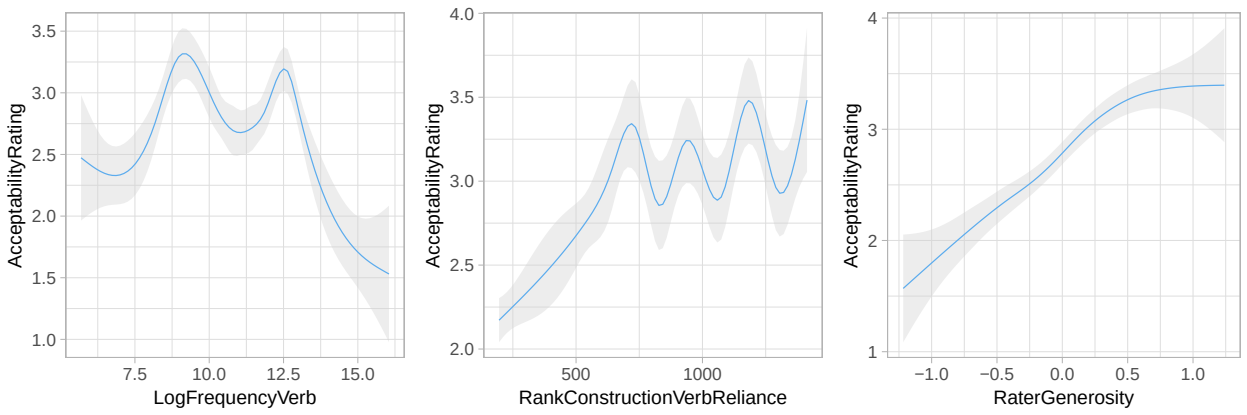



Fig. 1: Smooths for acceptability ratings as a function of frequency (left), construction verb reliance (center), and rater generosity (right) using the default smoother of **ggplot2**, `geom_smooth`.

occurrence in corpora. Modeling a non-linear response function as if it were linear not only results in inaccurate predictions, but also in structured errors that depart from the modeling assumptions about the relation between mean and variance. For Gaussian models, for instance, the errors may show heteroskedasticity, and when this happens, the validity of significances reported by the linear model is no longer assured and p-values listed in model summaries will be unreliable.

Consider, by way of example, Figure 1, which graphs acceptability ratings on a 5-point Likert scale for Polish sentences against three predictors: the frequency of the verb, construction-verb reliance (the frequency of a verb × construction combination given the frequency of the verb), and rater generosity, which gauges the extent to which participants tend to prefer the higher end of the rating scale. The first two predictors were transformed in order to avoid adverse effects of outliers. Figure 1 was obtained with `ggplot`, using its default method for visualizing nonlinear trends (`geom_smooth`).

```
ggplot(polish, aes(LogFrequencyVerb, AcceptabilityRating)) +
       geom_smooth()  # left panel of Figure 1
```

For each of the three panels, we observe departures from linearity. The left and center panels shows quite wiggly curves, and although the right panel reveals a nearly linear pattern, there is some leveling off for the highest values of the predictor. For two out of three predictors, a linear model appears to be inappropriate.

Figure 1 illustrates a property of GAMs which requires special attention: For the diagnostic plots shown, we used the `ggplot2` library default smoother `geom_smooth`, which defaulted to a smoothing method `gam`. The left and center panels of Figure 1 are overly wiggly, suggesting that **ggplot2**'s default settings for smoothing are overfitting and might actually not be appropriate for the Polish dataset. Although `geom_smooth` does provide a set of parameters to address this problem, adequate modification of the parameters is only feasible to an analyst equipped with a high level of understanding of the model and the data.

Consequently, the goal of this chapter is to provide the reader with sufficient background to be able to understand the GAMs presented in these studies, to start exploring working with GAMs oneself, and to evaluate whether GAMs have been used appropriately. Interpretation of models presented in this chapter requires a detailed understanding of the model, its implementation and a careful assesment of how both interact with the data set at hand. In what follows, we begin with recapitulating the basic concepts of the generalized linear model. Next, we introduce key concepts

underlying the generalized additive model. We then present a worked example of how GAMs can be used to obtain a thorough understanding of the quantitative structure of linguistic data.

### 2.1 The generalized linear model

Central to the generalized linear model is the idea that a response variable $Y_i$ for a datapoint $i$ that is described by $p$ predictors $x_1, x_2, \ldots, x_p$ is a random variable. For real-valued response variables, we assume that the probability $\Pr(Y_i = y_i | x_{i1}, x_{i2}, \ldots, x_{ip})$ follows a normal distribution with variance $\sigma^2$ and mean $\eta_i$:

$$\Pr(Y_i = y_i | x_{i1}, x_{i2}, \ldots, x_{ip}) \sim \mathcal{N}(\eta_i, \sigma^2),$$

where the linear predictor $\eta_i$ is given by an intercept $\beta_0$ and a weighted sum of the $p$ predictor values:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ip}.$$

The means $\mu_i = \eta_i$ are linear functions of $x$ (see the left panel of Figure 2). For each value of $x$, 20 randomly drawn values are shown. Note that the Gaussian model provides, for each value of $x$, the probability of the response. The most probable value is the mean. The scatter of the observed values around the mean is constant across the full range of the predictor.

For count data, a Poisson model is often used, with the same linear predictor $\eta_i$:

$$\Pr(Y_i = m | x_{i1}, x_{i2}, \ldots, x_{ip}) \sim \text{Poisson}(e^{\eta_i}).$$

Thus, the logarithm of the observed count is linear in the predictors. In this way, we ensure that predicted counts can never be negative. As can be seen in the center panel of Figure 2, the expected counts themselves are a nonlinear function of $x$. The variance of the counts, which for Poisson random variables is equal to the mean, increases with $x$.

When the response variable is binary (as for successes versus failures, or correct versus incorrect responses), we are interested in the probability of a success, which we model as a binomial random variable with a single trial and a probability of success $e^{\eta_i}/(1 + e^{\eta_i})$, i.e.,

$$\Pr(Y_i = 1 | x_{i1}, x_{i2}, \ldots, x_{ip}) \sim \text{binom}\left(\frac{e^{\eta_i}}{1 + e^{\eta_i}}, 1\right),$$

where the linear predictor $\eta_i$ again is defined exactly as above. In this case, the log odds (i.e, the logarithm of the ratio of successes to failures) is linear in the predictors. As can be seen in the right panel of Figure 2, for binomial random variables, the variance is greatest for $p = 0.5$, which in this example is the case when $x = -0.1/0.3 = -0.33$. Here, we observe the greatest overlap (with respect to $x$) for (jittered) failures and (jittered) successes.

The linear predictor is not restricted to expressing a "linear" functional relation between $\eta$ and the predictors. For instance, the linear predictor

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2$$

specifies a parabola rather than a straight line. In fact, very wiggly curves can be obtained by adding multiple powers of $x$ as predictors. This is illustrated in Figure 3. Instead of writing

$$\eta_i = \beta_0 x^0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \ldots + \beta_s x_i^s,$$

we can state the model more succintly as

$$\eta_i = \sum_{j=1}^{s} \beta_j x_i^j = f(x_i).$$

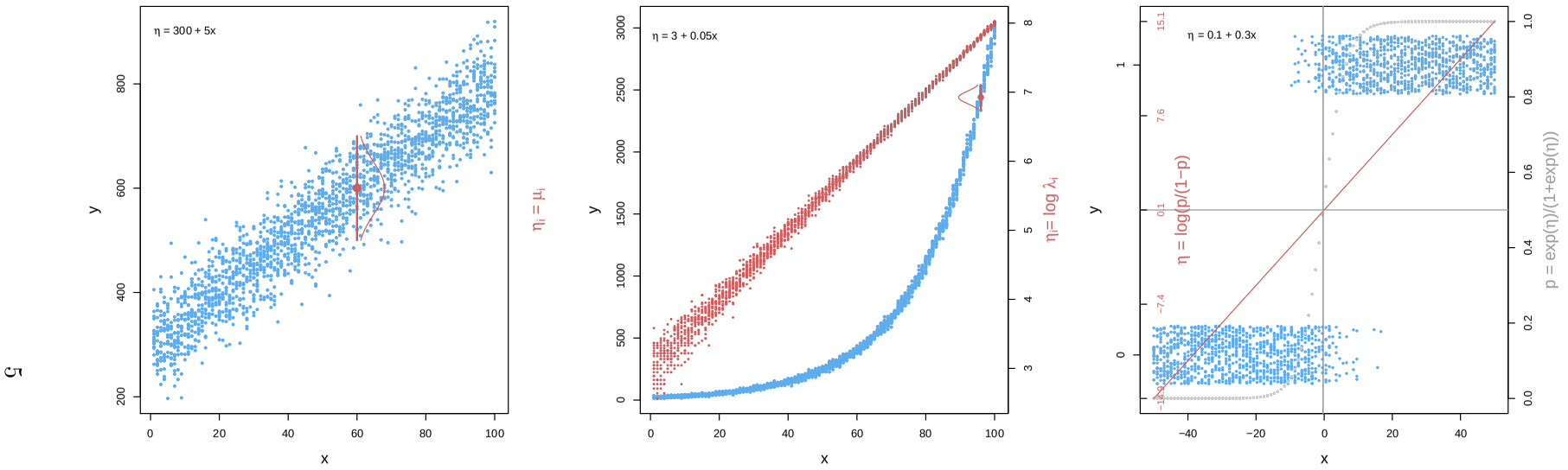



Fig. 2: Gaussian (left), Poisson (center), and binomial (right) models. Across models, 20 random values were generated for each value of the predictor $x$. For the Gaussian model, the linear predictor is $\eta = 300 + 5x$. For $x = 60$, mean ($\mu_i = \eta_i$) and the corresponding 95% confidence interval are highlighted. For the Poisson model, the linear predictor (in red) is $\eta = 3 + 0.05x$. This linear predictor specifies the logarithm of the Poisson parameter $\lambda$. For $x = 96$, the mean of the predicted count ($\exp(\eta_i)$) is shown together with its 95% confidence interval. For the binomial model, the linear predictor (in red) is $\eta = 0.1 + 0.3x$. The linear predictor specifies the log odds, the corresponding probabilities are shown in gray. The blue dots represent failures (0) and successes (1); jitter was added to bring out individual outcomes. The vertical gray line is at $x = -0.33$, here $\eta = 0$ and $p = 0.5$. The horizontal gray line is at $\eta = 0.1$, for which $p = 0.52$.

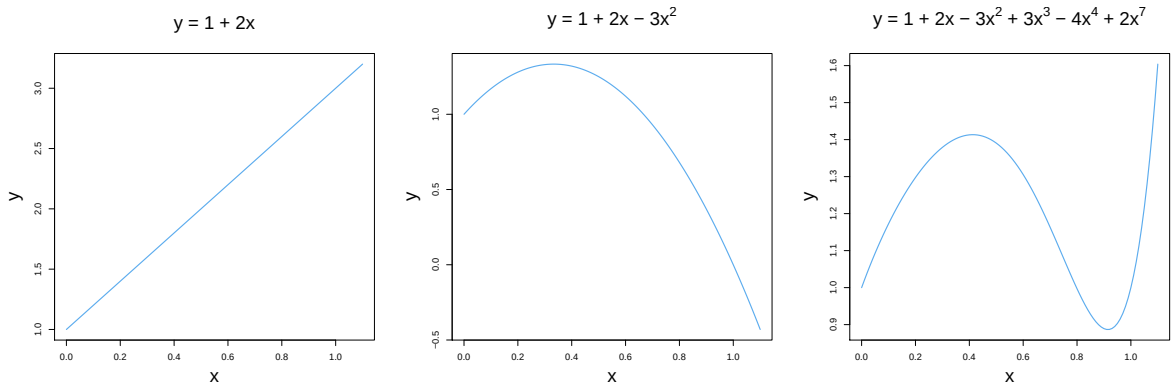



Fig. 3: Three polynomial functions. By adding powers of $x$ as predictors, increasingly wiggly curves are obtained.

Unfortunately, when $f(x)$ is set up as a polynomial of $x$, as we have done here, with the aim of using this polynomial as a predictor of the response, it turns out that doing so comes with substantial disadvantages. Polynomial functions require far too many parameters (the $s$ weights $\beta$), and they can perform very poorly when predicting the response for unseen data. In other words, polynomial functions overfit one's data, and cannot be used for prediction and generalization. This is where the generalized additive model comes in.

## 2.2 The Generalized additive model

The generalized additive model takes the linear predictor $\eta_i$ of the generalized linear model and enriches it with functions of one or more predictors, as, for instance:

$$\eta_i = \underbrace{\beta_0 + \beta_1 x_{i1}}_{\text{parametric part}} + \underbrace{f_1(x_{i2}) + f_2(x_{i3}, x_{i4})}_{\text{non-parametric part}}. \tag{1}$$

The parametric part is familiar from the generalized linear model. The non-parametric part specifies two functions, one of which takes $x_2$ as argument, and one of which takes two predictors, $x_3$ and $x_4$, as arguments. Instead of using polynomial functions, GAMs use smoothing splines for functions such as $f_1$ and $f_2$. A smoothing spline with one predictor as argument is used for fitting wiggly curves. A smoothing spline with two predictors can be used for fitting a wiggly surface. Splines can take more than two arguments, in which case wiggly hypersurfaces are modeled. Given a linear predictor with appropriate smooths, this linear predictor can be used to model Gaussian response variables, or Poisson or binomial responses. GAMs can also accomodate ordinal responses as well as multinomial responses.

In order to use GAMs appropriately, it is useful to have a general understanding of how smoothing splines work. Here, we illustrate one particular spline that is the default of the **mgcv** package (Wood, 2017), the so-called thin plate regression spline. The lower right panel of Figure 4 presents a thin plate regression spline smooth $f(x)$ estimated for the effect of a predictor $x$ on the response. This effect is known as the partial effect of $x$, as in models such as (1) there typically are many other predictors that also contribute to the predicted value of the response. The partial effect shown in the lower right panel is a weighted sum of the curves in the other 9 panels of Figure 4. Such elementary curves are known as basis functions.

The first two basis functions (in the upper left) are straight lines that are completely smooth. For predictors that have a strictly linear effect, these two basis functions suffice. Each basis function

is associated with a weight (shown in square brackets on the vertical axes). For straight lines, the weight for the first basis function is the intercept, and the weight for the second basis function is the slope.

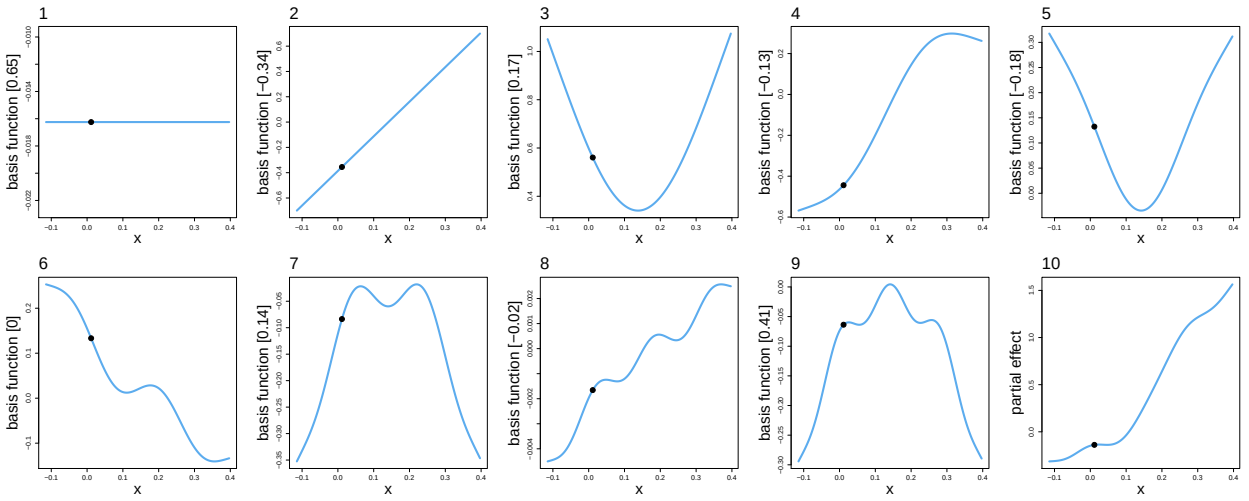



Fig. 4: A smooth using thin plate regression spline basis functions for the partial effect of a predictor $x$ on the response. The first 9 panels represent the basis functions, multiplied by their weights (given in square brackets on the vertical axes). The sum of these weighted basis functions results in the spline smooth in the lower right panel.

In this example, the predictor has a nonlinear effect, so we need more basis functions than just the first two. Figure 4 illustrates 7 additional basis functions, which become increasingly wiggly as we proceed from panel 3 to panel 9. The curve in panel 10 is the sum of all nine (weighted) basis functions. Thus, the vertical position of the black dot on the red curve in panel 10 is the sum of the vertical positions of the other black dots in panels 1–9 (the scaling on the vertical axes already incorporates the weighting).

The exact mathematical form of the wiggly basis functions is determined by the total number of basis functions requested by the user. Below, we return to the practical question of how to choose the number of basis functions. Here, we proceed on the assumption that a sufficiently large number of basis functions has been requested by the user.

The question that arises at this point is how to avoid the situation in which we have so many basis functions that we edge too close to the datapoints and start fitting the model to noise rather than signal. In other words, we need to find a good balance between undersmoothing (resulting in too much wiggliness) and oversmoothing (resulting in missing out on significant wiggliness). The solution offered by GAMs is to balance two constraints, one constraint demanding that we want to stay faithful to the data (by minimising the summed of squared errors), and one constraint that demands we keep our model as simple as possible. This second constraint can be rephrased as a prior belief that the truth is likely to be less wiggly rather than very wiggly. This belief is a restatement of Occam's razor, in that we don't want to make our theory more complex (by adding basis functions and associated weights) than is absolutely necessary. The two opposing constraints lead to a model cost $C_f$ for a smooth $f$,

$$C_f = \underbrace{\sum_{i=1}^{n}(y_i - f(x_i))^2}_{\text{faithfulness to the data}} + \underbrace{\lambda \int f''(x)^2 dx}_{\text{Occam's razor}},$$

that we want to keep as small as possible. This cost consists of two parts. To the left of the + we have the sum of squared deviations between the observed values $y_i$ and the values $f(x_i)$ that are predicted by the smooth. This is what an ordinary least squares regression model fitted with `lm` minimizes. To the right of the +, we have the integral of the squared second derivative of the smooth, weighted by a smoothing parameter $\lambda$. The integral over the squared second derivative is a measure of the wiggliness of the smooth which we also want to keep as small as possible. The parameter $\lambda$ regulates the balance between the desire to remain faithful to the data and the desire to keep wiggliness down.

The introduction of the smoothing parameter $\lambda$ raises a new question, namely, how to estimate $\lambda$. A first step towards a solution is to assume that the weights of the basis functions follow a normal distribution with mean zero and some unknown standard deviation $\sigma_s$. It turns out that the choice of $\lambda$ co-determines $\sigma_s$. This leads to the second step, namely, to choose some $\lambda$, sample from the (normal) distribution of weights for the smooth implied by $\lambda$, and keep tuning $\lambda$ until an optimal fit is obtained. This typically results in weights for a smooth that are smaller than if $\lambda$ were zero, i.e., when there is no penalization for wiggliness and all that counts is faithfulness to the data. This method has been shown to also yield good estimates for confidence intervals (Nychka, 1988; Marra and Wood, 2012).

Importantly, penalization for wiggliness can reduce the weights of basis functions to zero, in which case the pertinent basis functions are apparently unnecessary. For instance, when a smooth is fitted to data for which the functional relation between the response and a predictor is truly linear, all the weights for the wiggly basis functions in Figure 4, i.e., the basis functions in panels 3–9, are driven to zero, leaving untouched only the completely smooth (i.e., completely non-wiggly) first two basis functions. These two basis functions jointly determine a straight line. The horizontal basis function is merged into the intercept specified in the parametric part of the model, to ensure that the model remains identifiable.[1] Thus, the only weight of the smooth that remains is that for the slanted line. This weight is simply the slope of the regression line.

When the functional relation between response and a predictor is in fact non-linear, penalization will retain at least some wiggly basis functions, but with weights that are reduced compared to an unpenalized smooth with $\lambda = 0$. The proportion of the original weight of a basis function that is retained after penalization is referred to as the effective degree of freedom (edf) of that basis function. The sum of the effective degrees of freedom of all basis functions used to construct a smooth constitutes the effective degrees of freedom of that smooth. Summary tables for the smooth terms in a GAM list these effective degrees of freedom, which enter into a special $F$-test that is used to evaluate the significance of a smooth.

The edf of a smooth cannot be larger than $k$, the number of basis functions that is set by the user. If the edf for a smooth is close to $k$, hardly any penalization has occurred, and it is likely that a larger value of $k$ should be chosen (see the documentation of `choose.k` of **mgcv**) for detailed discussion). When penalization leaves a predictor with 1 edf, its effect is likely to be linear: All wiggly basis functions will have been taken out of commission by setting their weights to zero, and only the weight for the second basis function is retained (i.e, the slope of the regression line).

GAMs also accomodate random effect factors as predictors. The way in which this is done is different from the method used by the linear mixed model as implemented in the **lme4** package. The GAM implementation that we discuss here makes use of the mechanism of penalization to estimate the parameters of random-effect factors, using a so-called ridge penalty. This ridge penalty takes the sum of the absolute values of the random-effect coefficients, and seeks to keep this sum as small as

[1] If there are two specifications for the intercept in the model, an arbitrary amount $\delta$ can be taken from the one and added to the other without changing model predictions; in this case there are infinitely many models too choose from, with no principled reason for selection.

possible. In this way, parameters are shrunk towards zero, just as in the linear mixed effect model. However, computation times for mixed GAMs (henceforth GAMMs) are typically longer than for the corresponding models fitted with **lme4**, which is due to GAMMs not making any simplifying assumptions about the structure of the random effects.

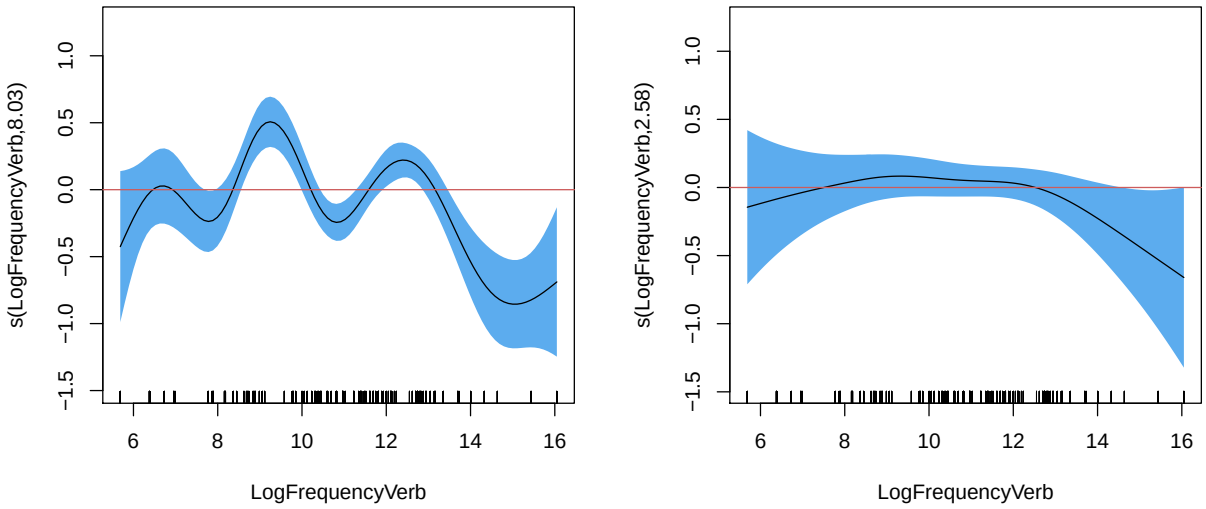



Fig. 5: Thin plate regression spline smooths for log verb frequency as predictor of acceptability ratings. Left panel: a model without by-verb random intercepts, right panel: a model that does include by-verb random intercepts.

Proper inclusion of random effects in the model specification protects against overly wiggly curves. Recall that Figure 1, obtained with **ggplot2**'s default smoother, produced highly wiggly and undulating curves for two out of three predictors. However, once `Verb` is included as random-effect factor, the partial effects of the predictors become much less wiggly (compare the left and right panels of Figure 5). A highly wiggly curve with narrow confidence bands is replaced by a shallow curve with wide confidence bands. The fact that the horizontal axis is included in the 95% confidence band for nearly all values of verb frequency indicates that a main effect of verb frequency is highly unlikely to be significant. Indeed, the p-value for this smooth provided by the model summary (not shown) is 0.22.

The reason that the smooth in the model without a random effect for `Verb` produces such a wiggly curve is that for each verb frequency we have as many repetitions as there are subjects. As a consequence, each distinct frequency value comes with substantial evidence that stands in the way of proper penalization. By including random intercepts for `Verb`, the idiosyncracies of individual verbs can be taken into account, and for the present example, the evidence for an undulating effect due to frequency evaporates.

**Representative study: Baayen, R. H. and Divjak, D. (2017). Ordinal GAMMs: a new window on human ratings.**

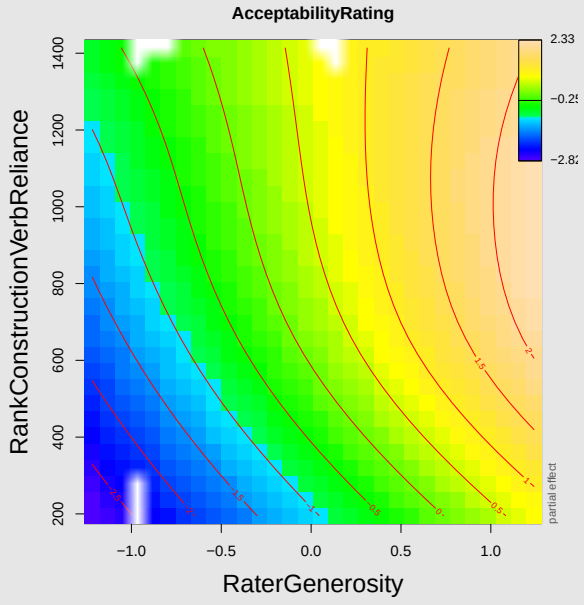
**Research Questions:**
Divjak(2016) investigated the extent to which speaker's experience contributes to the acceptability ratings for infinitival and finite *that*-complements in Polish. Baayen and Divjak (2017) reanalyzed these data using ordinal GAMs. Key questions are whether the frequency of occurrence of the verb (in the Polish National Corpus), and the conditional probability of the construction given the verb (reliance) are predictive for acceptability ratings. Rater generosity (i.e., the extent to which a rater is prone to give high ratings) was included as a control variable.

**Data:** The data set contains off-line acceptability ratings for verbs that occur with low frequency in *that*-constructions. A total of 95 verbs in *that*-constructions was presented to 285 undergraduate students of English/German in Poland. Participants were asked to rate "how Polish a sentence sounds" on a 5-point Likert scale. Each verb was responded to by 15 participants. The data set is available as polish.rda at https://opendata.uit.no/dataverse/trolling.

**Method:** Ratings elicited on a Likert scale yield ordinal data. Baayen and Divjak (2017) therefore used an ordinal GAM, which models the probability that a rating is $r$ ($r = 1, 2, \ldots, 5$) through the probability that a latent variable $u$ falls into the $r$-th interval defined on the real axis. Three models were compared: a model with (nonlinear) main effects only, a model with all pairwise interactions, and a model with a three-way interaction.

```
polish.gam = gam(AcceptabilityRating   te(RankConstructionVerbReliance, LogFrequencyVerb,
RaterGenerosity) + s(Verb, bs = "re") , data = polish, family = ocat(R = 5))
```

**Results:** Of the three models, the model with the three-way interaction, fitted with a tensor product smooth, provided the best fit to the ratings. Subjects with lower rater generosity showed stronger effects of frequency and reliance. Furthermore, ratings decreased for increasing frequency when reliance was low, and ratings increased only with reliance for high-frequency verbs (see the inset contour plot below). The GAM analysis succeeded in bringing together within one model a series of findings that Divjak (2016) could account for only in part with an ordinal linear model.



# 3 Practical guide with R

The dataset that we use to illustrate how to work with GAMs is taken from the Buckeye corpus of conversational American English as spoken in Columbus, Ohio (Pitt et al., 2005). In what follows, we restrict ourselves to the data of one speaker (S40). For this speaker, we compared the words as realized by this speaker with the corresponding dictionary pronunciations. For the first word uttered, *allright*, the dictionary form (in ARPAbet notation) is `aa l r ay t`. The speaker actually pronounced `ao r eh t`. For each word, we extracted its successive diphones for the dictionary form as well as those of the form actually produced (`ObsDiphone`), and checked for each dictionary diphone (`DictDiphone`) whether it was absent in the actual realization, irrespective of its position in the word. For a particular utterance of *allright*, results look as follows.

```
  DictDiphone DictDiphonePosition ObsDiphone DictDiphoneAbsent
1         #aa                   1        #ao              TRUE
2         aal                   2        aor              TRUE
3          lr                   3        reh              TRUE
4         ray                   4        eht              TRUE
5         ayt                   5         t#              TRUE
6          t#                   6          -             FALSE
```

This information was collected for all words, in the order in which they appear in the corpus, resulting in a table with 27062 observations, one for each diphone. For each of these 27062 diphones, we considered the following variables: `DictDiphoneAbsent`, with values `TRUE` or `FALSE`, depending on whether the dictionary diphone was realized by the speaker; this is the response variable for our analyses; `DictDiphonePosition`, an integer indicating the position of the dictionary diphone in the word; `DictDiphoneCount`, an integer with the number of dictionary diphones in the word; `PhraseInitial`, with values `TRUE` or `FALSE`, indicating whether the word carrying the diphone is phrase-initial; `PhraseFinal`, with values `TRUE` or `FALSE`, indicating whether the word carrying the diphone is phrase-final; `PhraseLength`, an integer indicating the length in words of the phrase; `PhraseRate`, the speech rate (number of syllables per second); `LogDuration`, the logarithm of the duration of the word (in seconds); `DictDiphoneActDiversity`, a measure, based on discriminative learning, gauging the lexical uncertainty caused by the diphone; `WordActDiversity`, a measure gauging the lexical uncertainty of the carrier word in a semantic vector space model derived with discriminative learning; `SemanticTypicality`, the extent to which the semantic vector of the carrier word is similar to the average semantic vector; and `CorpusTime`, the position of the diphone in the corpus (ranging from 1 to 27062) but scaled and centered to make this variable commensurable with the other numeric predictors. A detailed description of these predictors is available in Tucker et al. (2018).

```
load("data/buckeye.rda")
head(buckeye, 6) # variables for `allright'
```

```
  PhraseInitial PhraseFinal PhraseLength WordActDiversity SemanticTypicality
1          TRUE        TRUE            1       0.05973968          0.0401964
2          TRUE        TRUE            1       0.05973968          0.0401964
3          TRUE        TRUE            1       0.05973968          0.0401964
4          TRUE        TRUE            1       0.05973968          0.0401964
5          TRUE        TRUE            1       0.05973968          0.0401964
6          TRUE        TRUE            1       0.05973968          0.0401964
  DictDiphoneCount PhraseRate LogDuration DictDiphonePosition
1                6    5.12885  -0.9417342                   1
2                6    5.12885  -0.9417342                   2
```

```
3                6    5.12885  -0.9417342                   3
4                6    5.12885  -0.9417342                   4
5                6    5.12885  -0.9417342                   5
6                6    5.12885  -0.9417342                   6
  DictDiphoneActDiversity CorpusTime DictDiphoneAbsent    Word
1               2.5053406  -1.540405              TRUE alright
2               2.4392060  -1.540301              TRUE alright
3               0.3853351  -1.540198              TRUE alright
4               1.9609812  -1.540095              TRUE alright
5               2.1347266  -1.539991              TRUE alright
6               2.1825928  -1.539888             FALSE alright
```

Note that many variables are 'piece-wise' constant by word. For *allright*, the only variables (out of 12) that are not repeated 6 times (once for each diphone in the dictionary pronunciation) are `DictDiphonePosition`, `DictDiphoneActDiversity`, `CorpusTime`, and the response variable, `DictPhoneAbsent`.

### 3.1 A main-effects model

We begin with fitting a standard logistic model in which the log odds is assumed to vary linearly with the numeric predictor variables. We use the `bam` function from the **mgcv** package (Wood, 2017) (version 1.8.24), but exactly the same results are obtained with the `glm` function of base R.

```
m1 = bam(DictDiphoneAbsent ~ PhraseInitial + PhraseFinal + PhraseLength +
                             PhraseRate + LogDuration + DictDiphoneCount +
                             DictDiphonePosition + WordActDiversity +
                             SemanticTypicality + DictDiphoneActDiversity +
                             CorpusTime,
         data = buckeye, family = "binomial")
```

With the exception of `PhraseRate` ($p = 0.0704$), all predictors receive good support (the model summary is available in the supplementary materials). The AIC for this baseline model is:

```
AIC(m1)

[1] 29939.64
```

The assumption that a covariate has a strictly linear effect may be true, but it may also be unjustified. Often, exploratory data analysis will be required to establish whether, and for which variables, the linearity assumption is inappropriate. The following model relaxes the linearity assumption for all covariates using the `s` smoothing function from **mgcv**. The amount of wiggliness that a smooth allows for is controlled by the number of basis functions $k$, which has 10 as default value. This default is not motivated theoretically, and hence is a heuristic starting point. What is important is that $k$ has to be set to an integer value (the 'dimension' of the smooth) that is large enough. How large an initial value of $k$ should be depends on the number of different values of the predictor for which a spline is required. If there is only a handful of different values, one may want to set $k$ to 3 or 4. If there are thousands of different values, a possible value would be 200.

For the numeric predictors in the present data, we proceed as follows. We have two counts with a limited range, `DictDiphoneCount` (7 distinct values) and `DictDiphonePosition` (8 distinct values). The dimension of a smooth should be lower than the number of distinct values, so we choose $k = 5$. For `PhraseRate` (1171 distinct values), `LogDuration` (5842 distinct values),

`PhraseLength` (29 distinct values, we take the logarithm of this variable to reduce its rightward skew), `DictDiphoneActDiversity` (604 distinct values), `WordActDiversity` (825 distinct values), and `SemanticTypicality` (825 distinct values) we go with the default. `CorpusTime`, however, has no less than 27062 distinct values, and the default value of $k$ therefore comes with the risk of oversmoothing. We therefore set $k$ to 100.

```
buckeye$LogPhraseLength = log(buckeye$PhraseLength)
m2 = bam(DictDiphoneAbsent ~ PhraseInitial + PhraseFinal +
                             s(DictDiphoneCount, k = 5) +
                             s(DictDiphonePosition, k = 5) +
                             s(LogPhraseLength) +
                             s(PhraseRate) +
                             s(LogDuration) +
                             s(DictDiphoneActDiversity) +
                             s(WordActDiversity) +
                             s(SemanticTypicality) +
                             s(CorpusTime, k = 100),
         data = buckeye, family = "binomial")
```

It is noteworthy that by allowing predictors to have nonlinear effects, we have obtained a substantially improved fit:

```
AIC(m2)

[1] 27722.93
```

with a decrease in AIC of no less than 2216.7.

Figure 6 presents the partial effects of the covariates. These partial effects are centered around zero, and represent deviations from the group means defined by the factorial predictors (here `PhraseInitial` and `PhraseFinal`) when other covariates are held constant at their most typical value. In other words, the partial effect of a term in the model specification is the contribution of that term to the model predictions. The command

```
plot(m2, pages=1)
```

presents all smooths in the model in a one-page multipanel figure. Figure 6 also presents the partial effects, but uses customized code (available in the supplementary materials) that adds histograms or density plots and that also highlights, by means of vertical red lines, the 5, 25, 75, and 95 percentiles of the predictors.

Panels 1 and 2 of Figure 6 indicate that there is no effect of `PhraseLength` and `PhraseRate`: the X-axis (the horizontal red line) is within the 95% confidence interval for the full range of predictor values. The effects of all other predictors are non-linear.

The log odds of deviation from the dictionary norm increases with the number of diphones, but levels off for words with more than 5 diphones (panel 3). Panel 4 clarifies that the later the position of the diphone in the word is, the greater the log odds of deviation. Apparently this effect reverses for positions 7 and 8 (upper center panel). Here, however, data are sparse.

Unsurprisingly, the log odds of diphone deviation decreases with word duration (panel 5). As documented in detail by Johnson (2004), segment and even syllable deletion is common in this corpus, and as words that do not have deletions typically will be longer, a negative trend for the bulk of the data is expected. However, the distribution of durations has a few large-valued outliers, which give rise to the upward swing in the right-hand side of the plot. Due to the sparsity of data,

the confidence intervals are wide. The reason that these outliers nevertheless are taken seriously by the GAM is that the same (log) duration is repeated as many times for the six words with log duration exceeding zero as these words have diphones. As a consequence, this handful of words has stronger support than just the small number of words would suggest.

We see here an important advantage of GAMs over models that force the effect of duration to be linear. In such models, outliers may exert high leverage on the regression, and typically have to be removed from the data set. By contrast, the GAM clarifies that outliers behave differently, highlights the associated uncertainty with wide confidence intervals, and at the same time does not let the outliers influence conclusions about the effect of a predictor for the bulk of the data. In other words, GAMs provide the full picture, and protects the analyst against models based on flattened and simplified data.

The distribution of `DictDiphoneActDiversity` (panel 6) has a long tails. Here, we find an S-shaped curve. For the interquartile range (the center 50% of the data highlighted by the center vertical red lines), we observe an increase in the log odds with increasing activation diversity. The effect goes back to zero, however, for the first and third quartiles. Strongly undulating patterns are likewise visible for `WordActDiversity` (panel 7) and `SemanticTypicality` (panel 8).

The wiggliness of these curves is difficult to interpret theoretically. For cases such as these, the analyst has two options. The first option is to accept that these undulations are real, and that our theoretical understanding is too limited, or that our predictor is theoretically flawed. The second option is to reduce the dimension of the smooth. For the present data, such a reduction has some justification because of the abovementioned problem that lexical and phrasal variables are constant within words, a kind of problem that often arises when working with observational data from corpora. Since data points are not independent in the way one would like them to be, some conservatism with respect to nonlinearity is justified. When the model is refit with $k$ set to 5, the functional form of these effects becomes much simpler and easier to understand, as we shall see below (Figure 7). Simpler curves come at the cost of a reduction in the quality of the fit (difference in AIC: 454), but the model remains far superior to the model imposing linearity on the relation between the response and the predictors (difference in AIC: 1762.73).

The effect of `Corpus Time` in the lower right panel is quite wiggly, but as we are dealing with a predictor with 27062 distinct values, and as we have no a-priori hypothesis about how deviation probabilities might change in the course of the interview, we accept the smooth as providing a description of real changes over time in diphone deviation behavior.

### 3.2 A model with interactions

Thus far, we have considered models with main effects only. In this section, we consider interactions involving numerical covariates. There are two basic types of interaction: an interaction of a covariate with a factor, and an interaction of two covariates. First consider the interaction of a numerical predictor with a factorial predictor such as `PhraseFinal`. `PhraseFinal` has two levels (`TRUE/FALSE`), and an interaction of `PhraseFinal` with a covariate, say `SemanticTypicality`, requests two smooths for this covariate, one for phrase-final words and one for words that are not phrase-final. We request the two curves from the `bam` function with the `by` directive in the call to `s`:

```
s(SemanticTypicality, k = 5, by = PhraseFinal)
```

An update of model `m2` that includes several interactions,

```
m4 = bam(DictDiphoneAbsent ~ PhraseInitial + PhraseFinal +
                             s(DictDiphoneCount, k = 5) +
                             s(DictDiphonePosition, k = 5) +
```

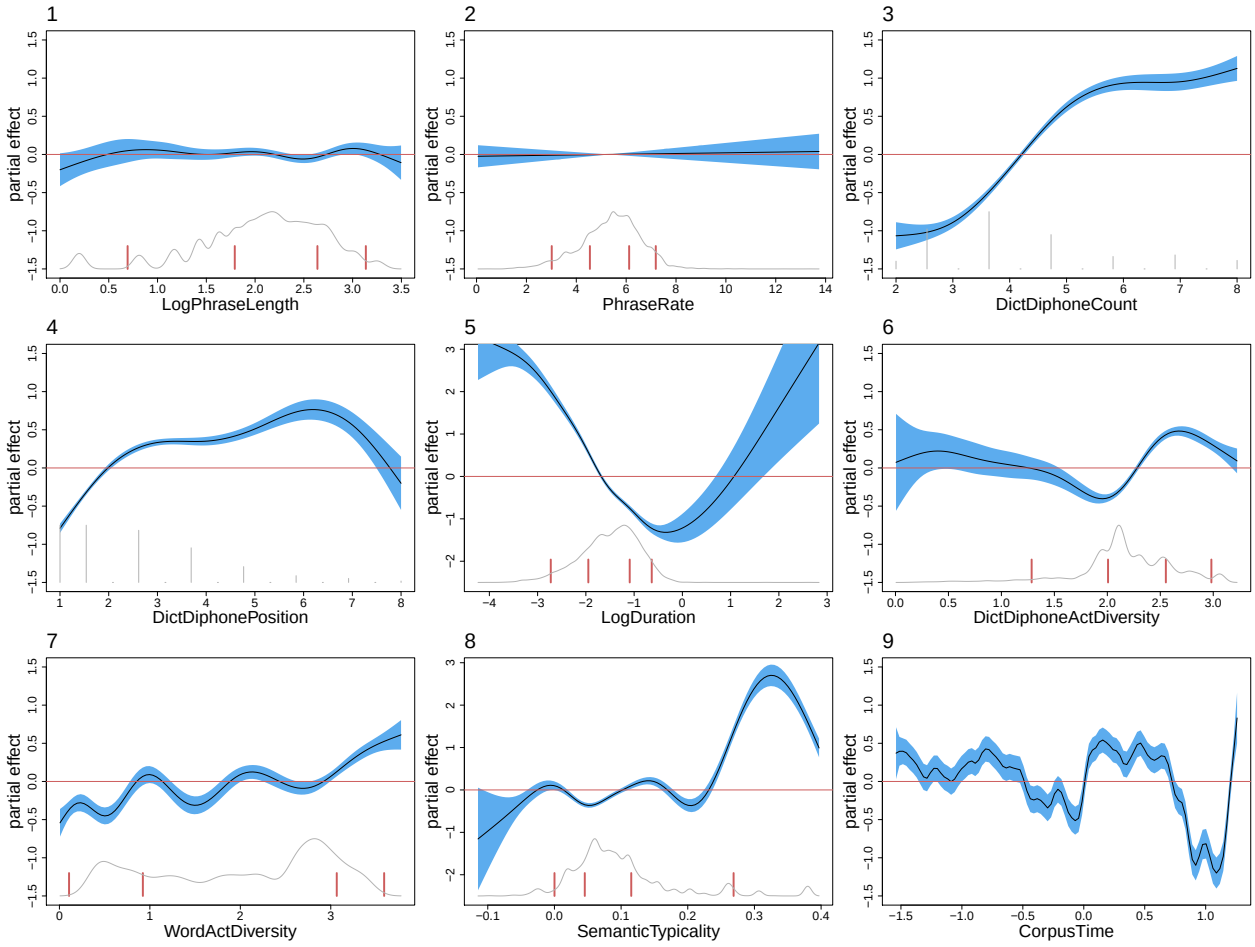



Fig. 6: Partial effects of predictors in a GAM (`m2`) for the log odds of diphone deviation for speaker 40 in the Buckeye corpus.

```
                            s(LogPhraseLength) +
                            s(LogDuration) +
                            te(WordActDiversity, PhraseRate, k = 5, by = PhraseFinal) +
                            s(SemanticTypicality, k = 5, by = PhraseFinal) +
                            s(DictDiphoneActDiversity, k = 5, by = PhraseFinal) +
                            s(CorpusTime, k = 100),
          data = buckeye, family="binomial")
AIC(m4)

[1] 28103.35
```

and which offers an improved fit (AIC: 28103.35) compared to model `m3` (AIC: 28176.91). The partial effects of `m4` are presented in Figure 7. Panels 1 and 2 indicate that the effect of `SemanticTypicality` is stronger in phrase-final position. The effect of this variable appears to be present primarily across its fourth quartile. Panels 3 and 4 reveal an effect of `DictDiphoneActDiversity` that is U-shaped for the bulk of the data points. The largest effect is again present for diphones in words that are in phrase-final position. The downward swing for low activation diversity in the left of panel 4 appears due to a small number of outliers.

The bottom panels of Figure 7 illustrate the three-way interaction of this model, which involves

two covariates, `WordActDiversity` and `PhraseRate`, and one factor, `PhraseFinal`. This interaction was specified with the model term

```
te(WordActDiversity, PhraseRate, by = PhraseFinal, k = 5)
```

Here, `te` requests a tensor product smooth,[2] which estimates a wiggly regression surface (or hypersurface). Such surfaces are visualized with contour plots. In these contour plots, just as in geographical maps indicating terrain height, contour lines connect points with the same partial effect. There are two ways in which the contour map can be shown: one in which 1 SE confidence regions are added (panels 5 and 7), and one in which color coding is used to represent the magnitude of the partial effect (panels 6 and 8). In panels 5 and 7, dotted green lines are 1 SE up from their contour lines, and dashed red lines are 1 SE down. In panels 6 and 8, darker shades of blue indicate lower values, and darker shades of yellow, higher values. With the directive `by=PhraseFinal`, we requested two wiggly surfaces, one for diphones in words that are not phrase-final (panels 5 and 6), and one for diphones in phrase-final words (panels 7 and 8). In panels 6 and 8, contour lines are 0.2 units apart. Comparing the color shadings, it is clear that effects are much stronger in phrase-final position. Comparing panels 5 and 7, it is also clear that 1 SE confidence regions are considerably tighter in phrase-final position. Unlike panels 6 and 8, panels 5 and 7 are informative about where there is a significant gradient. In panel 5, for instance, confidence regions of adjacent contour lines begin to overlap for high phrase rates, indicating the absence of a significant effect.

For understanding contour plots, it can be useful to trace changes in the value of the response with imaginary lines that are parallel to the axes. For instance, for `PhraseRate` to have an effect, contour lines should be crossed when moving in parallel to the y-axis. For words that are not phrase final, this does not happen for low values of `WordActDiversity`. It is only for higher values of this activation measure that an effect becomes visible, with larger phrase rates indexing reduced log odds of diphone deviation. When we consider imaginary horizontal lines, we cross more contour lines for low phrase rates than for high phrase rates, indicating that there is a stronger gradient up for `WordActDiversity` when `PhraseRate` is relatively low. It is noteworthy that for phrase-final words, the effect of `PhraseRate` reverses, such that higher phrase rates predict increased instead of decreasing log odds of diphone deviation.

Table 1 provides a summary of model `m4`, obtained by applying `R`'s general `summary` function to the model object (`summary(m4)`). The upper part of the table provides the statistics familiar from the generalized linear model for the parametric part of the model. The lower part of the table provides an evaluation of the significance of the smooth terms, using tests that are described in Wood (2013a,b). In Table 1, `LogPhraseLength` is associated with 1 effective degree of freedom (edf), suggesting a linear effect of this predictor. However, from the high p-value (0.79) it is clear that the slope of this regression line is effectively zero. Significant linear effects will show up with 1 edf and a low p-value. To obtain an estimate of the actual slope of a regression line, the model can be refitted without the smooth, in which case the slope will be listed in the parametric part of the model. An important property of GAMs is that if a predictor has a truly linear effect, the algorithm will discover this, and remove all wiggliness, leaving a straight line. GAMs only admit wiggliness where wiggliness is truly justified.

Table 1 lists summary statistics for six smooths, two smooths for each of the three interactions with `PhraseFinal`. The p-values for these smooths inform us about whether these individual smooths are likely to be just a flat horizontal line, or a flat surface. Importantly, the table does

[2] For isometric predictors, i.e., predictors with values on the same scale, nonlinear interactions can be modeled with somewhat greater precision with a thin plate regression spline smoother, which is requested with the `s()`. It is important to note that when the `s()` function is applied to non-isometric predictors, completely misleading results are typically obtained.

| A. parametric coefficients | Estimate | Std. Error | t-value | p-value |
|---|---|---|---|---|
| (Intercept) | -0.8919 | 0.0184 | -48.4523 | $< 0.0001$ |
| PhraseInitialTRUE | -0.2342 | 0.0767 | -3.0548 | 0.0023 |
| PhraseFinalTRUE | 0.5695 | 0.0795 | 7.1608 | $< 0.0001$ |
| B. smooth terms | edf | Ref.df | F-value | p-value |
| s(DictDiphoneCount) | 3.9681 | 3.9992 | 807.0226 | $< 0.0001$ |
| s(DictDiphonePosition) | 3.9465 | 3.9980 | 767.0939 | $< 0.0001$ |
| s(LogPhraseLength) | 1.0001 | 1.0002 | 0.0456 | 0.8310 |
| s(LogDuration) | 7.3465 | 8.3297 | 1491.4194 | $< 0.0001$ |
| te(WordActDiversity,PhraseRate):PhraseFinalFALSE | 3.4963 | 3.8133 | 75.7274 | $< 0.0001$ |
| te(WordActDiversity,PhraseRate):PhraseFinalTRUE | 5.1997 | 6.0181 | 101.2842 | $< 0.0001$ |
| s(SemanticTypicality):PhraseFinalFALSE | 3.6784 | 3.9381 | 485.0058 | $< 0.0001$ |
| s(SemanticTypicality):PhraseFinalTRUE | 3.3121 | 3.7477 | 160.0054 | $< 0.0001$ |
| s(DictDiphoneActDiversity):PhraseFinalFALSE | 3.9554 | 3.9985 | 297.9521 | $< 0.0001$ |
| s(DictDiphoneActDiversity):PhraseFinalTRUE | 3.8722 | 3.9880 | 56.4501 | $< 0.0001$ |
| s(CorpusTime) | 34.2773 | 42.5874 | 714.6882 | $< 0.0001$ |

Table 1: Summary table of model `m4` fitted to the log odds of diphone deviation for speaker 40 in the Buckeye corpus of American English as spoken in Columbus, Ohio.

not inform us about whether the interaction itself is significant. In other words, the situation is the exact parallel of a linear model with a two-level factor and a covariate that is specified in R as

```
formula(Response ~ Factor + Factor:Covariate)
```

The summary of this model informs us about whether the two regression lines for the two levels of the factor have slopes that differ significantly from zero, but it does not tell us whether there are significant differences between the slopes. To assess whether there truly is an interaction in a GAM, a possible first step is to plot the difference curve with the `plot_diff` function from the **itsadug** package (van Rij et al., 2017), as shown in Figure 8.

```
library(itsadug)
plot_diff(m4,
          view="SemanticTypicality",
          comp=list(PhraseFinal = c("TRUE", "FALSE")))
```

When this difference curve is added to the effect of `SemanticTypicality` for diphones in words that are not phrase-final (the reference level), the curve is obtained for its effect in phrase-final position. Consistent with the significant main effect for `SemanticTypicality` in Table 1, the 95% confidence interval of the difference curve does not include the horizontal axis: the log odds of diphone deviation is consistently higher for phrase-final words. Given the large effect for greater values of `SemanticTypicality` and the relatively constrained confidence interval, it is clear that the interaction is unlikely to be reducable to just a main effect. We check this by taking model `m4`, replacing the term `s(SemanticTypicality, k=5, by=PhraseFinal)` by the term `s(SemanticTypicality, k=5)`, and comparing the goodness of fit of this new, simpler model (`m5`, model not shown) with that of `m4`, the more complex model, using the `compareML` function from the **itsadug** package.

```
compareML(m4, m5)
```

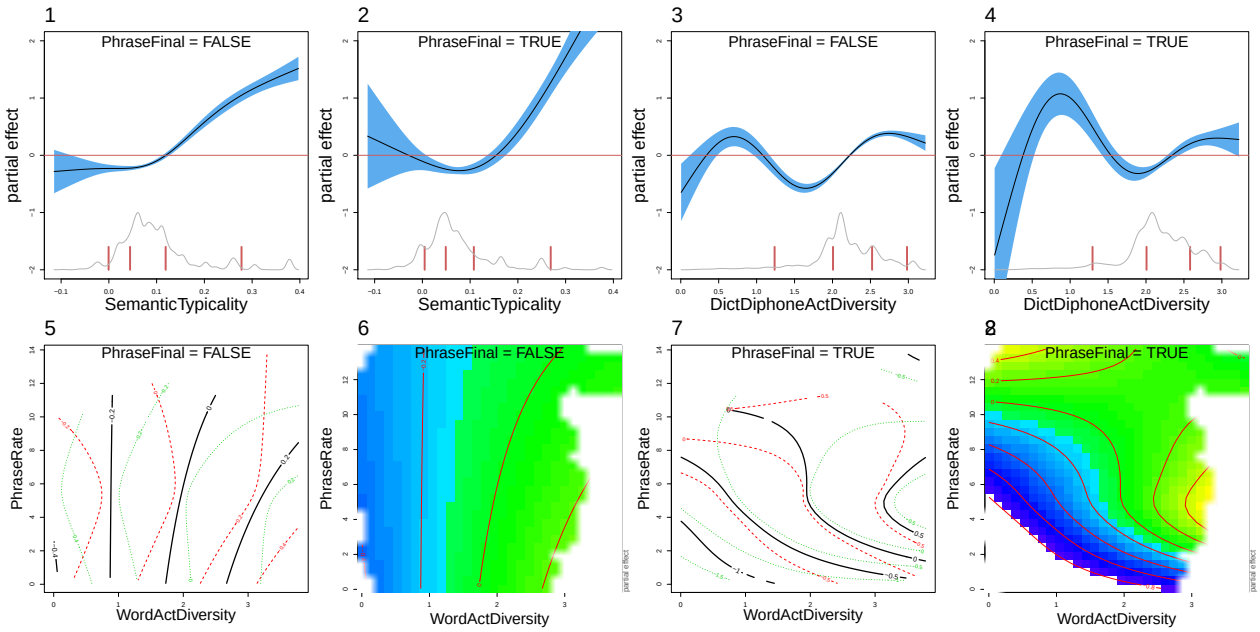



Fig. 7: Partial effects for interactions in a GAM for the log odds of diphone deviation for speaker *S*40 in the Buckeye corpus. In the contour plots, dotted green lines indicate 1 SE up, and dashed red lines 1 SE down from a contour line (panels 5 and 7); in panels 6 and 8, darker shades of blue indicate lower values, and darker shades of yellow higher values.

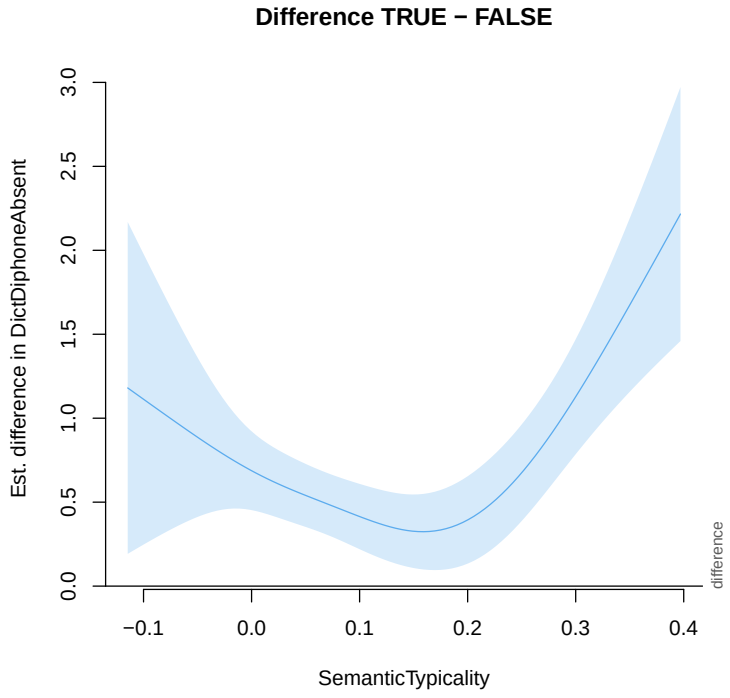



Fig. 8: The difference curve, given model `m4`, for `SemanticTypicality` for words that are and that are not in phrase-final position.

```
  Model    Score Edf Difference    Df   p.value
1    m5 38537.64  29
2    m4 38507.85  31     29.798 2.000 1.145e-13
```

Although `m4` requires 2 more effective degrees of freedom, these additional degrees of freedom enable it to bring the fREML score down by 29.8. The small p-value indicates that the increase in goodness of fit outweights the increased complexity of the model.

### 3.3 Random effects in GAMs

It is straightforward to include random effects in generalized additive models fitted with **mgcv**. By-subject random intercepts are requested with `s(subject, bs="re")` (notation in **lme4**: `(1|subject)`). By-subject random slopes for a covariate are specified as `s(covariate, subject, bs="re")` (**lme4**: `(0+covariate|subject)`). For factors, `s(factor, subject, bs="re")` directs the model to estimate, for each subject, random sum contrasts (**lme4**: `(1|factor:subject)`. Hence, no separate term for by-subject random intercepts should be requested. The variance components of a GAMM and associated confidence intervals are obtained with `gam.vcomp`. Unlike **lme4**, **mgcv** does not offer tools for modeling correlation parameters for random effects.

For corpus data, a random effect factor such as `Word` can cause serious problems for the analyst. Recall that in the present dataset predictors at the word level are repeated in the dataset for each of a word's diphones. One might think that adding by-word random intercepts would alleviate this problem. Technically, we can add the model term `s(Word, bs="re")` to `m4`, resulting in a new model, `m6` (not shown) that appears to provide an improved fit (for instance, AIC is down by 3470.3). However, of the 829 word types, 383 occur once only (46.2%). As a consequence, nearly half of the words have only one occurrence but are predicted by no less than three factorial variables: `PhraseInitial`, `PhraseFinal`, and a random intercept. In addition, there are several covariates that will further be specific to a given word, such as `LogDuration` and `WordActDiversity`. Thus, we have far too many predictors to one observation. As a consequence, model `m6` is severely overspecified.

The adverse effects of this overspecification become apparent when we consider the concurvity of the model. Concurvity is a generalization of co-linearity, and causes similar problems of interpretation, in the sense that when concurvity is high, it is difficult to say which variables are driving the model's predictions. As when co-linearity is present, concurvity can also make estimates somewhat unstable. The `concurvity` function of **mgcv** provides several measures of concurvity, each of which is bounded between zero and one. Values close to or equal to 1 indicate there is a total lack of identifiability. The index we consider here, which Wood describes as in some cases potentially too optimistic, is based on the idea that a smooth can be decomposed into a part `g` shared with other predictors, and a part `f` that is entirely its own unique contribution. The greater part `g` is compared to part `f`, the greater the concurvity. The `observed` index of concurvity is based on the square of the ratio of the Euclidian lengths of vectors `g` and `f` evaluated at the observed values of the predictors.

When we extract this measure from the output of `concurvity(m6)`, we obtain the concurvity values shown in Figure 9 in blue. The same figure shows, in red, the concurvity values of the corresponding terms of model `m4`, the model that is otherwise identical, except that m4 lacks by-word random intercepts. Concurvity values for model `m6` are higher across the board, and are extremely and unacceptably high for `DictDiphoneCount` and for the interactions of `WordActDiversity` and `SemanticTypicality` by `PhraseFinal`.

It is clear that `m6` is an overspecified model that must be simplified. We therefore completely remove `PhraseFinal` and `PhraseInitial` from the model specification, as this will attenuate the

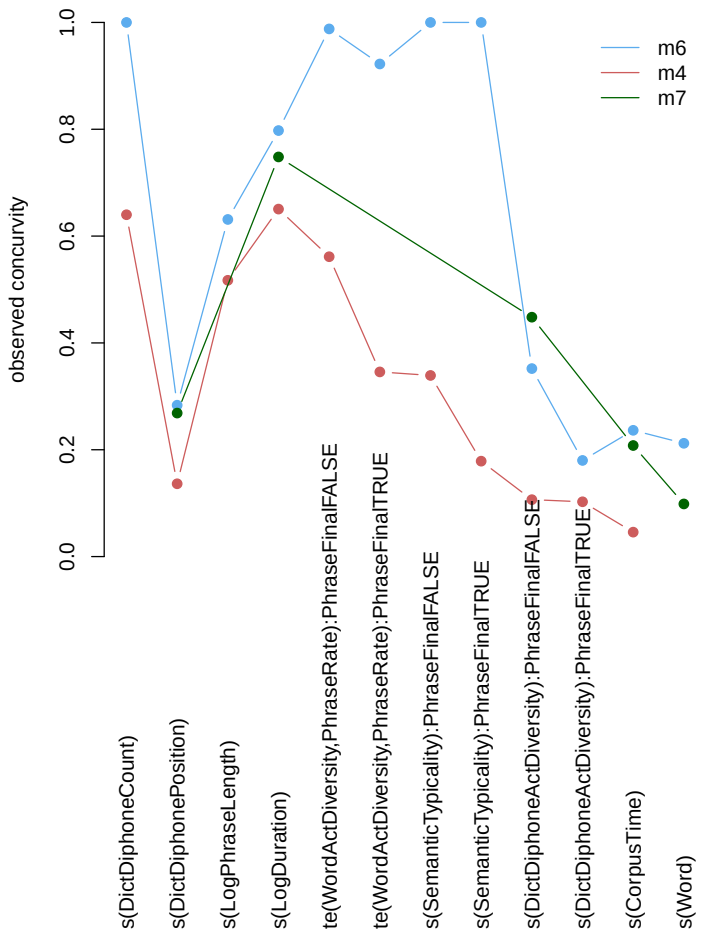



Fig. 9: Observed concurvity for models `m6` (blue), `m4` (red), and `m7` (green). Model `m7` does not have interactions with `PhraseFinal`, its concurvity value for `DictDiphoneActDiversity` is shown for the `PhraseFinal=FALSE` interaction term in the other two models.

adverse consequences of hapax legomena occurring with only one value for these predictors. After further simplification, model `m7`, with good support for all predictors, is obtained,

```
m7 = bam(DictDiphoneAbsent ~ WordActDiversity +
                             s(DictDiphonePosition, k = 5) +
                             s(LogDuration) +
                             s(DictDiphoneActDiversity, k = 5) +
                             s(CorpusTime, k = 100) +
                             s(Word, bs="re"),
         data = buckeye, family="binomial", discrete=T)
```

the concurvity values of which are presented in Figure 9 in green. Concurvity is now much reduced.

Given that linguistic covariates tend to be tightly correlated, and especially so for observational data from corpora that have not been hand-curated to minimize variation in specific dimensions, model `m7` appears to keep concurvity within bounds that are perhaps reasonable. Unfortunately, `m7` confronts us with another problem: the random intercepts it has estimated for `Word` should follow

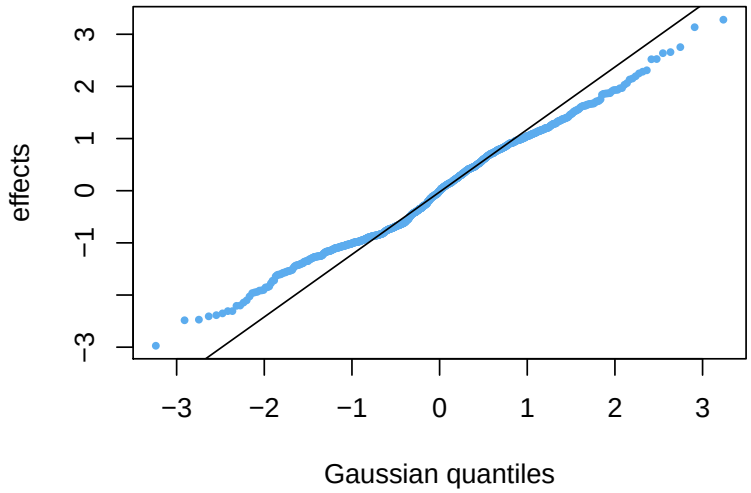



Fig. 10: A quantile-quantile plot of the by-word random intercepts of model `m7` shows marked departure from normality.

a Gaussian distribution, but as shown by Figure 10, they fail to do so. The tails of the observed distribution are too short for it to be Gaussian. That a Gaussian random effect is not really appropriate for the present data is also apparent from the fact that the by-word random intercepts can be predicted from `SemanticTypicality`, `DictDiphoneCount`, and `WordActDiversity` (all coefficients positive, all $p \ll 0.0001$, adjusted R-squared 0.263). In other words, what should be random noise is in fact structured variation. Part of the problem is the Zipfian distribution of words' frequencies: Random-effect factors with a Zipfian frequency distribution are not well suited for modeling with Gaussian random effects (Douglas Bates, p.c.).

As George Box famously said, "all models are wrong, but some are useful" (Box, 1976). Model `m7`, although far from perfect, is perhaps useful in two ways. First, it clarifies that there is substantial variation tied to individual words. Second, it is useful as a strong adversarial attach on the predictors of model `m4` that are tied to the word. The survival of `WordActDiversity` in `m7`, albeit only as a linear effect, is, from this perspective, an index of its robustness. At the same time, model `m4`, although also not perfect, provides much more insight into how words' properties (rather than sublexical properties) may co-determine the log odds of diphone deviation. Because `m4` is a logistic model, there is no assumption that the errors of this model should be Gaussian and that they should be independently and identically distributed. Nevertheless, model `m4` is overly optimistic because the observations from which it is constructed are not independent but, as shown by the analysis of concurvity, cluster by word in linguistically meaningful ways.

One important new kind of random effect that **mgcv** makes available is the factor smooth. In a model with covariates with linear effects, it is possible that regression lines for individual subjects differ both with respect to their slopes and with respect to their intercepts. The nonlinear counterpart of this situation is that subjects have their own wiggly curves. Factor smooths implement such wiggly random effects. For example,

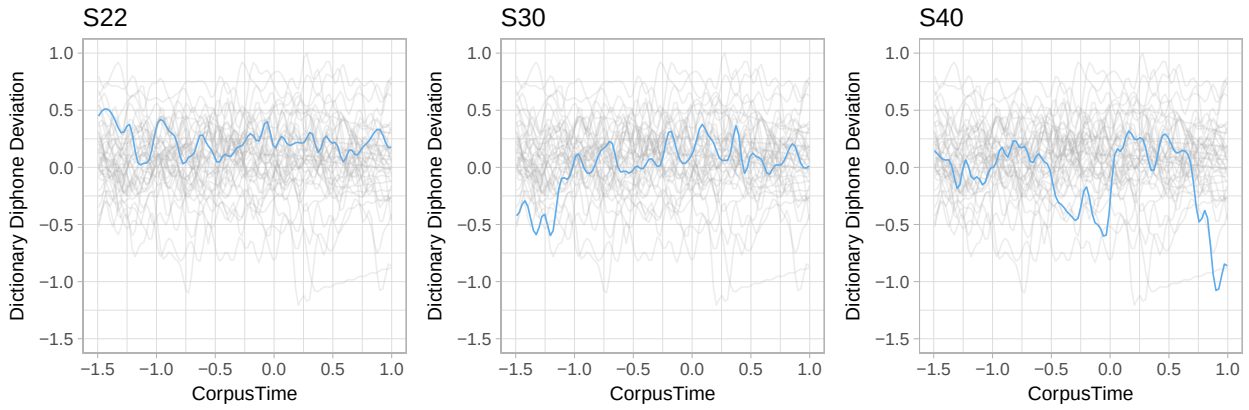



Fig. 11: By-speaker factor smooths for `CorpusTime` in a mixed GAM fitted to the log odds of diphone deviation in the Buckeye corpus, including all speakers. Smooths for selected speakers are highlighted.

```
s(CorpusTime, Speaker, bs = "fs", m = 1)
```

requests wiggly curves for log odds as a function of `CorpusTime` for all speakers in the corpus. Figure 11 illustrates by-speaker factor smooths in the Buckeye corpus. Each line represents a speaker. Some speakers show considerable wiggliness, whereas other speakers show only small local ups and downs. As the curves have their own intercepts, in a model with factor smooths, no separate term for by-speaker random intercepts should be included.

As is the case for random effects in the linear mixed model, the factor smooths are subject to shrinkage. Importantly, factor smooths are set up in such a way that if there is no wiggliness, horizontal straight lines are returned. In this case, the model has become a model with just straightforward random intercepts.

In the linear mixed model, a model with by-subject random intercepts as well as by-subject random slopes will provide population estimates for intercept and slope. In the case of factor smooths, it is possible to request both a general, speaker-independent smooth, together with by-speaker factor smooths.

```
s(CorpusTime) + s(CorpusTime, Speaker, bs = "fs", m = 1)
```

In this case, **mgcv** issues a warning, as in general multiple smooths for the same covariate should be avoided. For this special case, however, this warning can be ignored (Simon Wood, p.c.). For large datasets, it should be kept in mind that estimating factor smooths for large numbers of speakers or words can be computationally very expensive.

It is both an empirical and a theoretical issue whether a separate smooth that is supposed to be common to all speakers, such as `s(CorpusTime)` in the above specification, is really necessary and makes sense. Is it theoretically justified to expect that when speakers go through a one-hour interview, there truly should be a common way in which the diphones they realize in their speech deviate from the standard language? If not, perhaps the main effect for `CorpusTime` should be removed.

It is noteworthy that the interpretation of the individual curves estimated by a factor smooth is different from that of random intercepts and slopes in the linear mixed model. The individual curves provide an estimate of how a given speaker went through her/his interview, but how the same speaker would behave in a replication interview is unlikely to be a variation of the same curve

with greater or smaller amplitude. Instead, the expectation is that the speaker will show a similar amount of wiggliness, but with ups and downs at different moments in time.

Thus, GAMs not only offer the analyst new possibilities for understanding complex relations in large volumes of data, they also confront us with new challenges. For instance, in Figure 11, speaker `S40` shows substantial fluctuations in the log odds of diphone deviation. Such large deviations are unlikely to be due to just chance, and require further explanation and further reflection on the temporal dynamics of language use.

### 3.4 Extensions of GAMs

The toolkit of smoothing splines (see the documentation for `smooth.terms` for an overview of the many different splines that **mgcv** implements) is available for Gaussian models, as well as for Poisson and Binomial responses, using the `family` directive familiar from the generalized linear model (`glm`). GAMs allow for a more complex linear predictor $\eta$, but otherwise the linear predictor is used exactly as in the generalized linear model, as summarized in section 2.1. When the residuals of a Gaussian model follow a t-distribution rather than a normal distribution, the `family` directive can be set to `scat`, which requests a scaled t model for the residuals. For multinomial logit models, the `family` directive is set to `multinom`, and for the modeling of ordinal response variables, `family` is set to `ocat`. The documentation for `scat`, `multinom`, and `ocat` provides further detail on these extensions of the generalized additive model and their use.

### 3.5 Reporting GAMM models

The reportage of a generalized additive (mixed) model will generally include a summary table such as Table 1. The summary tables generated by the **mgcv** package report both thin plate regression spline smooths as well as random effects with the `s()` notation. As this can cause confusion for the reader, it is advisable to edit the names of the smooth terms in the model, for instance by renaming a random effect term `s(Word)` as `by-word random intercepts`, and a factor smooth `s(Subject, Trial)` as `by-subject factor smooth for Trial`. Furthermore, it is important to present graphs for the non-linear effects in the model, as the functional form of these effects cannot be deduced from the effective degrees of freedom.

## 4 Key Readings

Finally, as model interpretation and model criticism with GAMMs require a high level of understanding of both the method and the theoretical concepts it builds on, it is advisable to engage in a deeper exploration of the issues at hand prior to applying the models in a productive research environment. Here, we provide an overview of key readings, that the analyst may find useful for exploring various types of data and uncovering and adressing effects commonly found in human response data.

First, Wood (2017), a standard reference on GAMs, provides a necessary background in linear models linear mixed models and generalized linear models and introduction to the theory and applications of GAMs, complemented by a wide range of exercises.

In addition to that, we suggest the following articles, which go in depth into the possibilities offered by GAMMs for dealing with various types of language data and uncovering and handling autocorrelation arising from experiment structure. Baayen et al. (2017b) show techniques offered by GAMMs on the analysis of response times in a word naming task, investigation of a pitch contour task in a word naming experiment and a model fitted to the EEG response amplitude to visually

presented compound words. Baayen et al. (2017c) illustrate on three data sets how human factors like learning or fatigue may interact with predictors of interest, both factorial and metric, and demonstrate why fitting maximally complex models is not an advisable strategy, especially within the framework of the generalized additive mixed effects model. Wieling (2018) offers a hands-on tutorial, including the original data and all R commands, for analysing dynamic time series data on the example of articulator trajectories observed using electromagnetic articulography. The paper leads the reader through the steps of data exploration, visualization, modeling of complex interactions and model criticism, introducing a wide variety of techniques and strategies with a detailed and comprehensive rationale for the modeling decisions, offering the reader an opportunity to replicate the analyses and gain more understanding about the material. van Rij et al. (2019) is a tutorial introduction to GAMMs for pupilometry data, illustrating several methods from the *itsadug* package. Additionally, the extended online documentation of the *itsadug* package provides practical examples to guide visual inspection of GAMM models (`https://cran.r-project.org/web/packages/itsadug/vignettes/inspect.html`), checking for autocorrelation and dealing with it (`https://cran.r-project.org/web/packages/itsadug/vignettes/acf.html`) and significance testing (`https://cran.r-project.org/web/packages/itsadug/vignettes/test.html`).

## References


Baayen, R. H. and Divjak, D. (2017). Ordinal GAMMs: a new window on human ratings. In Makarova, A., Dickey, S. M., and Divjak, D. S., editors, *Thoughts on Language: Studies in Cognitive Linguistics in Honor of Laura A. Janda.*, pages 39–56. Slavica Publishers, Bloomington, IN.

Baayen, R. H., Tomaschek, F., Gahl, S., and Ramscar, M. (2017a). The Ecclesiastes principle in language change. In Hundt, M., Mollin, S., and Pfenninger, S., editors, *The changing English language: Psycholinguistic perspectives*, page in press. Cambridge University Press, Cambridge, UK.

Baayen, R. H., van Rij, J., de Cat, C., and Wood, S. N. (2017b). Autocorrelated errors in experimental data in the language sciences: Some solutions offered by generalized additive mixed models. In Speelman, D., Heylen, K., and Geeraerts, D., editors, *Mixed Effects Regression Models in Linguistics*, page to appear. Springer, Berlin.

Baayen, R. H., Vasishth, S., Bates, D., and Kliegl, R. (2017c). The cave of shadows. Addressing the human factor with generalized additive mixed models. *Journal of Memory and Language*, 94:206–234.

Box, G. E. P. (1976). Science and statistics. *Journal of the American Statistical Association*, 71:791–799.

Davies, M. (2010). The Corpus of Historical American English (COHA): 400+ million words, 1810–2009.

Divjak, D. (2016). The role of lexical frequency in the acceptability of syntactic variants: Evidence from that-clauses in Polish. *Cognitive Science*, DOI:10.1111/cogs.12335:1–29.

Divjak, D., Milin, P., and Baayen, R. H. (2017). A learning perspective on individual differences in skilled reading: Exploring and exploiting orthographic and semantic discrimination cues. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 43(11):1730–1751.

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

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall, London.

Johnson, K. (2004). Massive reduction in conversational American English. In *Spontaneous speech: data and analysis. Proceedings of the 1st session of the 10th international symposium*, pages 29–54, Tokyo, Japan. The National International Institute for Japanese Language.

Linke, M., Broeker, F., Ramscar, M., and Baayen, R. H. (2017). Are baboons learning "orthographic" representations? probably not. *PLOS-ONE*, 12(8):e0183876.

Marra, G. and Wood, S. N. (2012). Coverage properties of confidence intervals for generalized additive model components. *Scandinavian Journal of Statistics*, 39:53–74.

Milin, P., Feldman, L. B., Ramscar, M., Hendrix, P., and Baayen, R. H. (2017). Discrimination in lexical decision. *PLOS-one*, 12(2):e0171935.

Nychka, D. (1988). Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83:1134–1143.

Pitt, M., Johnson, K., Hume, E., Kiesling, S., and Raymond, W. (2005). The Buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Tomaschek, F., Tucker, B., and Baayen, R. H. (2018). Practice makes perfect: The consequences of lexical proficiency for articulation. *Linguistic Vanguard*, page to appear.

Tucker, B. V., Sims, M., and Baayen, R. H. (2018). Opposing forces on acoustic duration. *Manuscript, University of Alberta and University of Tübingen*.

van Rij, J., Hendriks, P., van Rijn, H., Baayen, R. H., and Wood, S. N. (2019). Analyzing the time course of pupillometric data. *Trends in hearing*, 23:2331216519832483.

van Rij, J., Wieling, M., Baayen, R. H., and van Rijn, H. (2017). itsadug: Interpreting time series and autocorrelated data using GAMMs. R package version 2.3.

Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed modeling: A tutorial focusing on articulatory differences between L1 and L2 speakers of English. *Journal of Phonetics*, 70:86–116.

Wieling, M., Montemagni, S., Nerbonne, J., and Baayen, R. H. (2014). Lexical differences between Tuscan dialects and standard Italian: Accounting for geographic and socio-demographic variation using generalized additive mixed modeling. *Language*, 90(3):669–692.

Wieling, M., Nerbonne, J., and Baayen, R. H. (2011). Quantitative social dialectology: Explaining linguistic variation geographically and socially. *PLoS ONE*, 6(9):e23613.

Wieling, M., Tomaschek, F., Arnold, D., Tiede, M., Bröker, F., Thiele, S., Wood, S. N., and Baayen, R. H. (2016). Investigating dialectal differences using articulography. *Journal of Phonetics*, 59:122–143.

Wood, S. N. (2006). *Generalized Additive Models*. Chapman & Hall/CRC, New York.

Wood, S. N. (2013a). On p-values for smooth components of an extended generalized additive model. *Biometrika*, 100:221–228.

Wood, S. N. (2013b). A simple test for random effects in regression models. *Biometrika*, 100:1005–1010.

Wood, S. N. (2017). *Generalized Additive Models*. Chapman & Hall/CRC, New York.