Supplementary Material for Nonlinearities in Bilingual Visual Word Recognition

Koji Miwa and Harald Baayen

12/08/2020

Table of Contents

1. Set-up

Load necessary packages.

```
library(languageR, quietly = T) # for the LMM
library(lme4, quietly = T) # for the LMM
library(lmerTest, quietly = T) # for the LMM
library(LMERConvenienceFunctions) # for the LMM
library(mgcv, quietly = T) # for the GAMM
library(itsadug, quietly = T) # for the GAMM
library(qgam, quietly = T) # for the QGAM/QGAMM
```

Throughout this tutorial, familiarity with the LMM and basic knowledge in R are assumed (see Pinheiro & Bates, 2000 and Baayen, Davidson, & Bates, 2008 for introduction of the LMM with R).

In what follows, we make use of treatment coding for factors, as this makes the interpretation of interactions of factors and numerical predictors in GAMMs much more straightforwardly interpretable.

2. Cross-language orthographic similarity effect

We first demonstrate how the GAMM can be applied in bilingual processing research, reanalyzing lexical decision data from Dijkstra et al. (2010). The study tested 21 Dutch-English bilinguals reading 194 English words and 194 nonwords in a lexical decision

experiment. We reanalyzed data for 189 target words for which subtitle *Frequency* data were available. The authors reported a substantial facilitation for identical cognates on top of a facilitatory effect of standardized orthographic similarity (hereafer *OS*).

```
load("Data/dijkstra.rda")
```

We start with the LMM that many psycholinguists are familiar with. In the following LMM model, we replicate Dijkstra et al's (2010) result with a rated orthographic similarity between target English words and their Dutch translation equivalent (*OS*) and a factor (*Ident*) with the two levels of *Identical* and *NonIdentical*. We also include subtitle frequency as a covariate (*Frequency*).

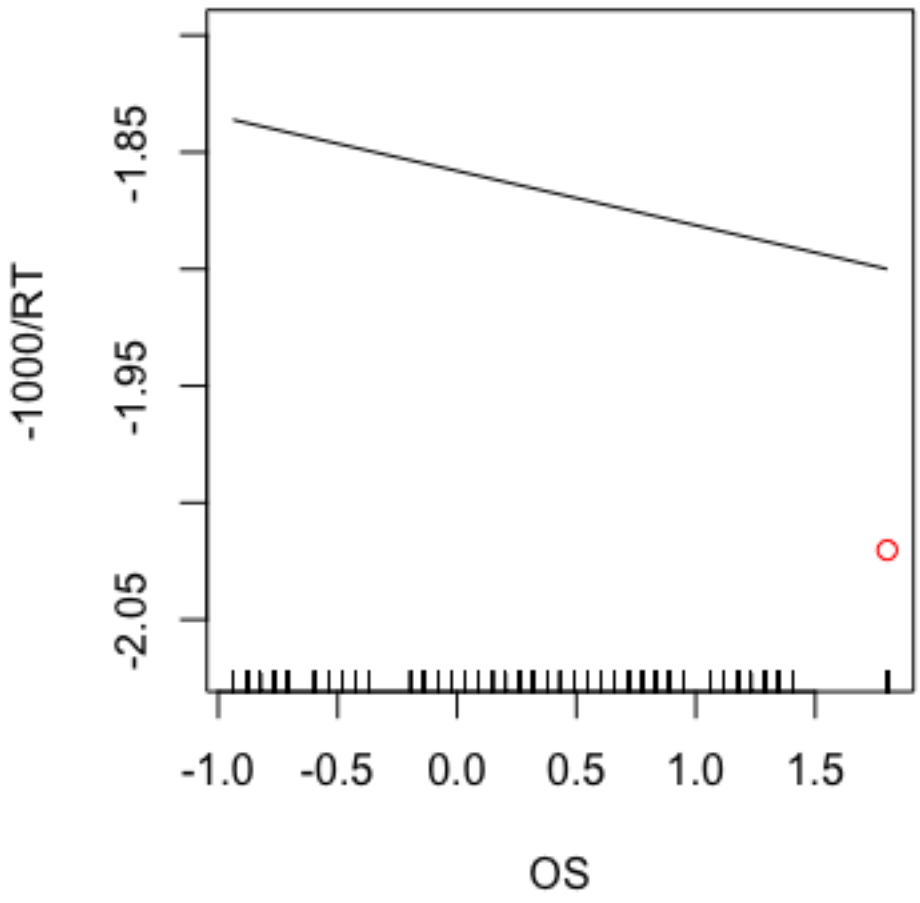
```
dijkstra$Ident = relevel(dijkstra$Ident, ref = "NonIdentical")
dijkstra.lmer = lmer(invRT ~
                       OS +
                       Ident +
                       Frequency +
                       (1 | Participant) +
                       (0 + Trial | Participant) +
                       (1 | Word) ,
                     data = dijkstra)
summary(dijkstra.lmer, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## invRT ~ OS + Ident + Frequency + (1 | Participant) + (0 + Trial |
##     Participant) + (1 | Word)
##    Data: dijkstra
##
## REML criterion at convergence: 2140.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8078 -0.6472 -0.0731  0.5623  4.2783
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Word          (Intercept) 0.010017 0.10009
##  Participant   Trial       0.001581 0.03976
##  Participant.1 (Intercept) 0.064322 0.25362
##  Residual                  0.097379 0.31206
## Number of obs: 3535, groups:  Word, 178; Participant, 21
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)     -1.858546   0.056281  20.985138 -33.023  < 2e-16 ***
## OS              -0.023302   0.011993 169.302742  -1.943  0.05369 .
## IdentIdentical  -0.119715   0.037983 166.176468  -3.152  0.00193 **
```

```
## Frequency       -0.076796   0.009192 168.956303  -8.355 2.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Identical cognates revealed substantial facilitation on top of a facilitatory effect of *OS* (indicated by the red circle).

```
x = sort(unique(dijkstra$OS))
y = -1.858546 +  (-0.023302*x) -0.119715
plotLMER.fnc(dijkstra.lmer, pred="OS" ,ylim =c(-2.07, -1.8), xlabel = "OS",
ylabel = "-1000/RT", main =
"");rug(dijkstra$OS);points(x[length(x)],y[length(y)], col="red")

## effect size (range) for  OS is  0.06383857
```



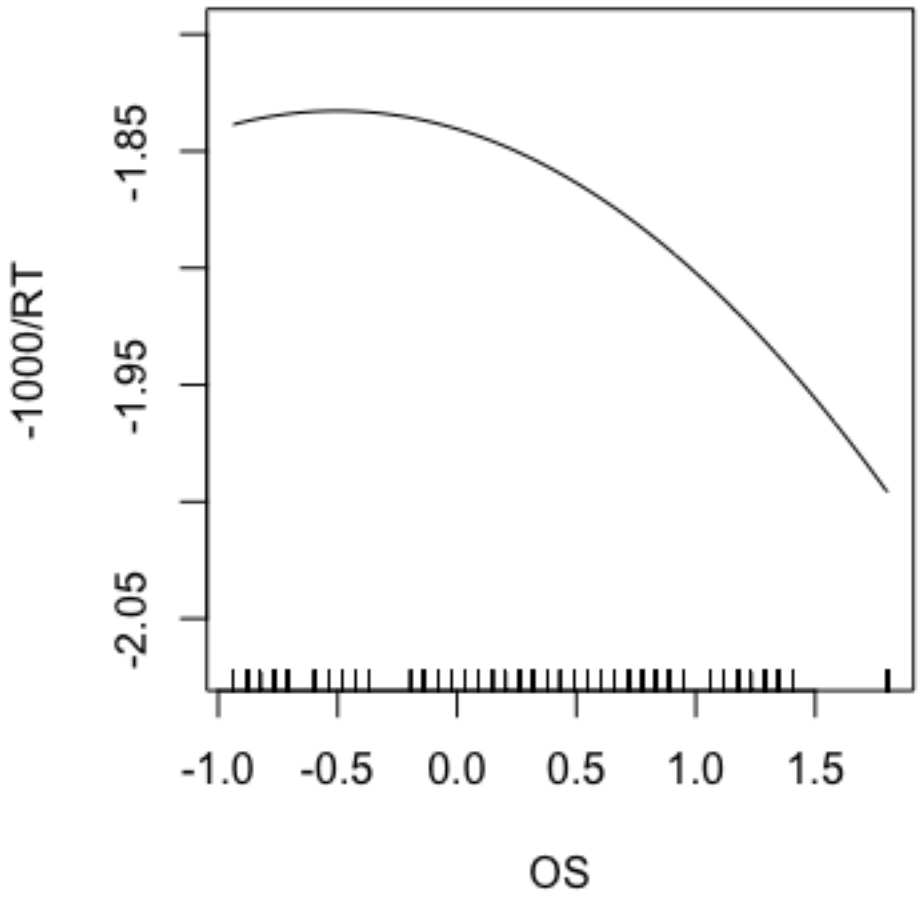
Within the context of the LMM, the effect of *OS* can be modeled as nonlinear by regressing the response on powers of *OS* (e.g., OS squared, cubed etc.). Here, we fit a model with a second-degree polynomial.

```
dijkstra.lmer.poly2 = lmer(invRT ~
                             poly(OS, 2, raw = T) +
                             # This is the same as OS + I(OS^2)
                             Frequency +
                             (1 | Participant) +
                             (0 + Trial | Participant) +
                             (1 | Word) ,
                             data = dijkstra)
summary(dijkstra.lmer.poly2, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: invRT ~ poly(OS, 2, raw = T) + Frequency + (1 | Participant) +
##     (0 + Trial | Participant) + (1 | Word)
##    Data: dijkstra
##
## REML criterion at convergence: 2147.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8062 -0.6419 -0.0721  0.5649  4.2484
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Word          (Intercept) 0.010424 0.10210
##  Participant   Trial       0.001591 0.03988
##  Participant.1 (Intercept) 0.064314 0.25360
##  Residual                  0.097373 0.31205
## Number of obs: 3535, groups:  Word, 178; Participant, 21
##
## Fixed effects:
##                            Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)               -1.841059   0.057809  23.346336 -31.847  < 2e-16
## poly(OS, 2, raw = T)1     -0.030847   0.011880 168.886107  -2.597   0.0102
## poly(OS, 2, raw = T)2     -0.030733   0.013471 166.399554  -2.281   0.0238
## Frequency                 -0.078955   0.009305 169.310729  -8.485 1.04e-14
##
## (Intercept)              ***
## poly(OS, 2, raw = T)1    *
## poly(OS, 2, raw = T)2    *
## Frequency                ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plotLMER.fnc(dijkstra.lmer.poly2, pred="poly(OS, 2, raw = T)" ,ylim =c(-2.07,
-1.8), xlabel = "OS", ylabel = "-1000/RT", main = ""); rug(dijkstra$OS)
```

```
## effect size (range) for  poly(OS, 2, raw = T) is  0.1630846
```



The polynomial regression line captured the overall negative accelerating trend. However, the model lacks precision when pitted against the model with a strictly linear effect of OS in combination with the factor *Ident*; the substantial facilitation for *Ident* is no longer visible. When the factor *Ident* is included in the model, the polynomial regression line is no longer supported.

```
dijkstra.lmer.poly3 = lmer(invRT  ~
                             poly(OS, 2, raw = T) +
                             Frequency +
                             Ident +
                             (1|Participant)+
                             (0+Trial|Participant)+
                             (1|Word),
```

```
                          data = dijkstra)
summary(dijkstra.lmer.poly3, corr=F)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## invRT ~ poly(OS, 2, raw = T) + Frequency + Ident + (1 | Participant) +
##     (0 + Trial | Participant) + (1 | Word)
##    Data: dijkstra
##
## REML criterion at convergence: 2146.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8111 -0.6460 -0.0726  0.5606  4.2785
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Word          (Intercept) 0.010078 0.10039
##  Participant   Trial       0.001582 0.03977
##  Participant.1 (Intercept) 0.064323 0.25362
##  Residual                  0.097379 0.31206
## Number of obs: 3535, groups:  Word, 178; Participant, 21
##
## Fixed effects:
##                          Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)            -1.868521   0.059078  25.424613 -31.628  < 2e-16
## poly(OS, 2, raw = T)1  -0.024195   0.012123 168.028962  -1.996   0.0476
## poly(OS, 2, raw = T)2   0.013339   0.023989 167.906048   0.556   0.5789
## Frequency              -0.076142   0.009286 168.045096  -8.200 5.93e-14
## IdentIdentical         -0.151425   0.068558 167.996356  -2.209   0.0285
##
## (Intercept)           ***
## poly(OS, 2, raw = T)1 *
## poly(OS, 2, raw = T)2
## Frequency             ***
## IdentIdentical        *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(dijkstra.lmer.poly2, dijkstra.lmer.poly3)

## refitting model(s) with ML (instead of REML)

## Data: dijkstra
## Models:
## dijkstra.lmer.poly2: invRT ~ poly(OS, 2, raw = T) + Frequency + (1 |
Participant) +
## dijkstra.lmer.poly2:     (0 + Trial | Participant) + (1 | Word)
## dijkstra.lmer.poly3: invRT ~ poly(OS, 2, raw = T) + Frequency + Ident + (1
| Participant) +
```

```
## dijkstra.lmer.poly3:     (0 + Trial | Participant) + (1 | Word)
##                      Df    AIC    BIC  logLik deviance  Chisq Chi Df
## dijkstra.lmer.poly2  8 2137.8 2187.1 -1060.9   2121.8
## dijkstra.lmer.poly3  9 2134.8 2190.4 -1058.4   2116.8 4.9242      1
##                      Pr(>Chisq)
## dijkstra.lmer.poly2
## dijkstra.lmer.poly3    0.02648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The generalized additive mixed model (GAMM) offers a toolkit for building a more precise statistical model for the present dataset. GAMMs can be used for both exploratory data analysis and for confirmatory data analysis. In the case of confirmatory data analysis, the researcher will have in mind a specific hypothesis about the functional form of a main effect or interaction. This hypothesis is then formalized in the form of a GAMM model, and once the data have been collected, a fit of this specific model to the data will straightforwardly reveal, first, whether a non-linear effect is indeed present, and second, whether the effect has the predicted functional form. Apart from potential further refinements to the model as dictated by model criticism, no further model fitting will necessary nor allowed. When the researcher has no clear theoretically motivated hypotheses about the shape of nonlinear effects, the analysis is necessarily exploratory in nature. In this case, incremental model building will typically afford the researcher more insight into the structure of a dataset than a model with the most complex structure that the data can tolerate. When adding in, step by step, more nonlinearities to the model, it is advisable to set alpha levels sufficiently low, for instance by using a Bonferroni correction for the number of models fitted, to safeguard against overvaluing effects.

In what follows, we make use of the `mgcv` package (Wood, 2017). In the following model formula, smooth terms for a numerical predictor are included in the model by the `s()` function, using the thin plate regression spline. `s(Word, bs = "re")` indicates random intercepts for *Word*. `s(Trial, Participant, bs="fs", m=1)` indicates factor smooths (the non-linear equivalent of random intercepts and random slopes), with the intercepts also estimated for *Participant*.

The mathematics underlying the penalization of basis functions and the mathematics underlying random effects are very similar in GAMMs, hence it makes sense mathematically that the same `s()` directive is used. There is an extensive technical literature on how to set up the basis functions for splines. In fact, there are many different kinds of splines, and how exactly the default splines are set up in MGCV requires at least one full page of mathematics. The reader is referred to Wood (2017) for details. Visual examples of how smooths are constructed out of basis functions are given in Baayen et al. (2017).

Factor smooths implement shrinkage for wiggly curves, just as the LMM implements shrinkage for random intercepts and random slopes. Shrinkage is a statistical technique that ensures that model parameters for individual subjects (or items) are more conservative than would be the case if models were fit to subjects (or items) individually. In this way, the researcher is protected against overfitting, and predictions will be more

precise for future observations on the same subjects (or items). The LMM implements shrinkage for random intercepts and random slopes, which protects the model against overfitting (see for detailed discussion Baayen, 2008). GAMMs likewise implement shrinkage for random-effect factors. Within the context of the GAMM, it is possible to set up the nonlinear equivalent of random intercepts and random slopes by means of special splines known as factor smooths. For factor smooths, shrinkage ensures that if the response does not covary with a given predictor, a factor smooth for that predictor will reduce to random intercepts.

```
dijkstra.gam = bam(invRT  ~
                     s(OS) +
                     s(Frequency) +
                     s(Trial, Participant, bs="fs", m=1) +
                     s(Word, bs="re"),
                   data = dijkstra, discrete = TRUE)
summary(dijkstra.gam)
```

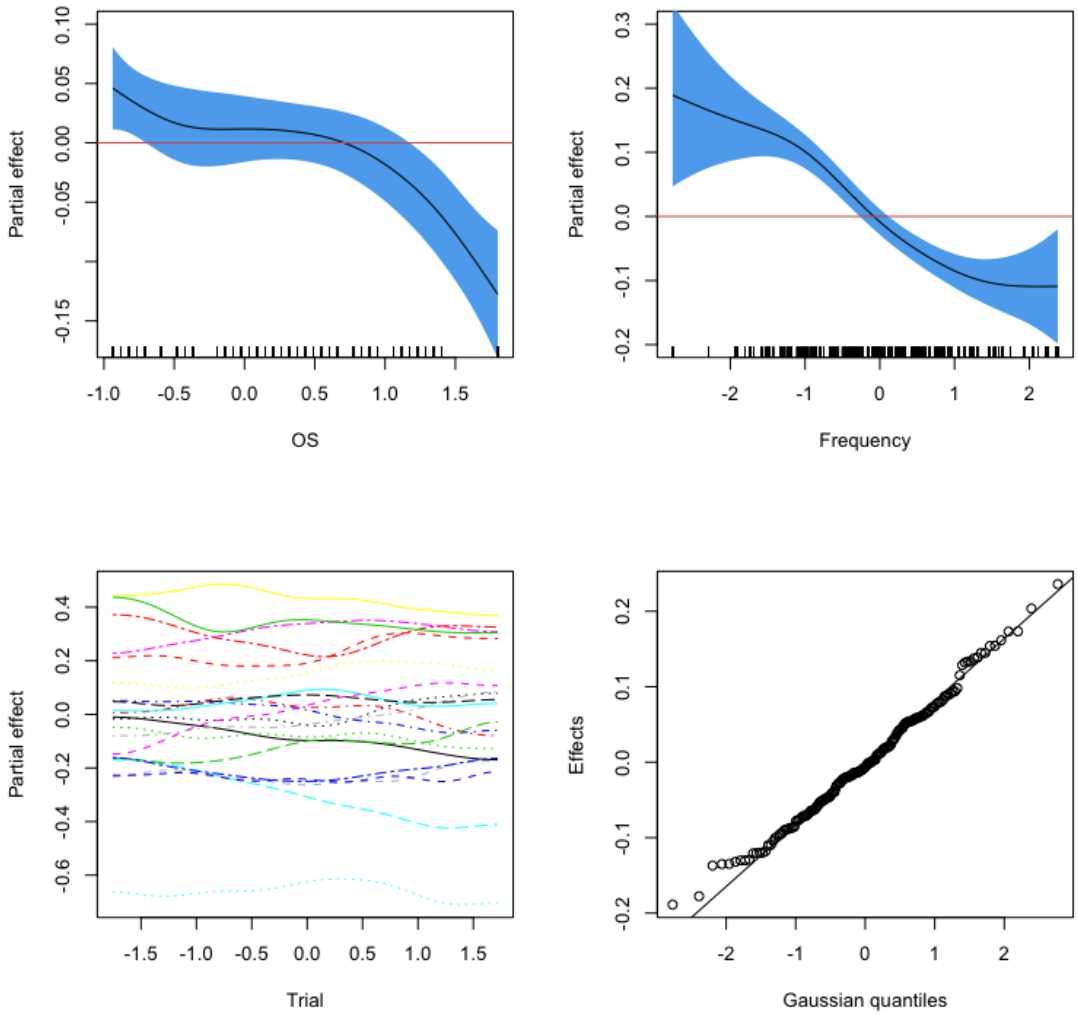
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(OS) + s(Frequency) + s(Trial, Participant, bs = "fs",
##     m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.87789    0.05761   -32.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F  p-value
## s(OS)                   2.745   2.940 11.629 1.13e-07 ***
## s(Frequency)            3.166   3.442 23.237 5.81e-16 ***
## s(Trial,Participant)  65.673 189.000 12.759  < 2e-16 ***
## s(Word)              113.601 175.000  2.186 6.11e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.454   Deviance explained = 48.2%
## fREML = 1063.8  Scale est. = 0.095747  n = 3535
```

Unlike the LMM, a GAMM summary has two parts: a parametric part for linear terms and a nonparametric part for smooth terms. The smooth terms comprise both splines (as for Frequency), factor smooths (as for the trial by participant interaction), and random intercepts (as for Word). An F-test (detailed in Wood, 2013) is reported that clarifies whether a smooth term provides a noteworthy contribution to the model fit. Comparison of

models with and without a given smooth term typically leads to the same conclusions as this F-test.

We visualize the model output. The top left panel visualizes the effect of *OS*. With the thin plate regression spline for *OS*, the GAMM is flexible enough to capture the greater facilitation for identical cognates (the high edf value indicates greater wiggliness). The evaluation of GAMMs relies more on visualization than LMMs do. Where the confidence interval (which is empirical Bayes rather than frequentist) does not include zero, indicated by the red line, the effect is significant. The effect shown is the partial effect of the predictor, i.e., the contribution of the predictor to the fitted values, independently of the contributions of the other predictors in the model. The argument "residuals=F" suppresses the visualization of raw data. The top right panel visualizes the effect of word frequency, which is strong in the middle range of log frequency, and tapers off at both tails of the distribution. The lower left panel presents the by-participant random curves for *Trial*, showing considerable variability between participants, with some showing stable behavior, with other showing nearly linear trends up or down (possibly indicators of fatigue or practice effects), and some showing undulating patterns suggestive of fluctuations in attention. The lower right panel presents a quantile-quantile plot for the model residuals, which roughly follow a normal distribution, as required. For a technical explanation of the summaries of the model, see Wood (2012).

```
par(mfrow=c(2,2))
plot(dijkstra.gam, select=1, shade.col="steelblue2", scheme=1, ylab="Partial
effect",
     ylim = c(-0.17, 0.1), residuals = F); abline(h=0, col="indianred")
plot(dijkstra.gam, select=2, shade.col="steelblue2", scheme=1, ylab="Partial
effect",
     ylim = c(-0.2, 0.3), residuals = F); abline(h=0, col="indianred")
plot(dijkstra.gam, select=3, main=" ", ylab="Partial effect")
plot(dijkstra.gam, select=4, main=" ", ylab="Effects")
```



We then include by-participant random slopes for *Frequency* and *OS* to see whether the model improves. This is equivalent to the specification in lme4: (0+Frequency|Participant). What happens in mgcv is that a ridge penalty is put on the by-subject random slopes for frequency. For straightforward datasets with linear predictors, this leads to virtually the same estimates as given by lme4.

```
dijkstra.gam2 = bam(invRT ~
                      s(OS) +
                      s(Frequency) +
                      s(Participant, Trial, bs = "fs", m=1) +
                      s(Participant, Frequency, bs = "re") +
                      s(Participant, OS, bs = "re") +
                      s(Word, bs = "re") ,
                    data = dijkstra, discrete=TRUE)
summary(dijkstra.gam2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(OS) + s(Frequency) + s(Participant, Trial, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Participant,
##     OS, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.87791    0.05772  -32.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf  Ref.df      F  p-value
## s(OS)                      2.724   2.913  9.580 2.23e-06 ***
## s(Frequency)               3.169   3.440 16.223 3.89e-11 ***
## s(Participant,Trial)      65.474 189.000 12.756  < 2e-16 ***
## s(Frequency,Participant)  12.143  20.000  1.550 0.000201 ***
## s(OS,Participant)          8.239  21.000  0.675 0.028204 *
## s(Word)                  114.701 178.000  2.097 0.190255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.461   Deviance explained = 49.2%
## fREML = 1056.2  Scale est. = 0.094453  n = 3535

compareML(dijkstra.gam, dijkstra.gam2)

## dijkstra.gam: invRT ~ s(OS) + s(Frequency) + s(Trial, Participant, bs =
"fs",
##     m = 1) + s(Word, bs = "re")
##
## dijkstra.gam2: invRT ~ s(OS) + s(Frequency) + s(Participant, Trial, bs =
"fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Participant,
##     OS, bs = "re") + s(Word, bs = "re")
##
## Chi-square test of fREML scores
## -----
##           Model    Score Edf Difference    Df   p.value Sig.
## 1  dijkstra.gam 1063.783   8
## 2 dijkstra.gam2 1056.208  10      7.576 2.000 5.128e-04  ***
##
## AIC difference: 24.20, model dijkstra.gam2 has lower AIC.
```

With the by-participant random slopes for *Frequency* and *OS*, the model improved (fREML score decreases from 1063 to 1056). We also test whether the factor *Ident* will improve the model.

```
dijkstra.gam3 = bam(invRT ~
                      Ident +
                      s(OS) +
                      s(Frequency) +
                      s(Participant, Trial, bs = "fs", m=1) +
                      s(Participant, Frequency, bs = "re") +
                      s(Participant, OS, bs = "re") +
                      s(Word, bs = "re"),
                    data = dijkstra, discrete=TRUE)
summary(dijkstra.gam3)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ Ident + s(OS) + s(Frequency) + s(Participant, Trial,
##     bs = "fs", m = 1) + s(Participant, Frequency, bs = "re") +
##     s(Participant, OS, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.86877    0.05758 -32.458  < 2e-16 ***
## IdentIdentical -0.11287    0.03807  -2.965  0.00305 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf Ref.df      F  p-value
## s(OS)                      1.000   1.00  4.405 0.035891 *
## s(Frequency)               3.137   3.41 15.641 1.08e-10 ***
## s(Participant,Trial)      65.379 189.00 12.737  < 2e-16 ***
## s(Frequency,Participant)  12.150  20.00  1.553 0.000193 ***
## s(OS,Participant)          8.233  21.00  0.674 0.025210 *
## s(Word)                  114.807 178.00  2.446 0.057992 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.461   Deviance explained = 49.2%
## fREML = 1055.6  Scale est. = 0.09446   n = 3535
```

It is noteworthy that once the factor *Ident* is taken into account, the effect of *OS* becomes linear (i.e., the edf is 1), with a substantially increased p-value. Because we are assessing the additive contribution of a fixed-effect, we set the method to "ML." The inclusion of *Ident* reduces the ML score.

```
dijkstra.gam2ml = bam(invRT ~
                        s(OS) +
                        s(Frequency) +
                        s(Participant, Trial, bs = "fs", m=1) +
                        s(Participant, Frequency, bs = "re") +
                        s(Participant, OS, bs = "re") +
                        s(Word, bs = "re") ,
                      data = dijkstra,  method = "ML")
dijkstra.gam3ml = bam(invRT ~
                        Ident +
                        s(OS) +
                        s(Frequency) +
                        s(Participant, Trial, bs = "fs", m=1) +
                        s(Participant, Frequency, bs = "re") +
                        s(Participant, OS, bs = "re") +
                        s(Word, bs = "re"),
                      data = dijkstra,  method = "ML")
compareML(dijkstra.gam2ml, dijkstra.gam3ml)
```

```
## dijkstra.gam2ml: invRT ~ s(OS) + s(Frequency) + s(Participant, Trial, bs =
"fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Participant,
##     OS, bs = "re") + s(Word, bs = "re")
##
## dijkstra.gam3ml: invRT ~ Ident + s(OS) + s(Frequency) + s(Participant,
Trial,
##     bs = "fs", m = 1) + s(Participant, Frequency, bs = "re") +
##     s(Participant, OS, bs = "re") + s(Word, bs = "re")
##
## Chi-square test of ML scores
## -----
##             Model    Score Edf Difference    Df p.value Sig.
## 1 dijkstra.gam2ml 1048.918  10
## 2 dijkstra.gam3ml 1044.504  11      4.414 1.000   0.003  **
##
## AIC difference: 2.44, model dijkstra.gam3ml has lower AIC.

## Warning in compareML(dijkstra.gam2ml, dijkstra.gam3ml): Only small
difference in ML...
```

In our experience with GAMMs, the amount of wiggliness for a given smooth term does not change much depending on the presence or absence of other predictors. This is shown by the following examples.

```
# Original model
dijkstra.gam = bam(invRT  ~
                     s(OS) +
                     s(Frequency) +
                     s(Trial, Participant, bs="fs", m=1) +
                     s(Word, bs="re"),
```

```
                  data = dijkstra, discrete = TRUE)
summary(dijkstra.gam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(OS) + s(Frequency) + s(Trial, Participant, bs = "fs",
##     m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.87789    0.05761   -32.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F  p-value
## s(OS)                   2.745   2.940 11.629 1.13e-07 ***
## s(Frequency)            3.166   3.442 23.237 5.81e-16 ***
## s(Trial,Participant)  65.673 189.000 12.759  < 2e-16 ***
## s(Word)              113.601 175.000  2.186 6.11e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.454   Deviance explained = 48.2%
## fREML = 1063.8  Scale est. = 0.095747  n = 3535

# Model with a covariate removed
dijkstra.gam4 = bam(invRT  ~
                      s(OS) +
                      #s(Frequency) +
                      s(Trial, Participant, bs="fs", m=1) +
                      s(Word, bs="re"),
                    data = dijkstra, discrete = TRUE)
summary(dijkstra.gam4)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(OS) + s(Trial, Participant, bs = "fs", m = 1) + s(Word,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.86963    0.05828  -32.08   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf  Ref.df      F  p-value
## s(OS)                  2.214   2.323  7.719 0.000143 ***
## s(Trial,Participant)  70.953 189.000 13.187  < 2e-16 ***
## s(Word)              142.523 187.000  5.149  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.455   Deviance explained = 48.6%
## fREML = 1172.5  Scale est. = 0.096576  n = 3747
```

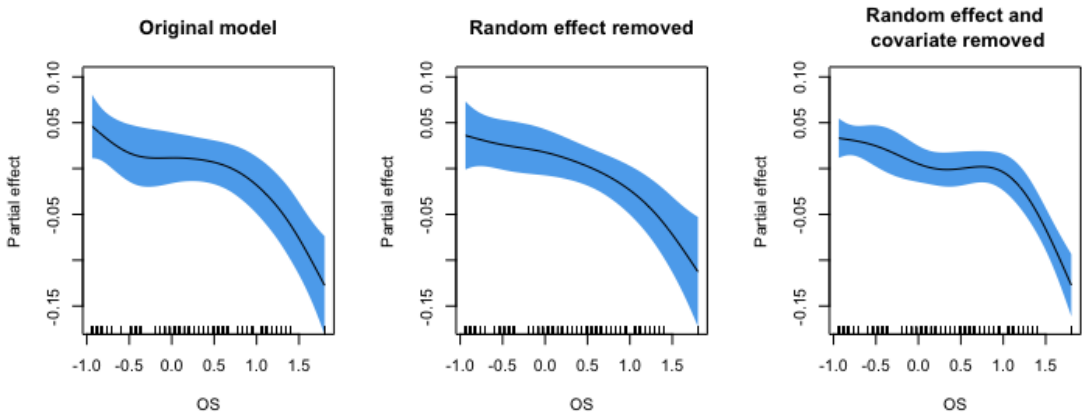
```
# Model with a covariate and a random effect removed
dijkstra.gam5 = bam(invRT  ~
                      s(OS, k=5) +
                      #s(Frequency) +
                      s(Trial, Participant, bs="fs", m=1) ,
                      #s(Word, bs="re"),
                    data = dijkstra, discrete = TRUE)
summary(dijkstra.gam5)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(OS, k = 5) + s(Trial, Participant, bs = "fs", m = 1)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.87485    0.05691  -32.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf  Ref.df     F  p-value
## s(OS)                  3.277   3.696 21.23 6.45e-16 ***
## s(Trial,Participant) 66.699 188.000 11.15  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.367   Deviance explained = 37.9%
## fREML = 1310.7  Scale est. = 0.11212   n = 3747
```

The number of basis functions used by mgcv by default is not theoretically motivated, and users are encouraged to set k to higher values than the default to make sure the model is not oversmoothing.

```
par(mfrow=c(1,3))
ylim = c(-0.17, 0.1)
```

```
plot(dijkstra.gam, select = 1, ylim = ylim, main = "Original model",
shade.col="steelblue2", scheme=1, ylab = "Partial effect")
plot(dijkstra.gam4, select = 1, ylim = ylim, main = "Random effect removed",
shade.col="steelblue2", scheme=1, ylab = "Partial effect")
plot(dijkstra.gam5, select = 1, ylim = ylim, main = "Random effect and\n
covariate removed", shade.col="steelblue2", scheme=1, ylab = "Partial
effect")
```



Regarding the by-participant random curves for *Trial*, it is possible to include a population effect for trial. The assumption would then be that all participants would show the same wiggly development through the experiment, with individual variation riding on top. As shown in the example below, such a putative curve is not significant.

```
dijkstra.gam.ml = bam(invRT  ~
                      s(OS) +
                      s(Frequency) +
                      s(Trial, Participant, bs="fs", m=1) +
                      s(Word, bs="re"),
                    data = dijkstra, discrete = TRUE, method = "ML")

## Warning in bam(invRT ~ s(OS) + s(Frequency) + s(Trial, Participant, bs =
## "fs", : discretization only available with fREML

dijkstra.gam6ml = bam(invRT  ~
                      s(OS) +
                      s(Trial) +
                      s(Frequency) +
                      s(Trial, Participant, bs="fs", m=1) +
                      s(Word, bs="re"),
                    data = dijkstra, discrete = TRUE, method = "ML")

## Warning in bam(invRT ~ s(OS) + s(Trial) + s(Frequency) + s(Trial,
## Participant, : discretization only available with fREML

## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(dijkstra.gam.ml, dijkstra.gam6ml)
```

```
## dijkstra.gam.ml: invRT ~ s(OS) + s(Frequency) + s(Trial, Participant, bs =
"fs",
##     m = 1) + s(Word, bs = "re")
##
## dijkstra.gam6ml: invRT ~ s(OS) + s(Trial) + s(Frequency) + s(Trial,
Participant,
##     bs = "fs", m = 1) + s(Word, bs = "re")
##
## Chi-square test of ML scores
## -----
##             Model    Score Edf Difference    Df p.value Sig.
## 1 dijkstra.gam.ml 1056.260   8
## 2 dijkstra.gam6ml 1056.259  10      0.000 2.000   1.000
##
## AIC difference: -0.45, model dijkstra.gam.ml has lower AIC.
```

```
## Warning in compareML(dijkstra.gam.ml, dijkstra.gam6ml): Only small
difference in ML...
```

3. Cross-language phonological similarity effect

To test whether a cross-language phonological similarity effect similarly shows nonlinearity, we reanalyze Miwa et al.'s (2014) lexical decision data. The authors tested 19 Japanese-English bilinguals and 19 English monolinguals reading English words. Here, we reanalyzed data for 228 target words for which subtitle *Frequency* data were available.

```
load("Data/miwa.rda")
```

We first fit a LMM with a second-degree polynomial, testing nonlinearity for phonological similarity (*PS*).

```
miwa.lmer1 = lmer(invRT  ~
                            poly(PS, 2, raw = T) +
                            # or PS + I(PS^2) +
                            Frequency +
                            Trial +
                            (1|Participant)+
                            (0+Trial|Participant)+
                            (0+Frequency|Participant)+
                            (1|Word),
                          data = droplevels(miwa[miwa$FirstLanguage ==
"Japanese", ]))
summary(miwa.lmer1, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## invRT ~ poly(PS, 2, raw = T) + Frequency + Trial + (1 | Participant) +
##     (0 + Trial | Participant) + (0 + Frequency | Participant) +
```

```
##     (1 | Word)
##    Data: droplevels(miwa[miwa$FirstLanguage == "Japanese", ])
##
## REML criterion at convergence: 1113.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9825 -0.6692 -0.0828  0.6023  3.5451
##
## Random effects:
##  Groups        Name        Variance  Std.Dev.
##  Word          (Intercept) 0.0114310 0.10692
##  Participant   Frequency   0.0004007 0.02002
##  Participant.1 Trial       0.0017353 0.04166
##  Participant.2 (Intercept) 0.0292469 0.17102
##  Residual                  0.0683019 0.26135
## Number of obs: 4104, groups:  Word, 228; Participant, 19
##
## Fixed effects:
##                           Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)           -1.468e+00  4.046e-02  1.992e+01 -36.280  < 2e-16
## poly(PS, 2, raw = T)1 -2.165e-02  9.566e-03  2.217e+02  -2.263   0.0246
## poly(PS, 2, raw = T)2  4.128e-04  5.553e-03  2.224e+02   0.074   0.9408
## Frequency             -8.148e-02  9.409e-03  7.840e+01  -8.660 4.76e-13
## Trial                 -1.140e-01  1.044e-02  1.804e+01 -10.914 2.23e-09
##
## (Intercept)           ***
## poly(PS, 2, raw = T)1 *
## poly(PS, 2, raw = T)2
## Frequency             ***
## Trial                 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

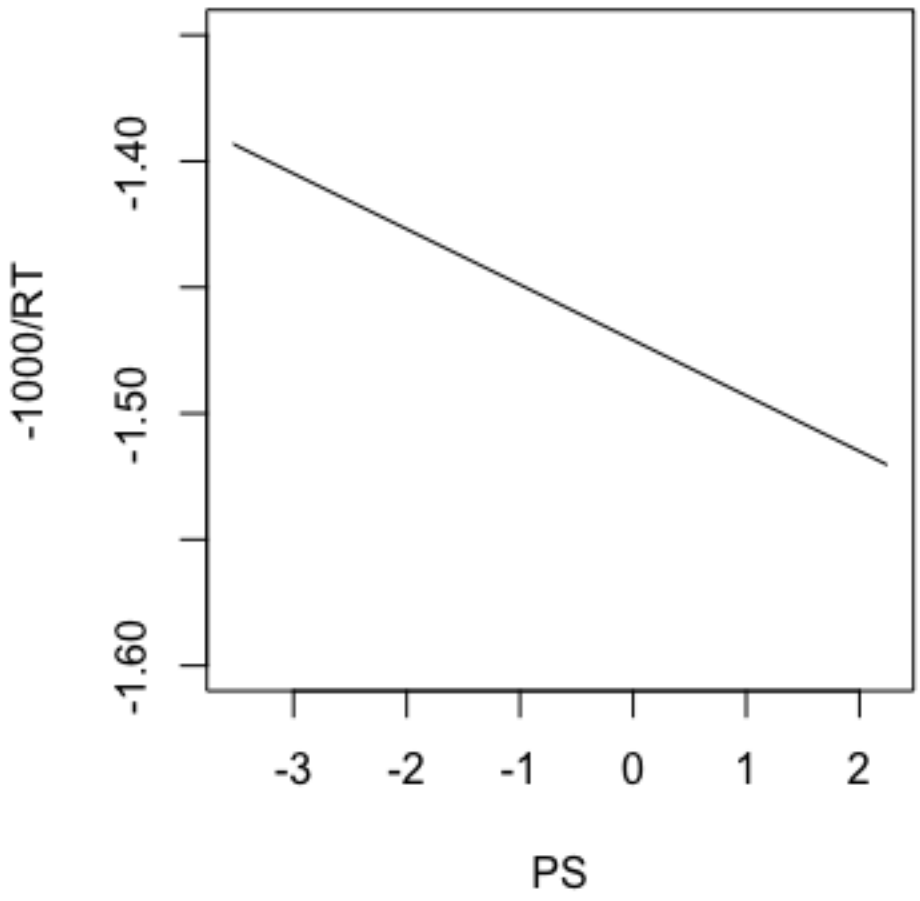
The non-linearity is not supported for *PS*. When we fit a LMM with *PS* without assuming nonlinearity, the slope is estimated to be -0.022.

```
miwa.lmer2 = lmer(invRT  ~
                            PS +
                            Frequency +
                            Trial +
                            (1|Participant)+
                            (0+Trial|Participant)+
                            (0+Frequency|Participant)+
                            (1|Word),
                          data = droplevels(miwa[miwa$FirstLanguage ==
"Japanese", ]))
summary(miwa.lmer2, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## invRT ~ PS + Frequency + Trial + (1 | Participant) + (0 + Trial |
##     Participant) + (0 + Frequency | Participant) + (1 | Word)
##    Data: droplevels(miwa[miwa$FirstLanguage == "Japanese", ])
##
## REML criterion at convergence: 1105.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9830 -0.6684 -0.0829  0.6020  3.5451
##
## Random effects:
##  Groups        Name        Variance  Std.Dev.
##  Word          (Intercept) 0.0113630 0.10660
##  Participant   Frequency   0.0004007 0.02002
##  Participant.1 Trial       0.0017356 0.04166
##  Participant.2 (Intercept) 0.0292475 0.17102
##  Residual                  0.0683020 0.26135
## Number of obs: 4104, groups:  Word, 228; Participant, 19
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)  -1.467519   0.040075  19.171944 -36.620  < 2e-16 ***
## PS           -0.022017   0.008172 225.210816  -2.694  0.00759 **
## Frequency    -0.081508   0.009387  77.926653  -8.683 4.48e-13 ***
## Trial        -0.113959   0.010443  18.042736 -10.913 2.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plotLMER.fnc(miwa.lmer2, pred="PS" ,ylim =c(-1.6, -1.35), xlabel = "PS",
ylabel = "-1000/RT", main = "")
```

```
## effect size (range) for  PS is  0.1272519
```



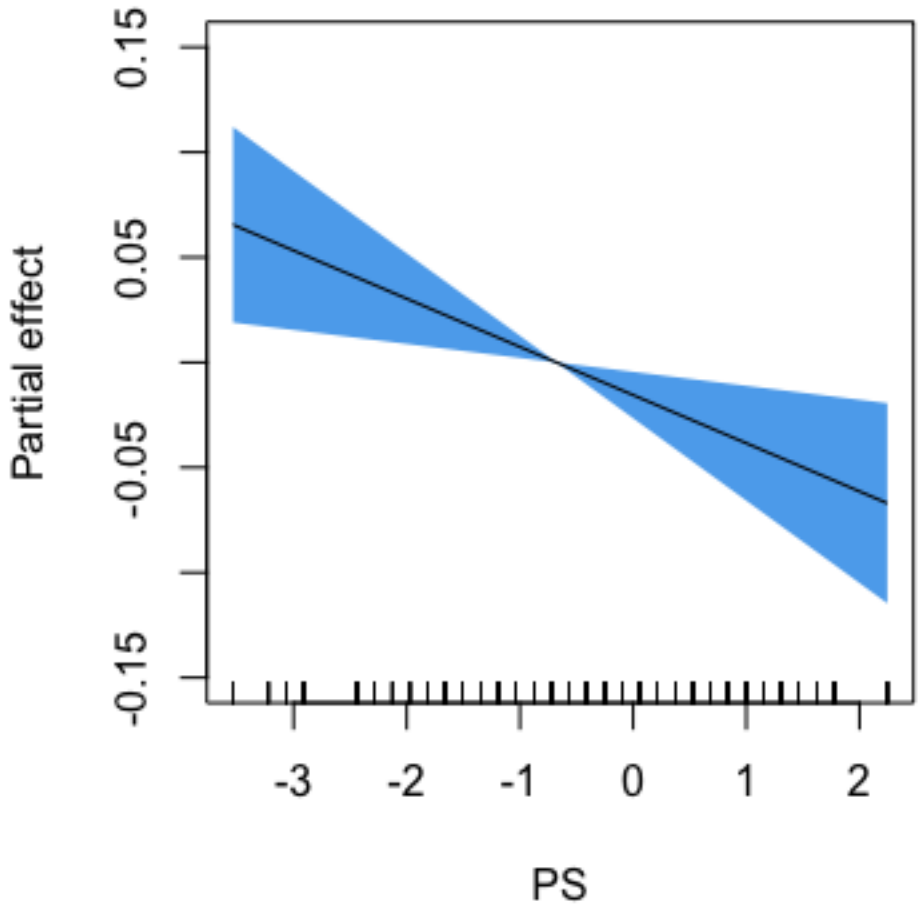
Now, we test the nonlinearity using a GAMM. The following model tests a nonlinear effect of phonological similarity (*PS*), using the thin plate regression spline.

```
miwa.gam1 = bam(invRT ~
                  s(PS) +
                  s(Frequency) +
                  s(Participant, Trial, bs = "fs", m=1) +
                  s(Participant, Frequency, bs = "re") +
                  s(Word, bs = "re") ,
                data = droplevels(miwa[miwa$FirstLanguage == "Japanese", ]),
discrete=T)
summary(miwa.gam1)

##
## Family: gaussian
## Link function: identity
```

```
##
## Formula:
## invRT ~ s(PS) + s(Frequency) + s(Participant, Trial, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.3903     0.0461  -30.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf  Ref.df      F p-value
## s(PS)                      1.000   1.000  7.928 0.00489 **
## s(Frequency)               2.965   3.149 27.691 < 2e-16 ***
## s(Participant,Trial)     119.911 171.000 19.435 < 2e-16 ***
## s(Frequency,Participant)   8.650  18.000  0.934 0.01524 *
## s(Word)                  168.444 228.000  3.468 0.34567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.513   Deviance explained = 54.9%
## fREML = 495.53  Scale est. = 0.063238  n = 4104
```

Nonlinearity was not found in the phonological similarity effect (i.e., edf = 1.000). GAMMs are designed in such a way that if the true effect is linear, the model will discover this and report the effect as such.

```
plot(miwa.gam1, select = 1, shade = T, shade.col = "steelblue2", ylab =
"Partial effect", main = "", xlab = "PS", ylim =c(-0.15, 0.15))
```



Because *PS* does not show any nonlinear effect, we move it to the parameteric part of the model so that the slope (-0.023) can be obtained. The fact that the effect of *PS* is linear is perhaps not surprising given the tiny nonlinearity of *OS* for English/Dutch, where identical orthography is a strong factor driving the nonlinearity.

```
miwa.gam1b = bam(invRT ~
                   PS +
                   s(Frequency) +
                   s(Participant, Trial, bs = "fs", m=1) +
                   s(Participant, Frequency, bs = "re") +
                   s(Word, bs = "re") ,
                 data = droplevels(miwa[miwa$FirstLanguage == "Japanese", ]),
discrete=T)
summary(miwa.gam1b)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ PS + s(Frequency) + s(Participant, Trial, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.41095    0.04578 -30.824  < 2e-16 ***
## PS          -0.02292    0.00814  -2.816  0.00489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                               edf  Ref.df      F p-value
## s(Frequency)                2.965   3.149 27.691  <2e-16 ***
## s(Participant,Trial)      119.911 171.000 19.550  <2e-16 ***
## s(Frequency,Participant)    8.650  18.000  0.934  0.0116 *
## s(Word)                   168.445 228.000  3.300  0.0989 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.513   Deviance explained = 54.9%
## fREML = 495.98  Scale est. = 0.063238  n = 4104
```

The by-participant random slopes for *Frequency* and *PS* are supported.

```
miwa.gam1c = bam(invRT ~
                   s(PS) +
                   s(Frequency) +
                   s(Participant, Trial, bs = "fs", m=1) +
                   #s(Participant, Frequency, bs = "re") +
                   #s(Participant, PS, bs = "re") +
                   s(Word, bs = "re") ,
                 data = droplevels(miwa[miwa$FirstLanguage == "Japanese", ]),
           discrete=TRUE)
compareML(miwa.gam1c, miwa.gam1)
```

```
## miwa.gam1c: invRT ~ s(PS) + s(Frequency) + s(Participant, Trial, bs =
"fs",
##     m = 1) + s(Word, bs = "re")
##
## miwa.gam1: invRT ~ s(PS) + s(Frequency) + s(Participant, Trial, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Word, bs = "re")
##
## Chi-square test of fREML scores
## -----
##        Model    Score Edf Difference    Df p.value Sig.
```

```
## 1 miwa.gam1c 497.9692   8
## 2  miwa.gam1 495.5266   9      2.443 1.000   0.027  *
##
## AIC difference: 6.70, model miwa.gam1 has lower AIC.

## Warning in compareML(miwa.gam1c, miwa.gam1): Only small difference in
fREML...
```

4. Responses to words and nonwords

To understand how participants responded to words and nonwords throughout the experiment in detail, we test how RTs changed as the experiment went by (*Trial*) for different levels of the factor *SimulusType* (levels: *Word*, *Nonword*). We use data from Miwa et al. (2014) study (19 Japanese, 19 English monolinguals, 250 words, 200 nonwords),

```
load("Data/miwacomp.rda")
miwacomp = miwacomp[miwacomp$ResponseCorrect == 1,]
miwacomp = droplevels(miwacomp[miwacomp$invRT > -5,]) # Remove outliers
```

In a typical lexical decision experiment, RTs to nonwords are generally longer than RTs to words. The Miwa et al. (2014) study was no exception; bilingual participants responded to words faster than to nonwords (median RTs: 670 ms vs. 763 ms), and monolingual participants similarly responded to words faster than to nonwords (median RTs: 516 ms vs. 593 ms).

```
setNames(aggregate(miwacomp$RT, list(miwacomp$StimulusType,
miwacomp$FirstLanguage), median), c("SimulusType","FirstLanguage","RT"))

##   SimulusType FirstLanguage     RT
## 1     Nonword       English 592.99
## 2        Word       English 515.50
## 3     Nonword      Japanese 763.15
## 4        Word      Japanese 670.38
```

We first fit two LMMs, one for each participant group, testing whether the effect of *Trial* is modulated by the factor *StimulusType*.

```
miwa.lmer3.jpn = lmer(invRT ~
                  Trial * StimulusType +
                  (1|Participant)+
                  (0+Trial|Participant)+
                  (1|Word),
                data = miwacomp[miwacomp$FirstLanguage == "Japanese", ])
summary(miwa.lmer3.jpn, corr=F)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: invRT ~ Trial * StimulusType + (1 | Participant) + (0 + Trial |
##     Participant) + (1 | Word)
##    Data: miwacomp[miwacomp$FirstLanguage == "Japanese", ]
##
```

```
## REML criterion at convergence: 1827.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1182 -0.6699 -0.0599  0.6121  3.5257
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Word          (Intercept) 0.018856 0.13732
##  Participant   Trial       0.002374 0.04872
##  Participant.1 (Intercept) 0.029413 0.17150
##  Residual                  0.065421 0.25578
## Number of obs: 7728, groups:  Word, 450; Participant, 19
##
## Fixed effects:
##                            Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)            -1.281e+00  4.077e-02  2.052e+01 -31.419  < 2e-16
## Trial                  -9.495e-02  1.204e-02  2.118e+01  -7.884 9.78e-08
## StimulusTypeWord       -1.624e-01  1.432e-02  4.358e+02 -11.344  < 2e-16
## Trial:StimulusTypeWord -1.808e-02  6.019e-03  7.371e+03  -3.003  0.00268
##
## (Intercept)            ***
## Trial                  ***
## StimulusTypeWord       ***
## Trial:StimulusTypeWord **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

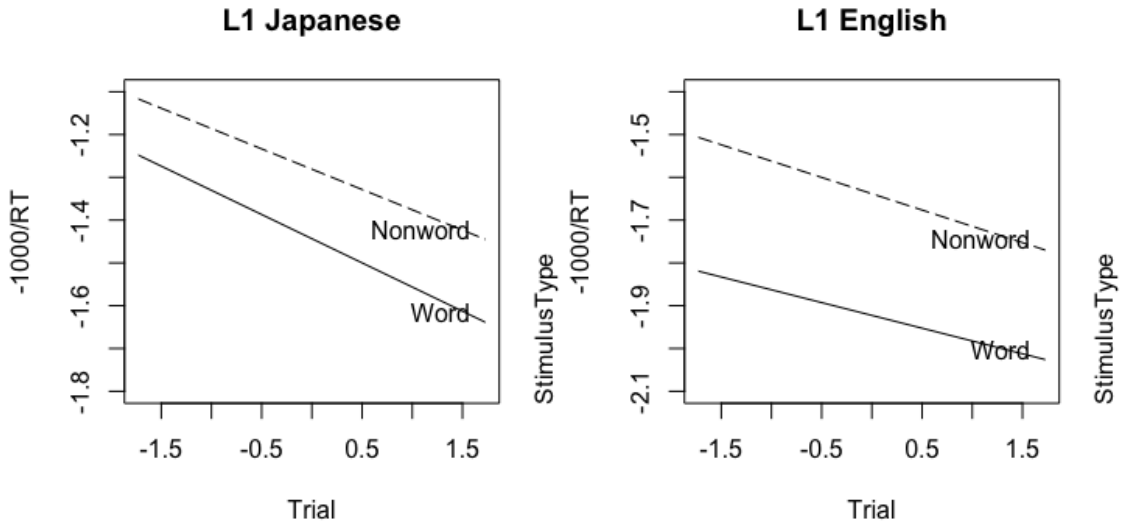
```
miwa.lmer3.engl = lmer(invRT ~
                  Trial * StimulusType +
                  (1|Participant)+
                  (0+Trial|Participant)+
                  (1|Word),
                data = miwacomp[miwacomp$FirstLanguage == "English", ])
summary(miwa.lmer3.engl, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: invRT ~ Trial * StimulusType + (1 | Participant) + (0 + Trial |
##     Participant) + (1 | Word)
##    Data: miwacomp[miwacomp$FirstLanguage == "English", ]
##
## REML criterion at convergence: 3261.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.2292 -0.6314 -0.0439  0.5879  4.5527
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
```

```
##  Word          (Intercept) 0.009841 0.09920
##  Participant   Trial       0.002563 0.05062
##  Participant.1 (Intercept) 0.035333 0.18797
##  Residual                  0.079724 0.28235
## Number of obs: 8281, groups:  Word, 450; Participant, 19
##
## Fixed effects:
##                           Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)            -1.638e+00  4.394e-02  1.921e+01 -37.280  < 2e-16
## Trial                  -7.624e-02  1.257e-02  2.132e+01  -6.066 4.77e-06
## StimulusTypeWord       -2.843e-01  1.132e-02  4.436e+02 -25.129  < 2e-16
## Trial:StimulusTypeWord  1.641e-02  6.389e-03  7.995e+03   2.568   0.0102
##
## (Intercept)            ***
## Trial                  ***
## StimulusTypeWord       ***
## Trial:StimulusTypeWord *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction between *Trial* and *StimulusTypeWord* is significant for both bilingual (L1 Japanese) and monolingual (L1 English) participants. When visualized, however, the difference between words and nonwords is subtle for both groups. In fact, since the p-values for the interactions in both miwa.lmer3.jpn and miwa.lmer3.engl are quite large compared to those estimated for the other predictors, especially in the context of exploratory data analysis, it is advisable to treat these interactions with great caution.

```
par(mfrow=c(1,2))
plotLMER.fnc(miwa.lmer3.jpn, pred="Trial",intr=list("StimulusType",
c("Nonword", "Word"), "end",  list(c("black",  "black"), c(5,1), rep(1,2)))
,ylim =c(-1.8, -1.1), xlabel = "Trial", ylabel = "-1000/RT", main = "L1
Japanese", cex = 1, verbose=F)
plotLMER.fnc(miwa.lmer3.engl, pred="Trial",intr=list("StimulusType",
c("Nonword", "Word"), "end",  list(c("black",  "black"), c(5,1), rep(1,2)))
,ylim =c(-2.1, -1.4), xlabel = "Trial", ylabel = "-1000/RT", main = "L1
English", cex = 1, verbose=F)
```

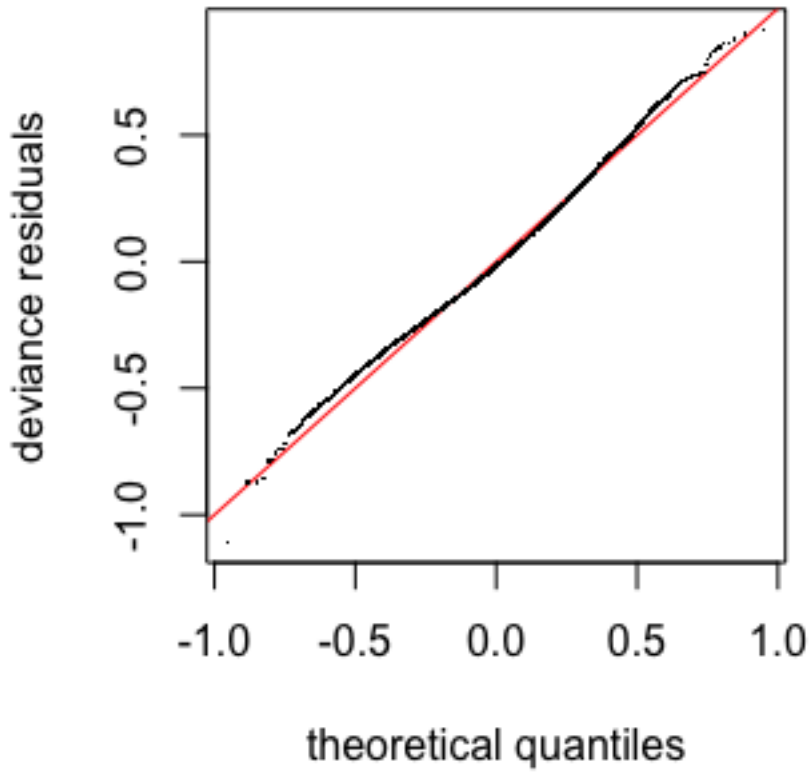


Now, we move to the GAMM. With the GAMM, it is possible to test an interaction between a nonlinear predictor and a factor. We fit a GAMM with the by-directive within the `s()` function, which requests two wiggly curves, one for each factor level. The main effect of *StimulusType* should be included so that two wiggly curves can be set at appropriate places on the y-axis.

```
jpn.gam1 = bam(invRT ~
                 StimulusType  +
                 s(Trial, by = StimulusType) +
                 s(Trial, Participant, bs = "fs", m = 1) +
                 s(Word, bs = "re"),
               data = miwacomp[miwacomp$FirstLanguage == "Japanese", ],
discrete=TRUE)
summary(jpn.gam1)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ StimulusType + s(Trial, by = StimulusType) + s(Trial,
##     Participant, bs = "fs", m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.27303    0.04296  -29.63   <2e-16 ***
## StimulusTypeWord -0.16258    0.01427  -11.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                   edf  Ref.df      F  p-value
```

```
## s(Trial):StimulusTypeNonword   1.002   1.003 30.483 3.42e-08 ***
## s(Trial):StimulusTypeWord      5.932   6.956 13.940  < 2e-16 ***
## s(Trial,Participant)         114.920 170.000 25.649  < 2e-16 ***
## s(Word)                      374.279 448.000  5.057  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.534   Deviance explained = 56.4%
## fREML = 786.76  Scale est. = 0.06169   n = 7728
```

```
qq.gam(jpn.gam1) # Check residuals
```



It is clear that participants showed different learning effects for words and nonwords throughout the experiment: nonlinear for words and linear for nonwords. The result shows much more nonlinearity in the *Participant* × *Trial* random effect, compared to English-Ducth bilinguals tested by Dijkstra et al. (2010). The task was obviously more difficult for Japanese-English bilinguals.

To find out whether the *Trial*×*StimulusType* interaction is significant, we use the `compareML()` function from the `itsadug` package (van Rij et al., 2017). In this example, the interaction indeed reduced the AIC score. When there are factor smooths in the model, GAMMs sometimes present warnings, which we can safely ignore (Simon Wood, personal communication).

```
jpn.gam1ml = bam(invRT ~
                   StimulusType +
                   s(Trial)+
                   s(Trial, Participant, bs = "fs", m=1) +
                   s(Word, bs = "re"),
                 data = miwacomp[miwacomp$FirstLanguage == "Japanese",],
method = "ML")

## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.

jpn.gam2ml = bam(invRT ~
                   StimulusType +
                   s(Trial, by = StimulusType)+
                   s(Trial, Participant, bs = "fs", m=1) +
                   s(Word, bs = "re"),
                 data = miwacomp[miwacomp$FirstLanguage == "Japanese",],
method = "ML")
compareML(jpn.gam1ml,jpn.gam2ml)

## jpn.gam1ml: invRT ~ StimulusType + s(Trial) + s(Trial, Participant, bs =
"fs",
##     m = 1) + s(Word, bs = "re")
##
## jpn.gam2ml: invRT ~ StimulusType + s(Trial, by = StimulusType) + s(Trial,
##     Participant, bs = "fs", m = 1) + s(Word, bs = "re")
##
## Chi-square test of ML scores
## -----
##        Model    Score Edf Difference    Df   p.value Sig.
## 1 jpn.gam1ml 792.7048   7
## 2 jpn.gam2ml 776.4201   9     16.285 2.000 8.466e-08  ***
##
## AIC difference: 43.42, model jpn.gam2ml has lower AIC.
```

A model with the same parameters is also fitted to the monolinguals' data.

```
eng.gam1 = bam(invRT ~
                 StimulusType +
                 s(Trial, by = StimulusType) +
                 s(Trial, Participant, bs = "fs", m=1) +
                 s(Word, bs = "re"),
               data = miwacomp[miwacomp$FirstLanguage == "English",],
discrete=TRUE)
summary(eng.gam1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ StimulusType + s(Trial, by = StimulusType) + s(Trial,
##     Participant, bs = "fs", m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.62399    0.04669  -34.78   <2e-16 ***
## StimulusTypeWord -0.28320    0.01129  -25.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                   edf  Ref.df      F  p-value
## s(Trial):StimulusTypeNonword   3.255   3.976  8.050  2.2e-06 ***
## s(Trial):StimulusTypeWord      1.000   1.000 12.198 0.000481 ***
## s(Trial,Participant)         115.267 170.000 25.322  < 2e-16 ***
## s(Word)                      314.730 448.000  2.742  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.488   Deviance explained = 51.5%
## fREML = 1536.8  Scale est. = 0.076213  n = 8281
```

To see how the two curves are different, it is useful to compute a difference curve for *Word* and *Nonword*. For this, the factor *StimulusType* is first converted to a numerical predictor *StimulusTypeNum* (i.e., 1 for *Nonword* and 0 for *Word*), and the additive contribution of the wiggly regression line for *Nonword* is assessed.

```
miwacomp$StimulusTypeNum = ifelse(miwacomp$StimulusType == "Nonword",1, 0)
jpn.gam1num = bam(invRT ~
                  s(Trial)  +
                  s(Trial, by = StimulusTypeNum) +
                  s(Trial, Participant, bs = "fs", m = 1) +
                  s(Word, bs = "re"),
                data = miwacomp[miwacomp$FirstLanguage == "Japanese", ],
discrete=TRUE)

## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.

summary(jpn.gam1num)

##
## Family: gaussian
## Link function: identity
##
## Formula:
```

```
## invRT ~ s(Trial) + s(Trial, by = StimulusTypeNum) + s(Trial,
##     Participant, bs = "fs", m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.44205    0.04401  -32.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                               edf  Ref.df      F p-value
## s(Trial)                    5.999   6.772 12.960 3.6e-16 ***
## s(Trial):StimulusTypeNum    6.947   8.029 22.265 < 2e-16 ***
## s(Trial,Participant)      110.093 170.000 25.493 < 2e-16 ***
## s(Word)                   374.418 448.000  5.026 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.535   Deviance explained = 56.5%
## fREML = 785.86  Scale est. = 0.061636  n = 7728
```

Using the same procedure, we compute the difference curve for monolingual participants.

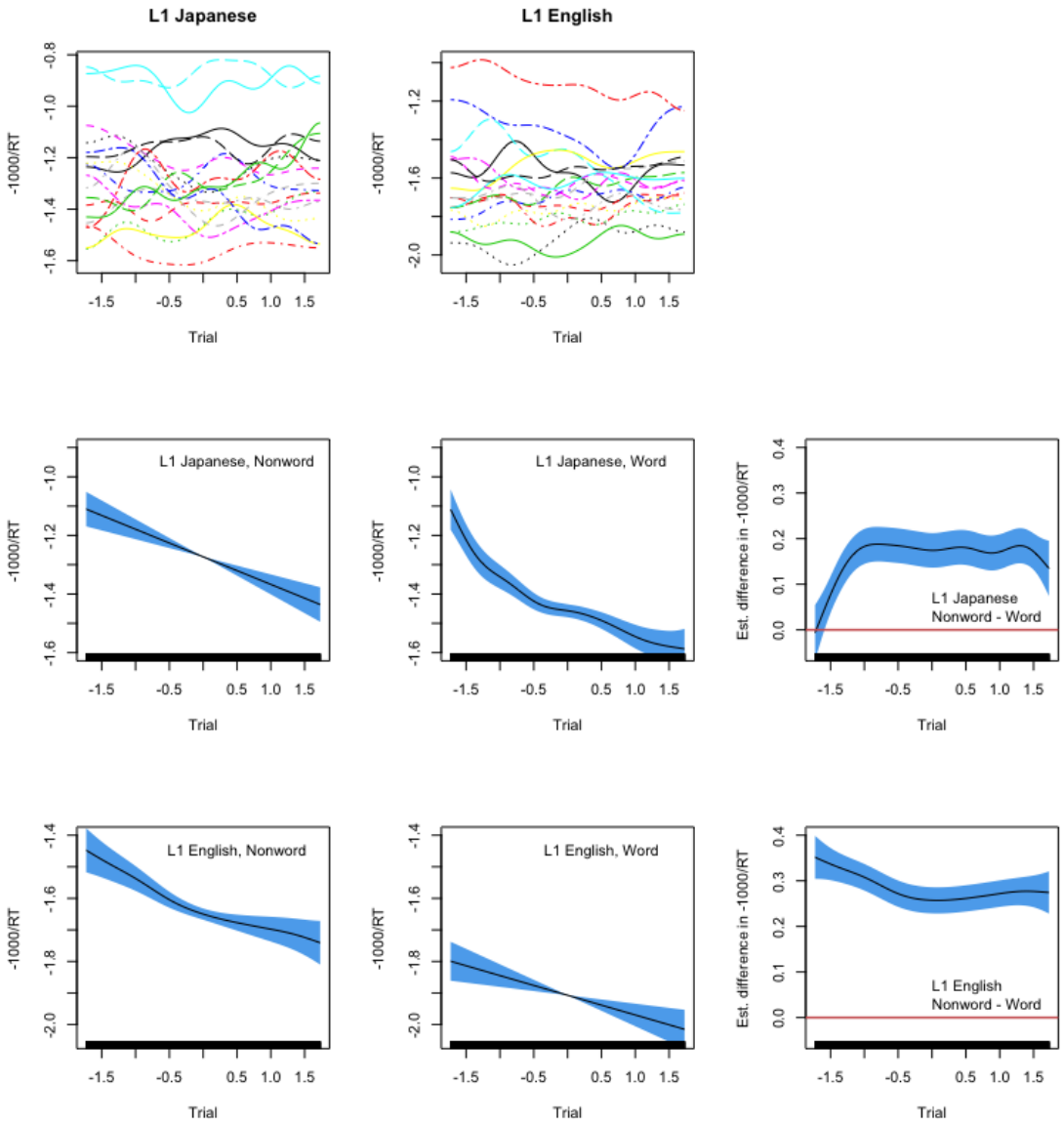
```
miwacomp$StimulusTypeNum = ifelse(miwacomp$StimulusType == "Nonword",1,0)
eng.gam1num = bam(invRT ~
                  s(Trial)  +
                  s(Trial, by = StimulusTypeNum) +
                  s(Trial, Participant, bs = "fs", m = 1) +
                  s(Word, bs = "re"),
                data = miwacomp[miwacomp$FirstLanguage == "English", ],
discrete=TRUE)

## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.

summary(eng.gam1num)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ s(Trial) + s(Trial, by = StimulusTypeNum) + s(Trial,
##     Participant, bs = "fs", m = 1) + s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.90719    0.04653  -40.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Approximate significance of smooth terms:
##                               edf  Ref.df       F  p-value
## s(Trial)                    1.000   1.000  12.199 0.000481 ***
## s(Trial):StimulusTypeNum    4.255   4.976 130.088  < 2e-16 ***
## s(Trial,Participant)      115.267 170.000  25.322  < 2e-16 ***
## s(Word)                   314.730 448.000   2.658  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.488   Deviance explained = 51.5%
## fREML = 1536.8  Scale est. = 0.076213  n = 8281
```

The results are now visualized. The top panels visualize factor smooths for *Participant* × *Trial*.

```
par(oma=rep(0,4), mfrow=c(3,3))
plot(jpn.gam1, select = 3, shift = coef(jpn.gam1)[1], main = "", ylab = "-
1000/RT"); title("L1 Japanese")
plot(eng.gam1, select = 3, shift = coef(eng.gam1)[1], main = "", ylab = "-
1000/RT"); title("L1 English")
plot(0, xaxt = "n", yaxt = "n", bty = "n", pch = "", ylab = "", xlab = "")
plot(jpn.gam1, select = 1, shift = coef(jpn.gam1)[1], main = "", shade = T,
shade.col= "steelblue2",ylab = "-1000/RT", ylim = c(-1.6, -0.9))
text(0.5, -0.95, "L1 Japanese, Nonword")
plot(jpn.gam1, select = 2, shift = coef(jpn.gam1)[1]+coef(jpn.gam1)[2], main
= "", shade = T,shade.col = "steelblue2", ylab = "-1000/RT", ylim = c(-1.6, -
0.9))
text(0.5, -0.95, "L1 Japanese, Word")
plot(jpn.gam1num, select = 2, main = "", shade = T,shade.col = "steelblue2",
ylab = "Est. difference in -1000/RT", ylim = c(-0.05, 0.4)); abline(h= 0,
col="indianred", lwd = 1.5)
#It is also possible to use plot_diff(jpn.gam1, view='Trial',
comp=list(StimulusType=c("nonword", "word")), main = "", ylab = "Est.
difference in -1000/RT", mark.diff = T, col.diff = "steelblue2")
text(0, 0.06, "L1 Japanese", adj=c(0,0))
text(0, 0.02, "Nonword - Word", adj=c(0,0))

plot(eng.gam1, select = 1, shift = coef(eng.gam1)[1], main = "", shade =
T,shade.col = "steelblue2", ylab = "-1000/RT", ylim = c(-2.05, -1.4))
text(0.5, -1.45, "L1 English, Nonword")
plot(eng.gam1, select = 2, shift = coef(eng.gam1)[1]+coef(eng.gam1)[2], main
= "", shade = T,shade.col = "steelblue2", ylab = "-1000/RT", ylim = c(-2.05,
-1.4))
text(0.5, -1.45, "L1 English, Word")
plot(eng.gam1num, select = 2, main = "", shade = T,shade.col = "steelblue2",
ylab = "Est. difference in -1000/RT", ylim = c(-0.05, 0.4)); abline(h= 0,
col="indianred", lwd = 1.5)
#It is also possible to use plot_diff(eng.gam1, view='Trial',
comp=list(StimulusType=c("nonword", "word")), main = "", mark.diff=T,
col.diff = "steelblue2")
```

```
text(0, 0.06, "L1 English", adj =c(0,0))
text(0, 0.02, "Nonword - Word", adj = c(0,0))
```



5. Cross-language semantic similarity and word frequency effects

Under the assumption that languages have semantic representations in common (see, e.g., Bilingual Interactive Activation (BIA+) model, a model of bilingual visual word recognition with orthographic, phonological, and semantic representations, Dijkstra & van Heuven, 2002), cross-language semantic similarity (henceforth *SS*) is expected to facilitate word recognition. Furthermore, because bottom-up processing to the semantic level should proceed faster for higher frequency words, there may be more opportunity for *SS* to contribute for these words. Here we test the interaction between *SS* and *Frequency* and further assume that the pattern of interaction is different for different *FirstLaguage* groups.

We first fit a LMM with this three-way interaction. *Trial* is also included as a control variable.

```
miwa.lmer3 = lmer(invRT  ~
                            SS*Frequency*FirstLanguage +
                            Trial*FirstLanguage +
                            (1|Participant)+
                            (0+Trial|Participant)+
                            (0+Frequency|Participant)+
                            (1|Word),
                          data = miwa)
summary(miwa.lmer3, corr=F)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: invRT ~ SS * Frequency * FirstLanguage + Trial * FirstLanguage +
##     (1 | Participant) + (0 + Trial | Participant) + (0 + Frequency |
##     Participant) + (1 | Word)
##    Data: miwa
##
## REML criterion at convergence: 3003.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3731 -0.6467 -0.0641  0.5878  4.9792
##
## Random effects:
##  Groups        Name        Variance  Std.Dev.
##  Word          (Intercept) 0.0066869 0.08177
##  Participant   Frequency   0.0003102 0.01761
##  Participant.1 Trial       0.0023733 0.04872
##  Participant.2 (Intercept) 0.0302636 0.17396
##  Residual                  0.0774537 0.27830
## Number of obs: 8389, groups:  Word, 228; Participant, 38
##
## Fixed effects:
##                                     Estimate Std. Error         df
## (Intercept)                       -1.941e+00  4.052e-02  3.739e+01
## SS                                -1.546e-02  7.109e-03  3.377e+02
## Frequency                         -5.358e-02  8.173e-03  1.080e+02
## FirstLanguageJapanese              4.734e-01  5.678e-02  3.604e+01
## Trial                             -5.870e-02  1.197e-02  3.598e+01
## SS:Frequency                       1.164e-02  7.715e-03  3.375e+02
## SS:FirstLanguageJapanese          -5.935e-03  6.295e-03  8.063e+03
## Frequency:FirstLanguageJapanese   -2.554e-02  8.511e-03  3.885e+01
## FirstLanguageJapanese:Trial       -5.580e-02  1.696e-02  3.623e+01
## SS:Frequency:FirstLanguageJapanese -1.681e-02  6.871e-03  8.067e+03
##                                   t value Pr(>|t|)
## (Intercept)                       -47.895  < 2e-16 ***
## SS                                 -2.175  0.03030 *
```

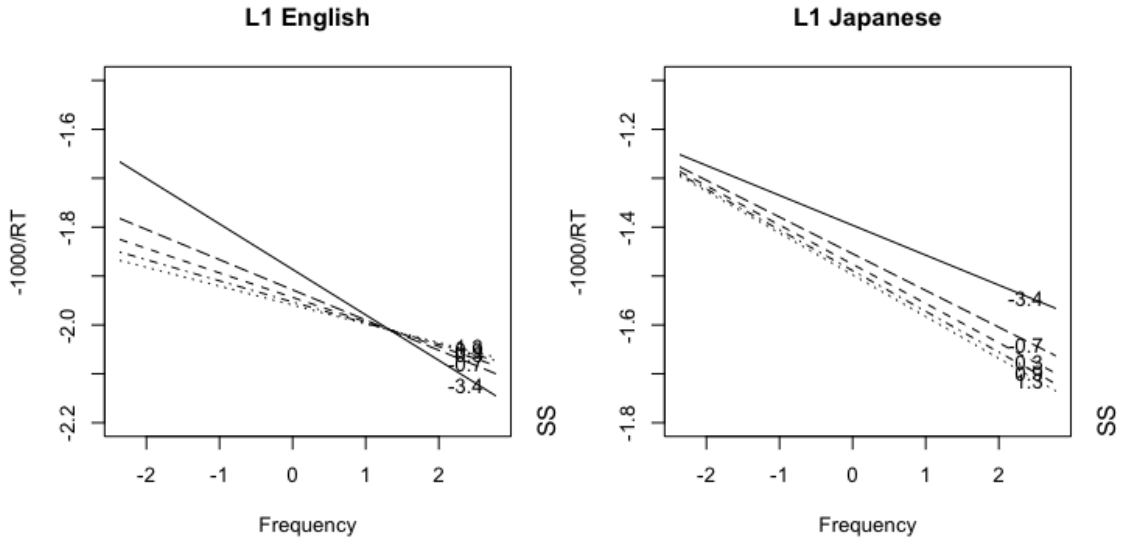
```
## Frequency                                -6.555 1.96e-09 ***
## FirstLanguageJapanese                     8.337 6.26e-10 ***
## Trial                                    -4.902 2.03e-05 ***
## SS:Frequency                              1.509  0.13232
## SS:FirstLanguageJapanese                 -0.943  0.34576
## Frequency:FirstLanguageJapanese          -3.001  0.00468 **
## FirstLanguageJapanese:Trial              -3.289  0.00224 **
## SS:Frequency:FirstLanguageJapanese       -2.446  0.01446 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

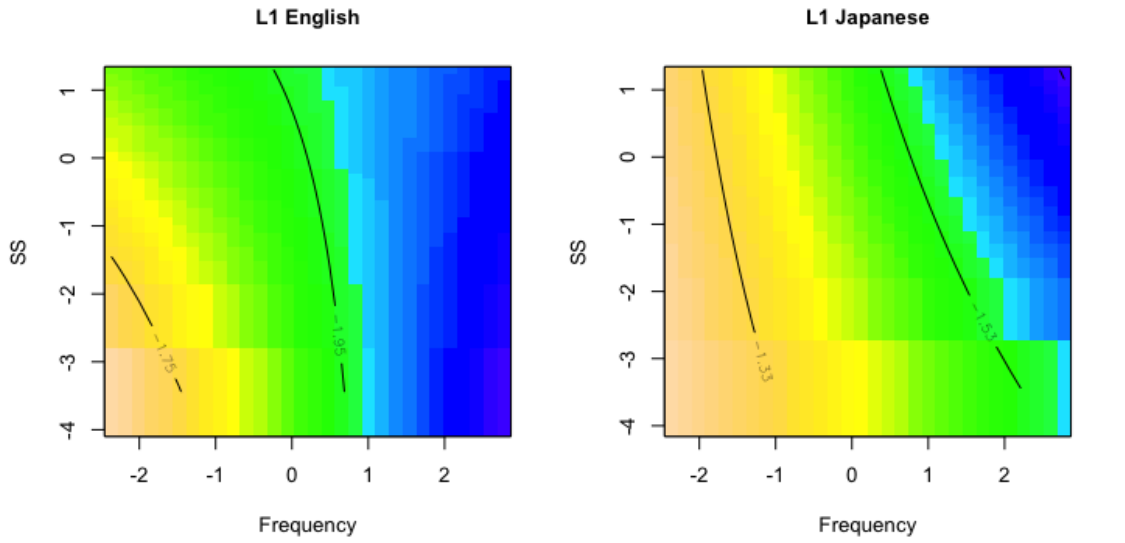
The interaction is significant. It is visualized below, using perspective plots and contour plots, which show the same results in different ways.

```
miwa.lmer3.jpn = lmer(invRT  ~
                            SS*Frequency +
                            Trial +
                            (1|Participant)+
                            (0+Trial|Participant)+
                            (0+Frequency|Participant)+
                            (1|Word),
                          data = droplevels(miwa[miwa$FirstLanguage ==
"Japanese", ]))
miwa.lmer3.engl = lmer(invRT  ~
                            SS*Frequency +
                            Trial +
                            (1|Participant)+
                            (0+Trial|Participant)+
                            (0+Frequency|Participant)+
                            (1|Word),
                          data = droplevels(miwa[miwa$FirstLanguage ==
"English", ]))

par(mfrow=c(2,2))
plotLMER.fnc(miwa.lmer3.engl , pred="Frequency",intr=list("SS",
round(quantile(miwa$SS),1), "end",  list(c("black",  "black",  "black",
"black",  "black"), c(1,5,2,4,3), rep(1,5))) ,ylim =c(-2.2, -1.5), xlabel =
"Frequency", ylabel = "-1000/RT", main = "L1 English", cex = 1, verbose=F)
plotLMER.fnc(miwa.lmer3.jpn, pred="Frequency",intr=list("SS",
round(quantile(miwa$SS),1), "end",  list(c("black",  "black",  "black",
"black",  "black"), c(1,5,2,4,3), rep(1,5))) ,ylim =c(-1.8, -1.1), xlabel =
"Frequency", ylabel = "-1000/RT", main = "L1 Japanese", cex = 1, verbose=F)

plotLMER3d.fnc(miwa.lmer3.engl, pred="Frequency",intr="SS",  plot.type =
"contour", main = "L1 English")

## effect sizes (ranges) for the interaction of  Frequency  and  SS :
##     SS  =  -3.444185 :  0.4813693
##     SS  =  -2.135803 :  0.4033064
##     SS  =  -1.594404 :  0.3710046
```

```
##     SS  =  -1.413937 :  0.3602373
##     SS  =  -1.188354 :  0.3467782
##     SS  =  -1.012901 :  0.33631
##     SS  =  -0.9627713 :  0.3333191
##     SS  =  -0.7694144 :  0.3217827
##     SS  =  -0.7121235 :  0.3183645
##     SS  =  -0.5116052 :  0.3064009
##     SS  =  -0.3612165 :  0.2974282
##     SS  =  -0.2602412 :  0.2914036
##     SS  =  -0.2108278 :  0.2884554
##     SS  =  -0.1105687 :  0.2824736
##     SS  =  0.00401319 :  0.2756372
##     SS  =  0.1902087 :  0.2645281
##     SS  =  0.390727 :  0.2525645
##     SS  =  0.390727 :  0.2525645
##     SS  =  0.5912452 :  0.2406008
##     SS  =  0.61631 :  0.2391054
##     SS  =  0.6915044 :  0.234619
##     SS  =  0.7917635 :  0.2286372
##     SS  =  0.8920226 :  0.2226554
##     SS  =  0.8920226 :  0.2226554
##     SS  =  0.8920226 :  0.2226554
##     SS  =  0.9922818 :  0.2166735
##     SS  =  0.9922818 :  0.2166735
##     SS  =  1.092541 :  0.2106917
##     SS  =  1.1928 :  0.2047099
##     SS  =  1.293059 :  0.1987281
##     SS  =  1.293059 :  0.1987281

plotLMER3d.fnc(miwa.lmer3.jpn, pred="Frequency",intr="SS",  plot.type =
"contour", main = "L1 Japanese")

## effect sizes (ranges) for the interaction of  Frequency  and  SS :
##     SS  =  -3.444185 :  0.3135175
##     SS  =  -2.015492 :  0.3515125
##     SS  =  -1.594404 :  0.3627111
##     SS  =  -1.413937 :  0.3675105
##     SS  =  -1.188354 :  0.3735097
##     SS  =  -1.012901 :  0.3781758
##     SS  =  -0.9627713 :  0.3795089
##     SS  =  -0.7694144 :  0.3846511
##     SS  =  -0.6920716 :  0.386708
##     SS  =  -0.5116052 :  0.3915074
##     SS  =  -0.3612165 :  0.3955068
##     SS  =  -0.2108278 :  0.3995063
##     SS  =  -0.1732306 :  0.4005062
##     SS  =  -0.1105687 :  0.4021726
##     SS  =  0.05235242 :  0.4065054
##     SS  =  0.2779355 :  0.4125046
##     SS  =  0.390727 :  0.4155042
```

```
##      SS  =  0.390727 :  0.4155042
##      SS  =  0.5912452 :  0.4208369
##      SS  =  0.61631 :  0.4215035
##      SS  =  0.6915044 :  0.4235032
##      SS  =  0.7917635 :  0.4261695
##      SS  =  0.8920226 :  0.4288358
##      SS  =  0.8920226 :  0.4288358
##      SS  =  0.8920226 :  0.4288358
##      SS  =  0.9922818 :  0.4315022
##      SS  =  0.9922818 :  0.4315022
##      SS  =  1.092541 :  0.4341685
##      SS  =  1.1928 :  0.4368348
##      SS  =  1.293059 :  0.4395011
##      SS  =  1.293059 :  0.4395011
```





Using the GAMM, we can also fit a model with interaction between two nonlinear predictors, using the tensor product smooth with the `te()` function. The LMM models interactions of two covariates by means of (part of) a hyperbolic plane. However, interactions between covariates are often not well-described by this functional form. GAMMs provide several tools for modeling such more complex interactions. In what follows, we use the so-called tensor product smooth, which is appropriate for predictors that are not isometric (i.e., that are not expressed on the same scale). For technical details on tensor product smooths, see Wood, 2017, and for a conceptual introduction, Baayen et al., 2017).

In this example, we first test whether an effect of semantic similarity (*SS*) is modulated by *Frequency*, by specifiying `te(SS, Frequency)`.

```
miwa.gam2 = bam(invRT ~
                  FirstLanguage +
                  te(SS, Frequency) +
                  s(Trial, by=FirstLanguage) +
                  s(Trial, Participant, bs = "fs", m=1) +
                  s(Participant, Frequency, bs = "re") +
                  s(Word, bs = "re"),
                data = miwa, discrete=TRUE)
summary(miwa.gam2)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ FirstLanguage + te(SS, Frequency) + s(Trial, by = FirstLanguage) +
##     s(Trial, Participant, bs = "fs", m = 1) + s(Participant,
##     Frequency, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.91689    0.04583 -41.829  < 2e-16 ***
## FirstLanguageJapanese  0.45485    0.06527   6.969 3.45e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                    edf  Ref.df      F  p-value
## te(SS,Frequency)                 3.001   3.001 34.395  < 2e-16 ***
## s(Trial):FirstLanguageEnglish    1.000   1.000 10.236  0.00138 **
## s(Trial):FirstLanguageJapanese   5.134   5.939 12.098 2.18e-13 ***
## s(Trial,Participant)           190.342 342.000 11.838  < 2e-16 ***
## s(Frequency,Participant)        22.080  37.000  1.488 1.23e-06 ***
## s(Word)                        172.272 224.000  3.316  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =    0.6   Deviance explained = 61.9%
## fREML = 1378.1  Scale est. = 0.073136  n = 8389
```

The interaction between *SS* and *Frequency* is significant. Because this is expected to be bilingual-specific interaction, we then request different wiggly surfaces, one for each of the two levels of *FirstLanguage* using the by-directive. The main effect of *FirstLanguage* should be included so that two wiggly surfaces can be set at appropriate places on the y-axis.

```
miwa.gam3 = bam(invRT ~
                  FirstLanguage +
                  te(SS, Frequency, by=FirstLanguage) +
                  s(Trial, by=FirstLanguage) +
                  s(Trial, Participant, bs = "fs", m=1) +
                  s(Participant, Frequency, bs = "re") +
                  s(Word, bs = "re"),
                data = miwa, discrete=TRUE)
summary(miwa.gam3)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ FirstLanguage + te(SS, Frequency, by = FirstLanguage) +
##     s(Trial, by = FirstLanguage) + s(Trial, Participant, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Word, bs = "re")
##
## Parametric coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.92434    0.04625  -41.60  < 2e-16 ***
## FirstLanguageJapanese  0.46499    0.06567    7.08 1.56e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                           edf  Ref.df      F  p-value
## te(SS,Frequency):FirstLanguageEnglish   3.812   4.069 14.451 7.59e-12 ***
## te(SS,Frequency):FirstLanguageJapanese  5.609   6.357 20.179  < 2e-16 ***
## s(Trial):FirstLanguageEnglish           1.237   1.297  8.656 0.003088 **
## s(Trial):FirstLanguageJapanese          5.073   5.870 12.236 2.41e-13 ***
## s(Trial,Participant)                  190.127 340.000 11.880  < 2e-16 ***
## s(Frequency,Participant)               17.019  36.000  0.902 0.000787 ***
## s(Word)                               171.215 224.000  3.251  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.601   Deviance explained =   62%
## fREML =   1371  Scale est. = 0.072991  n = 8389
```

The inclusion of *FirstLanguage* in the interaction led to an improved model fit.

```
miwa.gam2ml = bam(invRT ~
                    FirstLanguage +
                    te(SS, Frequency) +
                    s(Trial, by=FirstLanguage) +
                    s(Trial, Participant, bs = "fs", m=1) +
                    s(Participant, Frequency, bs = "re") +
                    s(Word, bs = "re"),
                  data = miwa, discrete=TRUE, method = "ML")

## Warning in bam(invRT ~ FirstLanguage + te(SS, Frequency) + s(Trial, by =
## FirstLanguage) + : discretization only available with fREML

miwa.gam3ml = bam(invRT ~
                    FirstLanguage +
                    te(SS, Frequency, by=FirstLanguage) +
                    s(Trial, by=FirstLanguage) +
                    s(Trial, Participant, bs = "fs", m=1) +
                    s(Participant, Frequency, bs = "re") +
                    s(Word, bs = "re"),
                  data = miwa, discrete=TRUE, method = "ML")

## Warning in bam(invRT ~ FirstLanguage + te(SS, Frequency, by =
## FirstLanguage) + : discretization only available with fREML

compareML(miwa.gam2ml, miwa.gam3ml)

## miwa.gam2ml: invRT ~ FirstLanguage + te(SS, Frequency) + s(Trial, by =
FirstLanguage) +
##     s(Trial, Participant, bs = "fs", m = 1) + s(Participant,
##     Frequency, bs = "re") + s(Word, bs = "re")
##
## miwa.gam3ml: invRT ~ FirstLanguage + te(SS, Frequency, by = FirstLanguage)
+
##     s(Trial, by = FirstLanguage) + s(Trial, Participant, bs = "fs",
##     m = 1) + s(Participant, Frequency, bs = "re") + s(Word, bs = "re")
##
## Chi-square test of ML scores
## -----
##         Model    Score Edf Difference    Df   p.value Sig.
## 1 miwa.gam2ml 1363.192  15
## 2 miwa.gam3ml 1351.859  20     11.333 5.000 3.911e-04  ***
##
## AIC difference: 8.27, model miwa.gam3ml has lower AIC.
```

The regression surfaces are significant for both natives and non-natives. Unexpectedly, monolingual participants were also sensitive to some aspect of *SS*. To test how natives and non-natives differ, we compute a difference surface.

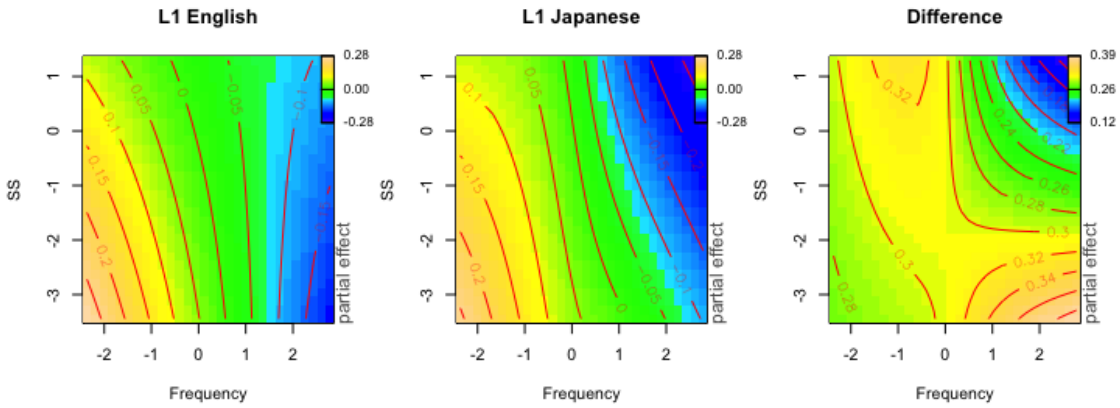
```
miwa$FirstLanguageNum = ifelse(miwa$FirstLanguage == "Japanese", 1,0) #
Japanese is 1
miwa.gam3num = bam(invRT ~
```

```
                    te(SS ,Frequency) +
                    te(SS, Frequency, by=FirstLanguageNum) +
                    s(Trial) +
                    s(Trial, by=FirstLanguageNum) +
                    s(Trial, Participant, bs = "fs", m=1) +
                    s(Participant, Frequency, bs = "re") +
                    s(Word, bs = "re"),
                  data = miwa, discrete=TRUE)
summary(miwa.gam3num)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## invRT ~ te(SS, Frequency) + te(SS, Frequency, by = FirstLanguageNum) +
##     s(Trial) + s(Trial, by = FirstLanguageNum) + s(Trial, Participant,
##     bs = "fs", m = 1) + s(Participant, Frequency, bs = "re") +
##     s(Word, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.92553    0.04653  -41.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                        edf  Ref.df      F  p-value
## te(SS,Frequency)                     4.543   4.803 12.498 2.19e-11 ***
## te(SS,Frequency):FirstLanguageNum    6.667   7.509  5.069 6.04e-06 ***
## s(Trial)                             1.659   1.809  6.695 0.005412 **
## s(Trial):FirstLanguageNum            4.845   5.607  3.987 0.000803 ***
## s(Trial,Participant)               189.875 340.000 11.871  < 2e-16 ***
## s(Frequency,Participant)            17.011  36.000  0.901 0.000793 ***
## s(Word)                            170.227 224.000  3.205  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 676/677
## R-sq.(adj) =  0.601   Deviance explained =   62%
## fREML = 1369.7  Scale est. = 0.073001  n = 8389
```

The difference surface is then visualized. It indicates that, for bilingual participants, *SS* afforded more facilitation compared to monolinguals, specifically for high-frequency words, as can be seen in the right half of the difference surface. The figures below should be read like a topographic contour map, with colder colors indicating shorter RTs.

```
par(mfrow=c(1,3))
zlim= c(-0.28, 0.28)
```

```
pvisgam(miwa.gam3, select=1, view = c("Frequency","SS"), main = "L1 English",
plot.type = "contour", color="topo",zlim=zlim)

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageEnglish"

pvisgam(miwa.gam3, select=2, view = c("Frequency","SS"), main = "L1
Japanese", plot.type = "contour", color="topo",zlim=zlim)

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageJapanese"

pvisgam(miwa.gam3num, select=2, view = c("Frequency","SS"), main =
"Difference")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"
```



6. Quantile generalized additive mixed model (QGAMM)

The QGAMM is a recent extension of the GAMM that enables constructing models for any desired quantile of the response variable's distribution. In this way, it becomes possible to clarify whether the effect of a predictor is variable rather than constant across the distribution of the response variable. When applied to RT data, at the 0.1 decile, one can study early effects, at the median one can study the central tendency, and at the 0.9 decile, late effects. The QGAMM can also be applied to other data like test scores.

When fitting a model for a specific quantile, all datapoints in the distribution are taken into account, but datapoints far away from a given quantile are given less weight. Note that quantile regression does not fit subsets of data, but always considers all datapoints jointly.

We fit QGAMMs to Dutch-English bilinguals' lexical decision RTs to test whether the effects of *OS* was constant across the distribution of RTs. The QGAMM can be achieved with the `qgam()` function (Fasiolo, Goude, Nedellec, & Wood, 2017). Here, the `mqgam()` function is used to fit models at multiple quantiles. The `qdo()` function is used to retrieve an output at one particular decile. In the following example, the random-effects structure is kept simple as QGAMMs require substantial computation time for complex random effects structure.

```
qntls = seq(0.1, 0.9, by = 0.2)
dijkstra.qgam = mqgam(RT ~
```

```
                        s(OS, k = 5)+
                        s(Frequency) +
                        s(Trial) +
                        s(Participant, bs = "re") ,
                      data =dijkstra, qu=qntls)
```

```
## Estimating learning rate. Each dot corresponds to a loss evaluation.
## qu = 0.5.............done
## qu = 0.3...........done
## qu = 0.7............done
## qu = 0.1............done
## qu = 0.9............done
```

```
summary(qdo(dijkstra.qgam, qu = 0.1))
```

```
##
## Family: elf
## Link function: identity
##
## Formula:
## RT ~ s(OS, k = 5) + s(Frequency) + s(Trial) + s(Participant,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   445.36      11.81    37.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                    edf Ref.df   Chi.sq p-value
## s(OS)            3.293  3.709   59.540 3.1e-12 ***
## s(Frequency)     1.026  1.052   83.336 < 2e-16 ***
## s(Trial)         1.954  2.437    3.145   0.228
## s(Participant) 19.743 20.000 1535.118 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.197   Deviance explained = 78.6%
## -REML =  21613  Scale est. = 1         n = 3535
```

```
summary(qdo(dijkstra.qgam, qu = 0.3))
```

```
##
## Family: elf
## Link function: identity
##
## Formula:
## RT ~ s(OS, k = 5) + s(Frequency) + s(Trial) + s(Participant,
##     bs = "re")
##
```

```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   493.97      12.97   38.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df   Chi.sq p-value
## s(OS)           3.242  3.670   91.203  <2e-16 ***
## s(Frequency)    6.241  7.410  162.688  <2e-16 ***
## s(Trial)        1.003  1.007    0.952   0.331
## s(Participant) 19.802 20.000 2012.964  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.227   Deviance explained = 45.8%
## -REML =  21501  Scale est. = 1         n = 3535
```

```
summary(qdo(dijkstra.qgam, qu = 0.5))
```

```
##
## Family: elf
## Link function: identity
##
## Formula:
## RT ~ s(OS, k = 5) + s(Frequency) + s(Trial) + s(Participant,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   535.06      14.58    36.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df   Chi.sq p-value
## s(OS)           2.856  3.327   87.528  <2e-16 ***
## s(Frequency)    6.407  7.557  151.953  <2e-16 ***
## s(Trial)        1.134  1.255    1.178   0.287
## s(Participant) 19.734 20.000 1489.904  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.248   Deviance explained = 24.5%
## -REML =  21871  Scale est. = 1         n = 3535
```

```
summary(qdo(dijkstra.qgam, qu = 0.7))
```

```
##
## Family: elf
## Link function: identity
```

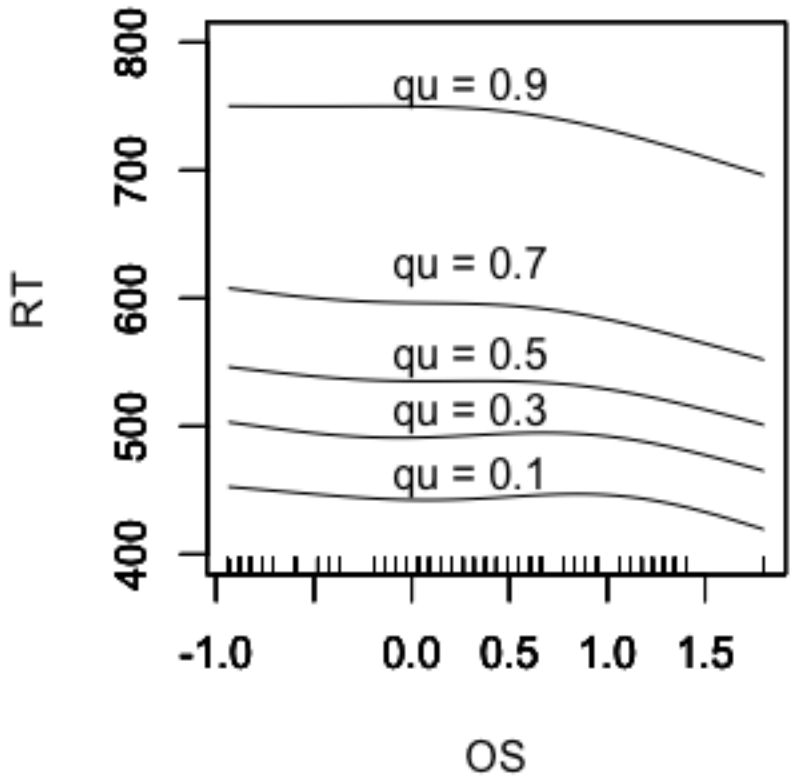
```
##
## Formula:
## RT ~ s(OS, k = 5) + s(Frequency) + s(Trial) + s(Participant,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   593.74      18.73   31.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df   Chi.sq p-value
## s(OS)           2.569  3.025   71.197   3e-15 ***
## s(Frequency)    3.782  4.720  135.355  <2e-16 ***
## s(Trial)        1.246  1.451    0.082   0.837
## s(Participant) 19.663 20.000 1065.851  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.27   Deviance explained = 21.8%
## -REML =  22714  Scale est. = 1         n = 3535
```

```
summary(qdo(dijkstra.qgam, qu = 0.9))
```

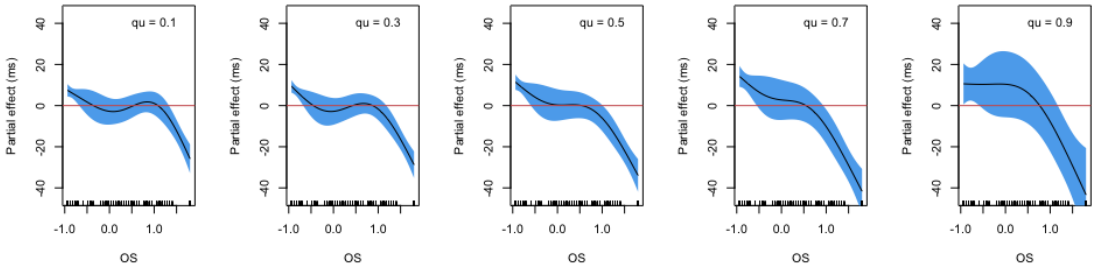
```
##
## Family: elf
## Link function: identity
##
## Formula:
## RT ~ s(OS, k = 5) + s(Frequency) + s(Trial) + s(Participant,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   739.63      33.36   22.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df  Chi.sq p-value
## s(OS)           2.053  2.441  14.607 0.00111 **
## s(Frequency)    4.387  5.421 110.779 < 2e-16 ***
## s(Trial)        1.051  1.100   5.683 0.02250 *
## s(Participant) 19.512 20.000 559.008 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.107   Deviance explained =   55%
## -REML =  25310  Scale est. = 1         n = 3535
```

The effect of *OS* is visualized at different deciles of the RT distribution.

```
par(mfrow = c(1,1))
ylim = c(400,800)
plot(qdo(dijkstra.qgam, qu = 0.1), select = 1, shift =
coef(qdo(dijkstra.qgam, qu = 0.1))[1], se=F, ylim = ylim, ylab = "RT");
text(0.3, 460, "qu = 0.1")
par(new = T)
plot(qdo(dijkstra.qgam, qu = 0.3), select = 1, shift =
coef(qdo(dijkstra.qgam, qu = 0.3))[1], se=F, ylim = ylim, ylab = "", rug = F,
xlab = ""); text(0.3, 510, "qu = 0.3")
par(new = T)
plot(qdo(dijkstra.qgam, qu = 0.5), select = 1, shift =
coef(qdo(dijkstra.qgam, qu = 0.5))[1], se=F, ylim = ylim, lwd = 1, ylab = "",
rug = F, xlab = ""); text(0.3, 555, "qu = 0.5")
par(new = T)
plot(qdo(dijkstra.qgam, qu = 0.7), select = 1, shift =
coef(qdo(dijkstra.qgam, qu = 0.7))[1], se=F, ylim = ylim, ylab = "", rug = F,
xlab = ""); text(0.3, 625, "qu = 0.7")
par(new = T)
plot(qdo(dijkstra.qgam, qu = 0.9), select = 1, shift =
coef(qdo(dijkstra.qgam, qu = 0.9))[1], se=F, ylim = ylim, ylab = "", rug = F,
xlab = ""); text(0.3, 765, "qu = 0.9")
```




```
par(mfrow = c(1,5))
ylim = c(-45,45)
plot(qdo(dijkstra.qgam, qu = 0.1), select = 1, ylim = ylim, ylab = "Partial
effect (ms)", shade = T, shade.col="steelblue2");abline(h=0,
col="indianred"); text(1,40, "qu = 0.1")
plot(qdo(dijkstra.qgam, qu = 0.3), select = 1, ylim = ylim, ylab = "Partial
effect (ms)", shade = T, shade.col="steelblue2");abline(h=0,
col="indianred"); text(1,40, "qu = 0.3")
plot(qdo(dijkstra.qgam, qu = 0.5), select = 1, ylim = ylim, ylab = "Partial
effect (ms)", shade = T, shade.col="steelblue2");abline(h=0,
col="indianred"); text(1,40, "qu = 0.5")
plot(qdo(dijkstra.qgam, qu = 0.7), select = 1, ylim = ylim, ylab = "Partial
effect (ms)", shade = T, shade.col="steelblue2");abline(h=0,
col="indianred"); text(1,40, "qu = 0.7")
plot(qdo(dijkstra.qgam, qu = 0.9), select = 1, ylim = ylim, ylab = "Partial
effect (ms)", shade = T, shade.col="steelblue2");abline(h=0,
col="indianred"); text(1,40, "qu = 0.9")
```

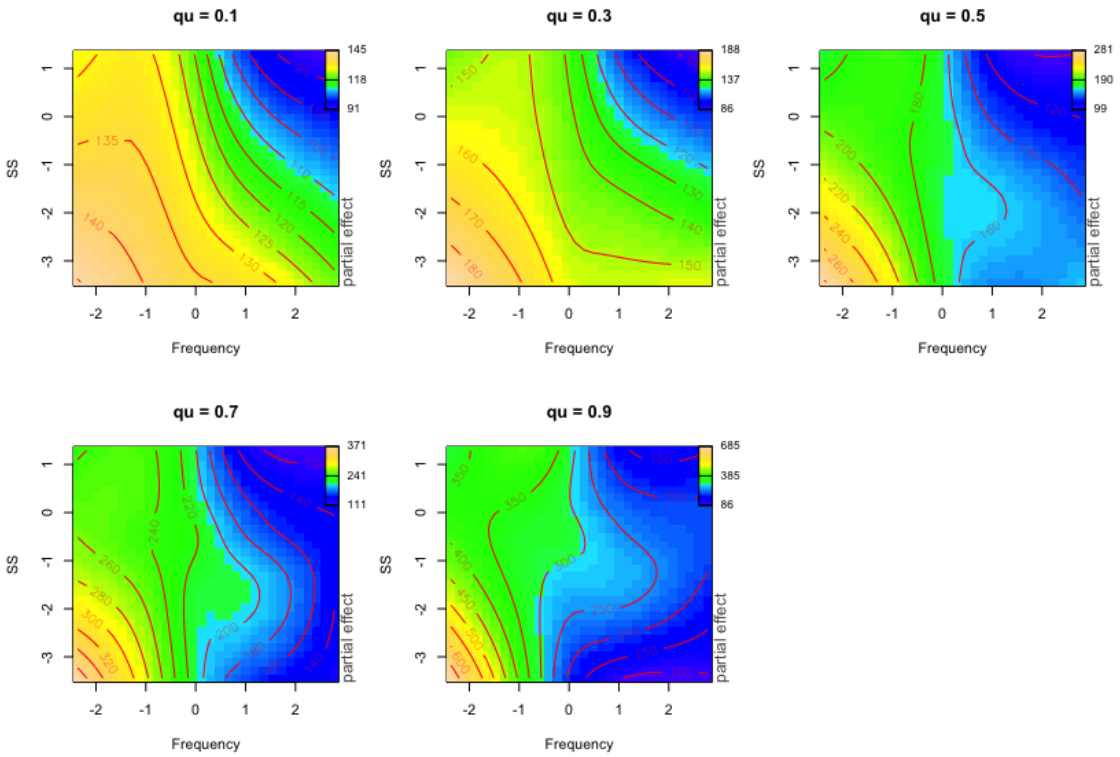


The nonlinear effect of *OS* observed in the GAMM (i.e., strong facilitation driven by identical cognates) was also observed in the QGAMMs, even in lower deciles. We can infer that *OS* indeed contributes already at the earliest stages of word recognition.

A QGAMM can also be used to track the time-course of an interactive effect of two (or more) predictors. In the GAMM analysis, the interaction between *SS* and *Frequency* was observed. We fit QGAMMs to see whether this interaction remains constant across the distribution of RTs.

```
load("Data/miwa.rda")
miwa$FirstLanguageNum = ifelse(miwa$FirstLanguage == "Japanese", 1,0)
miwa.qgam = mqgam(RT ~
                    te(SS, Frequency)+
                    te(SS, Frequency, by= FirstLanguageNum) +
                    s(Trial) +
                    s(Participant,  bs = "re"),
                  data =miwa, qu=qntls)

## Estimating learning rate. Each dot corresponds to a loss evaluation.
## qu = 0.5..........done
## qu = 0.3...........done
## qu = 0.7.............done
## qu = 0.1.................done
## qu = 0.9........done

par(mfrow = c(2,3))
pvisgam(qdo(miwa.qgam, qu = 0.1), select=2, view = c("Frequency","SS"), main
= "qu = 0.1", plot.type = "contour", color="topo")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"

pvisgam(qdo(miwa.qgam, qu = 0.3), select=2, view = c("Frequency","SS"), main
= "qu = 0.3", plot.type = "contour", color="topo")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"

pvisgam(qdo(miwa.qgam, qu = 0.5), select=2, view = c("Frequency","SS"), main
= "qu = 0.5", plot.type = "contour", color="topo")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"
```

```
pvisgam(qdo(miwa.qgam, qu = 0.7), select=2, view = c("Frequency","SS"), main
= "qu = 0.7", plot.type = "contour", color="topo")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"

pvisgam(qdo(miwa.qgam, qu = 0.9), select=2, view = c("Frequency","SS"), main
= "qu = 0.9", plot.type = "contour", color="topo")

## [1] "Tensor(s) to be plotted: te(SS,Frequency):FirstLanguageNum"
```



The regression surface at the 0.1 and 0.3 deciles are comparable to the surface estimated by the GAMM for the mean. However, the shape of the interaction surface changed for higher deciles; here, for words with low semantic similarity, the word frequency effect was particularly large, whereas the effect of semantic similarity was present only for low word frequency. This pattern of results may reflect a response strategy optimized for difficult stimuli eliciting long RTs, such that familiarity with a word dominated lexicality decisions, with semantics coming into play only for the least familiar words. It should be warned that, in the above figure, the zlimits change across the panels.

For interested readers, there are various other applications of GAMMs in language studies (see Chuang, Fon, Papakyritsis, & Baayen, 2020 for various aspects of Chinese including tone contours, Hendrix, Bolger, & Baayen, 2017 for an analysis of ERP data, Murakami, 2016 for an analysis of second language development data, van Rij, Hendriks, van Rijn, Baayen, & Wood, 2019 for an analysis of pupilometry data, and Wieling, 2018 for an analysis of phonetic data).

7. References

- Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, *59*, 390-412.
- Chuang, Y. Y., Fon, J., Papakyritsis, I. & Baayen, R. H. (2020). Analyzing phonetic data with generalized additive mixed models. In M. J. Ball (Ed.), *Handbook of clinical phonetics*. London: Routledge.
- Dijkstra, T., Miwa, K., Brummelhuis, B., Sappeli, M., & Baayen, R. H. (2010). How cross-language similarity and task demands affect cognate recognition. *Journal of Memory and Language*, *62*, 284-301.
- Dijkstra, T., & van Heuven, W. J. B. (2002). The architecture of the bilingual word recognition system: From identification to decision. *Bilingualism: Language and Cognition*, *5*, 175–197.
- Fasiolo M., Goude Y., Nedellec R. and Wood S. N. (2017). *Fast calibrated additive quantile regression*. URL: https://arxiv.org/abs/1707.03307
- Hendrix, P., Bolger, P., & Baayen, H. (2017). Distinct ERP signatures of word frequency, phrase frequency, and prototypicality in speech production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *43*, 128–149.
- Miwa, K., Dijkstra, T., Bolger, P., & Baayen, R. H. (2014). Reading English with Japanese in mind: Effects of frequency, phonology, and meaning in different-script bilinguals. *Bilingualism: Language and Cognition*,*17*, 445-463.
- Murakami, A. (2016). Modeling systematicity and individuality in nonlinear second language development: The case of English grammatical morphemes. *Language Learning*, *66*, 834-871.
- Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in S and S-PLUS*. New York: Springer.
- van Rij, J., Hendriks, P., van Rijn, H., Baayen, R. H., & Wood, S. N. (2019). Analyzing the time course of pupilometric data. *Trends in Hearing*, *23*, 1-22.
- van Rij, J., Wieling, M., Baayen R., 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.
- Wood, S. N. (2012). On p-values for smooth components of an extended generalized additive model. *Biometrika*, *100*, 221-228.
- Wood, S. N. (2013). On p-values for smooth components of an extended generalized additive model. *Biometrika*, *100*, 221-228.
- Wood, S. N. (2017). *Generalized additive models: An introduction with R* (2nd edition). Chapman and Hall/CRC.