Cognitive influences in language evolution: Supporting Materials

Padraic Monaghan & Seán Roberts

September 11, 2018

Contents

1 Introduction

These are the supporting materials for Monaghan & Roberts (2018). All data and scripts are available in a github repository:
https://github.com/seannyD/BorrowingFreqAoA .

2 English data (study 1)

Cognitive influences in language evolution: English data

Contents

Introduction

This is the model code for Monaghan & Roberts, "Cognitive influences in language evolution: Psycholinguistic predictors of loan word borrowing". It takes data from the WOLD database of borrowing for English and tries to predict whether a word has been borrowed or not according to various psycholinguitic measures.

The main fields in the data frame are:

- word: Orthographic form
- borrowing: variable from WOLD indicating level of evidence for borrowing:
- 1 = definately borrowed
- 5 = no evidence of borrowing
- bor15: Conversion of the WOLD borrowing variable into a numeric (0 = not borrowed, 1 = borrowed)
- phonology: Phonological form
- phonlength: Number of segments in the phonological form
- AoA: Age of acquisition ratings from Kuperman, Stadthagen-Gonzalez, and Brysbaert (2012).
- AoA_obj: Objective, test-based age of acuqisition from Brysbaert & Biemiller (2017)
- subtlexzipf: Log frequency of word from the SUBTLEX database
- conc: Concreteness ratings from Brysbaert, Warriner, & Kuperman (2014)
- cat: Dominant part of speech according to SUBTLEX.
- age_oldest, age_youngest: Dates from WOLD indicating estiamte of data of entry into English.

- age_oldest_num, age_youngest_num, age: Conversions into numeric year values for oldest, youngest and average estimate.
- source.language: Source language according to WOLD.
- source: Source language, made more specific by PM.
- source.word: the source word that was borrowed
- source.language.mean.word.length: Mean word length of Swadesh list words in the source language (from ASJP database).
- source.language.word.freq: Frequency of words as long as the source word in the source language (estimated from ASJP)
- effect: Type of transition (insertion, coexistence, replacement)
- modern.english, middle.english, old.english: form of the word in various stages of English
- old.english.length: length of the word in old English

A note on EDF values and random effects

The Estimated Degrees of Freedom (EDF) is an indication of how non-linear a smooth term is (higher = less linear). It is intended as a diagnostic measure of the shape of the curve, rather than a value used in estimating significance. The EDF is not the same as a simple polynomial curve's degree. Instead of a single polynomial curve, each smooth term is a collection of underlying basis functions (simpler curves). When each basis function is weighted by a coefficient, they add up to fit the data. The model attempts to find a collection of basis functions and weighting coefficients that add up to fit the data. Smaller collections of basis functions (simpler models) are preferred and larger (more complex) collections are penalised.

When the model converges on a solution, each smooth term is a collection of simpler curves. Each curve might have a polynomial degree, but it is also useful to have an estimate of the linearity/non-linearity of the whole smooth term. This is what the EDF provides: an EDF of 1 indicates a linear relationship, and higher values indicate more non-linear relationship. The definition of EDF in the implementaion we use is described in Wood (2008):

> "Associated with each smooth function is one or more measures of function 'wiggliness' $\beta_T^j \tilde{S}_j \beta_j$ where $\tilde{S}_j$ is a matrix of known coefficients. Typically the wiggliness measure evaluates something like the univariate spline penalty."

That is, an EDF value is a combination of non-linearity measures of the basis functions, weighted by the weighting coefficient of each basis function. So, in general, a curve with an EDF of around 2 will look like a quadratic curve, and an EDF of around 3 will look like a cubic curve. However, this does not have to be the case: a smooth term could have a strong linear term, and a very weak non-linear term. The EDF captures this possibility as a continuous value. The simplest way to actually assess the smooth term is to plot it.

Random effects in the GAM implementation we use are treated just like a smooth term with the identity matrix as the penalty coefficient matrix. When entering part of speech as a random (intercept) effect, coefficients are created for each part of speech, modelled as independent and identically distributed normal random variables. The values are defined as discrete points along a smooth function. So, just like in a mixed effects model, the probability of borrowing can be adjusted by a random intercept (the coefficients), e.g. the model can represent nouns as having a higher probability of borrowing, adjectives as slightly less probable and so on. Stronger differences between levels of the random effect would need be represented by more complex functions, which would be penalised (similar to how a linear mixed effect model penalises random effect coefficient estimates which deviate from a normal distribution). The EDF value for the random effects relates to the 'wiggliness' of these coefficients when plotted in a regular space. This makes the EDF difficult to interpret. A random effect where there were no differences between levels would have an EDF of 1 (a flat line), but it would also be 1 when there were consistent distances between each level. So a high EDF would indicate something like an imbalance in the distribution of coefficients. i.e. a few parts of speech are very likely to be borrowed, and most are very unlikely. This is in fact what we have, as shown in table 2 of the manuscript, and is consistent with previous studies of the effect of borrowing relating to grammatical category.

Load libraries

```r
library(mgcv)
library(sjPlot)
library(lattice)
library(ggplot2)
library(dplyr)
library(party)
library(lmtest)
library(gridExtra)
library(scales)
library(itsadug)
library(ggfortify)
library(factoextra)
library(gridExtra)
library(reshape2)
library(binom)

logit2per = function(X){
  return(exp(X)/(1+exp(X)))
}

rescaleGam = function(px, n, xvar, xlab="",breaks=NULL,xlim=NULL){
  y = logit2per(px[[n]]$fit)
  x = px[[n]]$x *attr(xvar,"scaled:scale") + attr(xvar,"scaled:center")
  se.upper = logit2per(px[[n]]$fit+px[[n]]$se)
  se.lower = logit2per(px[[n]]$fit-px[[n]]$se)
  dx = data.frame(x=x,y=y,ci.upper=se.upper,ci.lower=se.lower)
  plen = ggplot(dx, aes(x=x,y=y))+
    geom_ribbon(aes(ymin=ci.lower,ymax=ci.upper), alpha=0.3)+
    geom_line(size=0.5,linetype=3) +
    xlab(xlab)+
    ylab("Probability of borrowing")
  if(!is.null(breaks)){
    plen = plen + scale_x_continuous(breaks = breaks)
  }
  if(!is.null(xlim)){
   plen = plen + coord_cartesian(ylim = c(0,1),xlim=xlim)
  } else{
    plen = plen + coord_cartesian(ylim = c(0,1))
  }
  return(plen)
}
# Code for assessing significance of GAM slopes
source("GAM_derivaties.R")
```

Load data

```r
dataloan <- read.csv("../data/loanword12.csv",stringsAsFactors = F)
dataloan$bor15 <- ifelse(dataloan$borrowing==1,1, ifelse(dataloan$borrowing==5,0,NA))
dataloan$bor15.cat <- factor(dataloan$bor15)
```

Convert to numbers.

```
dataloan$subtlexzipf = as.numeric(dataloan$subtlexzipf)
dataloan$AoA = as.numeric(dataloan$AoA)
dataloan$conc = as.numeric(dataloan$conc)

aoaSD = sd(dataloan$AoA,na.rm = T)
aoaMean = mean(dataloan$AoA/aoaSD,na.rm=T)
dataloan$cat = factor(dataloan$cat)
```

Select only complete cases.

```
dataloan2 = dataloan[complete.cases(dataloan[,
               c("phonlength","AoA",
               "subtlexzipf", "cat",
               'conc','bor15')]),]
```

Scale and center:

```
dataloan2$AoAscale <- scale(dataloan2$AoA)

dataloan2$subtlexzipfscale <- scale(dataloan2$subtlexzipf)

phonlength.center = median(dataloan2$phonlength)
dataloan2$phonlengthscale <-
  dataloan2$phonlength - phonlength.center
phonlength.scale = sd(dataloan2$phonlengthscale)
dataloan2$phonlengthscale = dataloan2$phonlengthscale/phonlength.scale

attr(dataloan2$phonlengthscale,"scaled:scale") = phonlength.scale
attr(dataloan2$phonlengthscale,"scaled:center") = phonlength.center

dataloan2$concscale <- scale(dataloan2$conc)
conc.scale = attr(dataloan2$concscale,"scaled:scale")
conc.center = attr(dataloan2$concscale,"scaled:center")

dataloan2$cat = relevel(dataloan2$cat,"Noun")

dataloan2$AoA_objscaled = scale(dataloan2$AoA_obj)


dataloan2$source.language[dataloan2$bor15==0] = "English"
dataloan2$source.language = factor(dataloan2$source.language)

dataloan2$SLMWL = scale(log(dataloan2$source.language.mean.word.length))
dataloan2$SWF = scale(dataloan2$source.language.word.freq)
```

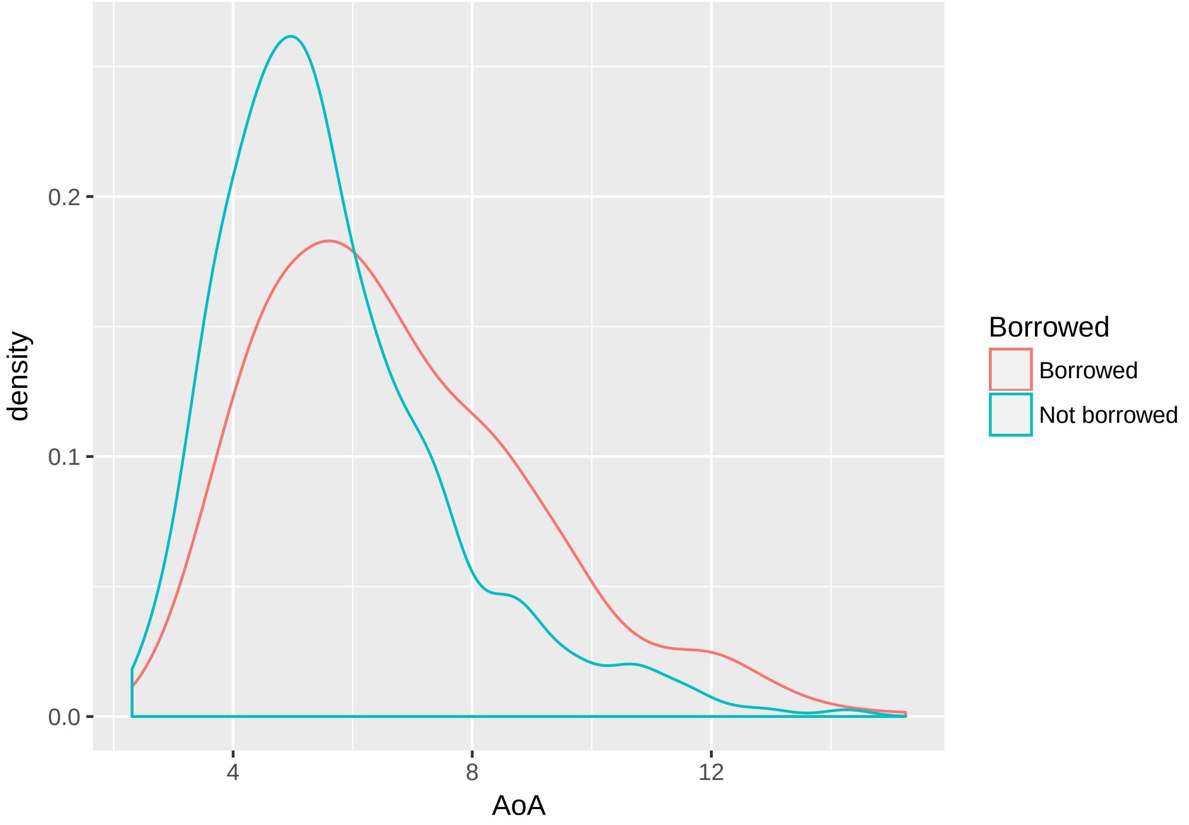
Identify Swadesh words:

```
swd = read.csv("../data/SwadeshConcepts.txt", header = F, stringsAsFactors = F)$V1
dataloan2$Swadesh = dataloan2$word %in% swd
```

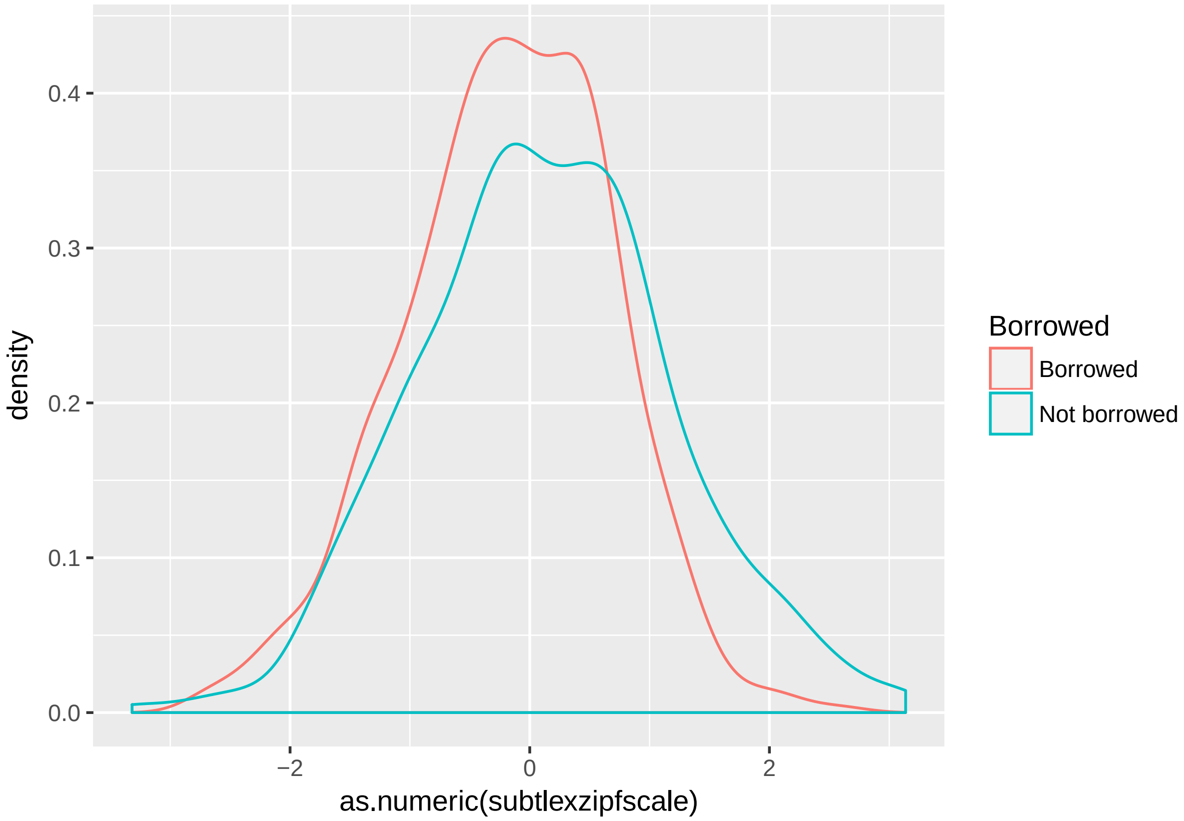
Plots

Raw data

```
dataloan2$Borrowed = c("Not borrowed", "Borrowed")[dataloan2$bor15+1]
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=AoA, colour=Borrowed)) +
  geom_density()
```

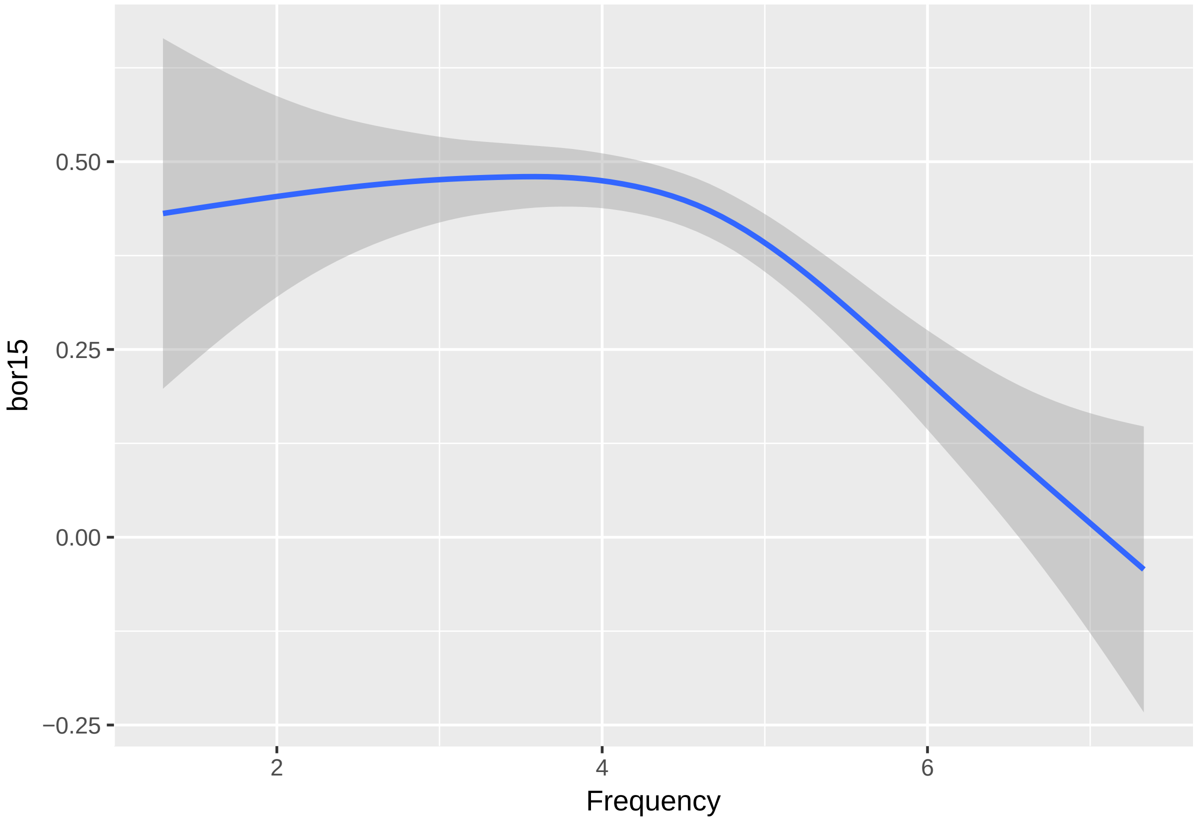


```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipfscale), colour=Borrowed)) +
  geom_density()
```



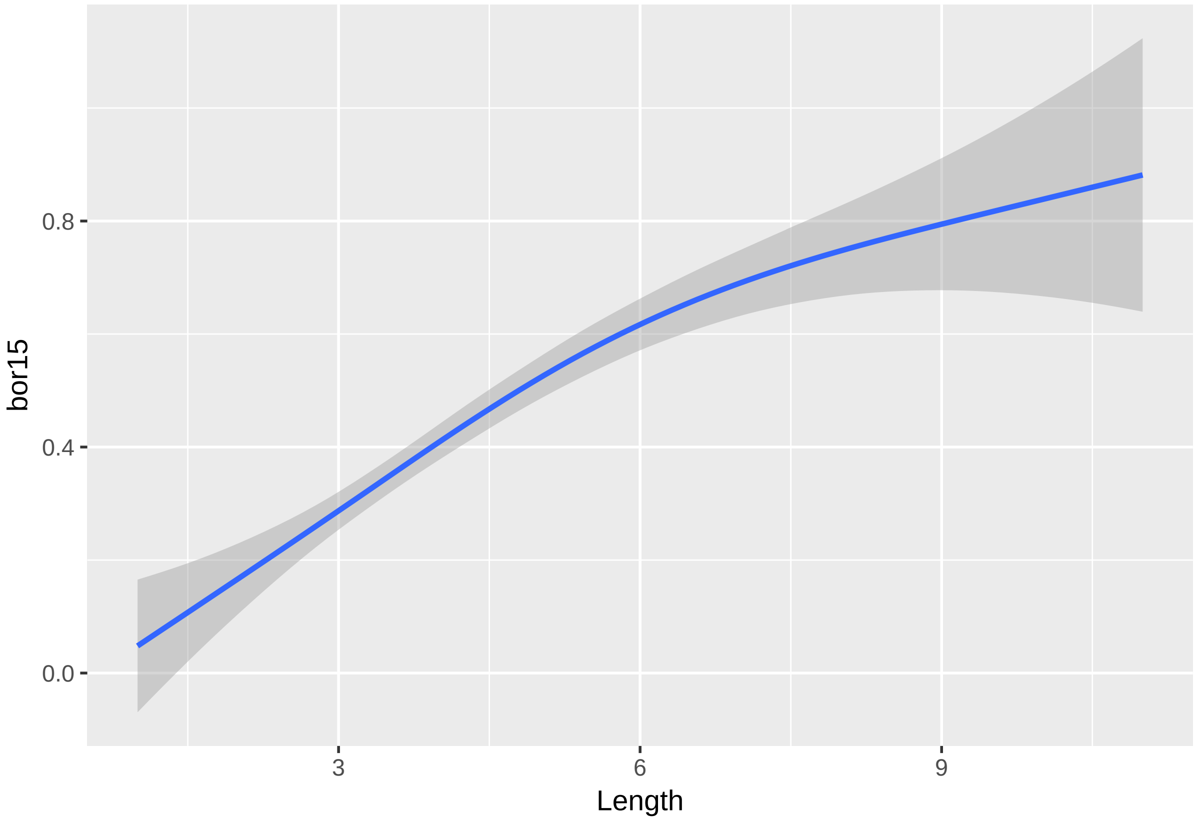
```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipf), y=bor15)) +
  stat_smooth()+
  xlab("Frequency")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(phonlength), y=bor15)) +
  stat_smooth() +
  xlab("Length")
```

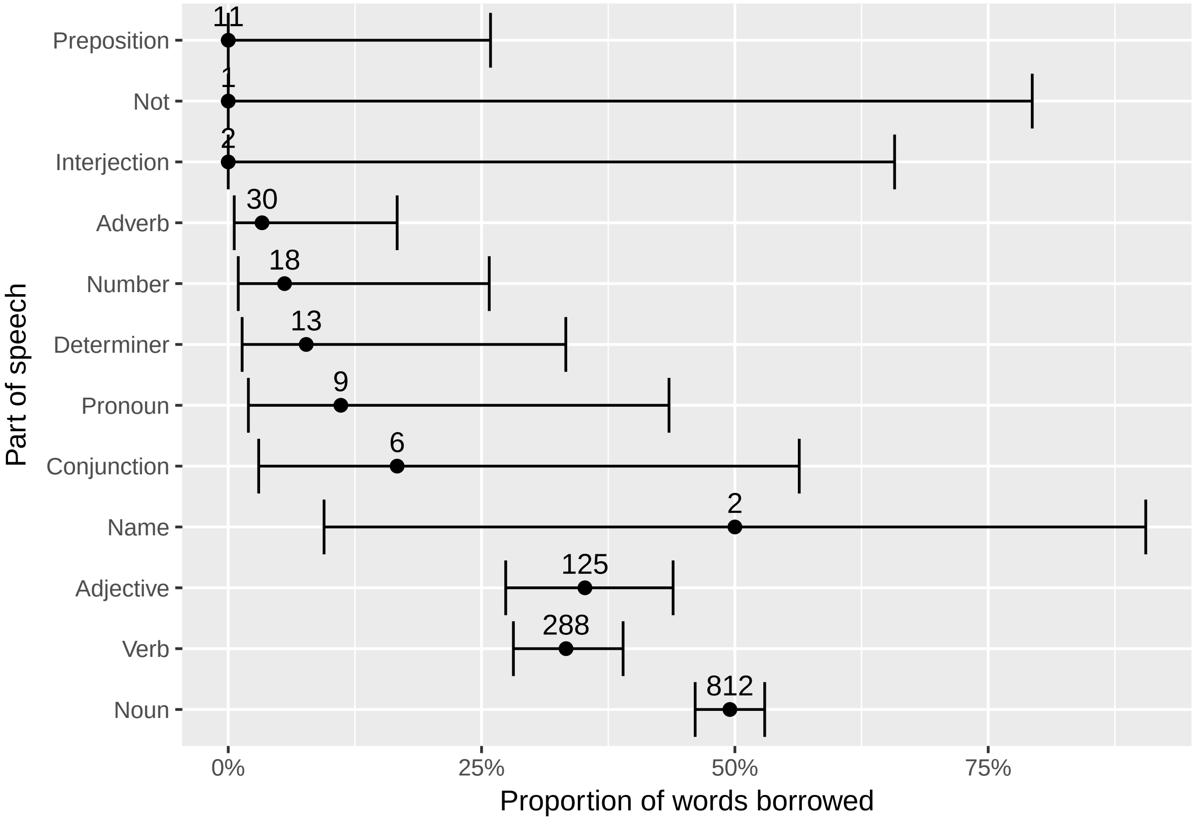
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
dataloan2$subtlexzipf.cat = cut(
  dataloan2$subtlexzipf,
  breaks = quantile(dataloan2$subtlexzipf,
                    prob=seq(0,1,length.out=4)),
  include.lowest = T)
```

Look at variation between parts of speech. We calculate the means, but the number of observations is very different for each category. We estimate confidence intervals around the mean with Wilson's binomial confidence interval method.

```
catx = data.frame(
  PoS = tapply(dataloan2$cat, dataloan2$cat, function(X){as.character(X[1])}),
  mean = tapply(dataloan2$bor15, dataloan2$cat, mean),
  n = tapply(dataloan2$bor15, dataloan2$cat, length),
  confint = binom.confint(
    tapply(dataloan2$bor15, dataloan2$cat, sum),
    tapply(dataloan2$bor15, dataloan2$cat, length),
    methods="wilson"
  )
)
catx = catx[order(catx$confint.lower, decreasing = T),]
catx$PoS = factor(catx$PoS, levels = catx[order(catx$confint.lower, decreasing = T),]$PoS)

posg = ggplot(catx, aes(x=mean, y=PoS)) +
  geom_point(size=2) +
  ylab("Part of speech") +
  xlab("Proportion of words borrowed")+
  scale_x_continuous(labels=percent_format()) +
  geom_text(aes(label=n), nudge_y=0.4) +
  geom_errorbarh(aes(xmin=confint.lower, xmax=confint.upper))
posg
```



```
pdf("../results/graphs/POS_Borrowing.pdf",
    width = 6,
    height = 4)
posg
dev.off()
```

```
## pdf
##   2
```

```
catx$mean= catx$mean*100
catx$confint.lower= catx$confint.lower*100
catx$confint.upper= catx$confint.upper*100
write.csv(catx[,c("PoS","mean",
                  'n','confint.lower','confint.upper')],
    "../results/English_POS_BorrowingProportions.csv",
    row.names = F)
```

GAM

```
m0 = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2,
    family='binomial')
```

```
summary(m0)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4908     0.4402  -3.386 0.000709 ***
## ---
## 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(phonlengthscale)      1.622e+00  2.036 32.336 1.12e-07 ***
## s(AoAscale)             1.000e+00  1.000 35.555 2.48e-09 ***
## s(subtlexzipfscale)     3.407e+00  4.328 32.599 2.74e-06 ***
## s(concscale)            2.680e+00  3.343  7.640   0.0728 .
## s(cat)                  5.878e+00 11.000 39.654 1.24e-08 ***
## s(cat,phonlengthscale)  1.191e+00 11.000  3.186   0.0626 .
## s(cat,AoAscale)         2.542e-06 11.000  0.000   0.7221
## s(cat,subtlexzipfscale) 8.010e-06 11.000  0.000   0.7499
## s(cat,concscale)        9.244e-06 11.000  0.000   0.7037
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.19   Deviance explained = 16.2%
## fREML = 1864.5  Scale est. = 1         n = 1317
```

Interactions

Test whether an interaction between AoA and frequency is warranted using likelihood ratio comparisons:

```
m1 = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(AoAscale,subtlexzipfscale),
    data = dataloan2,
    family='binomial')

lrtest(m0,m1)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(AoAscale,
##     subtlexzipfscale)
##      #Df  LogLik      Df  Chisq Pr(>Chisq)
## 1 19.758 -749.22
## 2 20.708 -748.38 0.95022 1.6707     0.1962
```

No significant improvement.

Test whether an interaction between AoA and length is warranted:

```
m2 = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(AoAscale,phonlengthscale),
    data = dataloan2,
    family='binomial')

lrtest(m0,m2)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(AoAscale,
##     phonlengthscale)
##      #Df  LogLik         Df Chisq Pr(>Chisq)
## 1 19.758 -749.22
## 2 19.758 -749.22 1.2566e-05     0  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `lrtest` function above suggests a significant improvement, but actually the log likelihood has not changed:

```
logLik(m0)
```

```
## 'log Lik.' -749.2182 (df=19.75776)
```

```
logLik(m2)
```

```
## 'log Lik.' -749.2182 (df=19.75777)
```

Therefore, there is no improvement and we should prefer the simpler model (without the interaction).

Test whether an interaction between Frequency and length is warranted:

```
m3 = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(subtlexzipfscale,phonlengthscale),
    data = dataloan2,
    family='binomial')

lrtest(m0,m3)
```

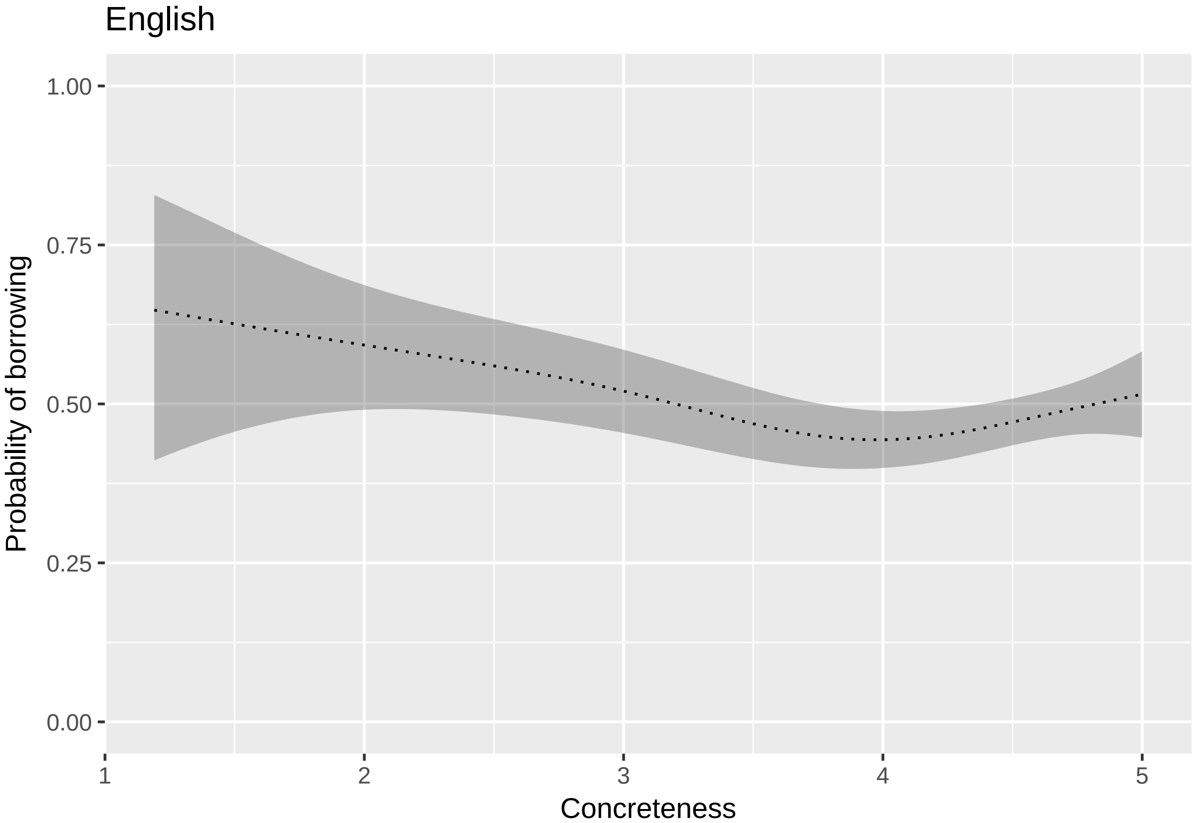
```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(subtlexzipfscale,
##     phonlengthscale)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 19.758 -749.22
## 2 22.651 -747.83 2.8934 2.7782     0.4271
```

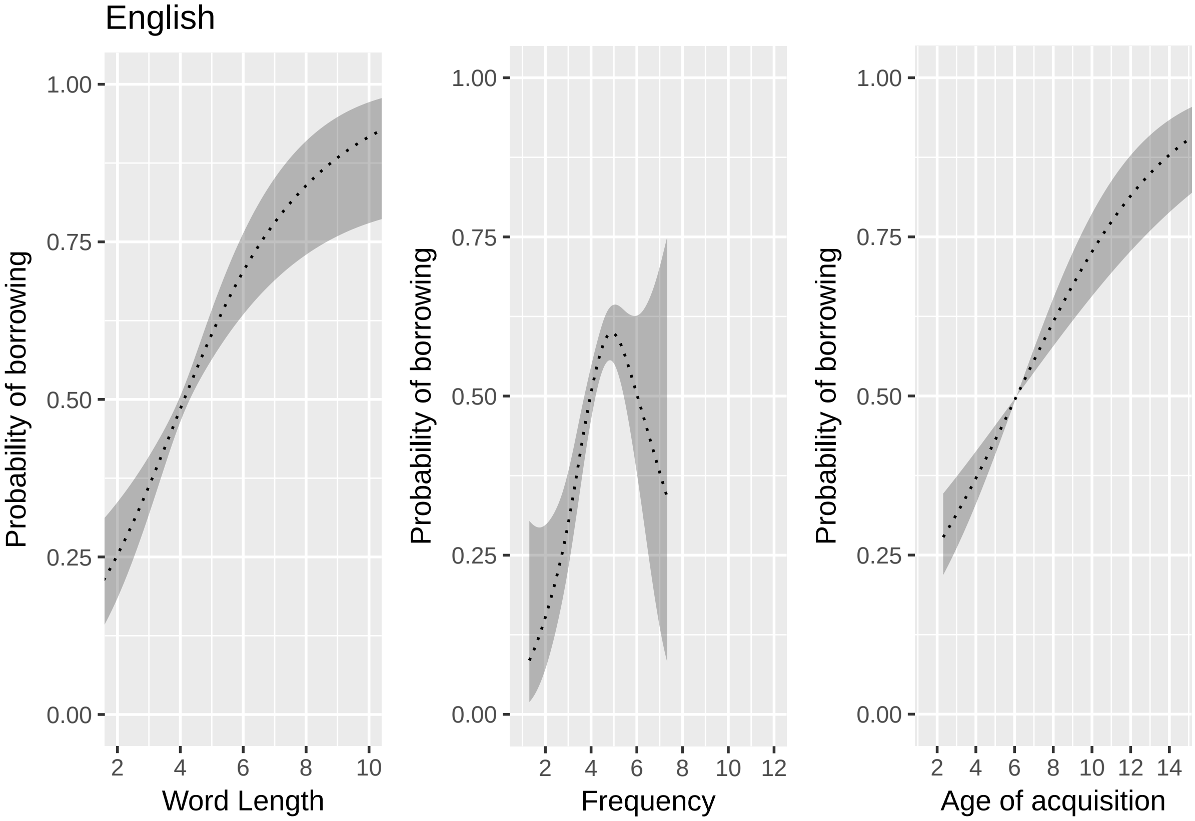
No significant improvement.

So no interactions are necessary.

Model plots

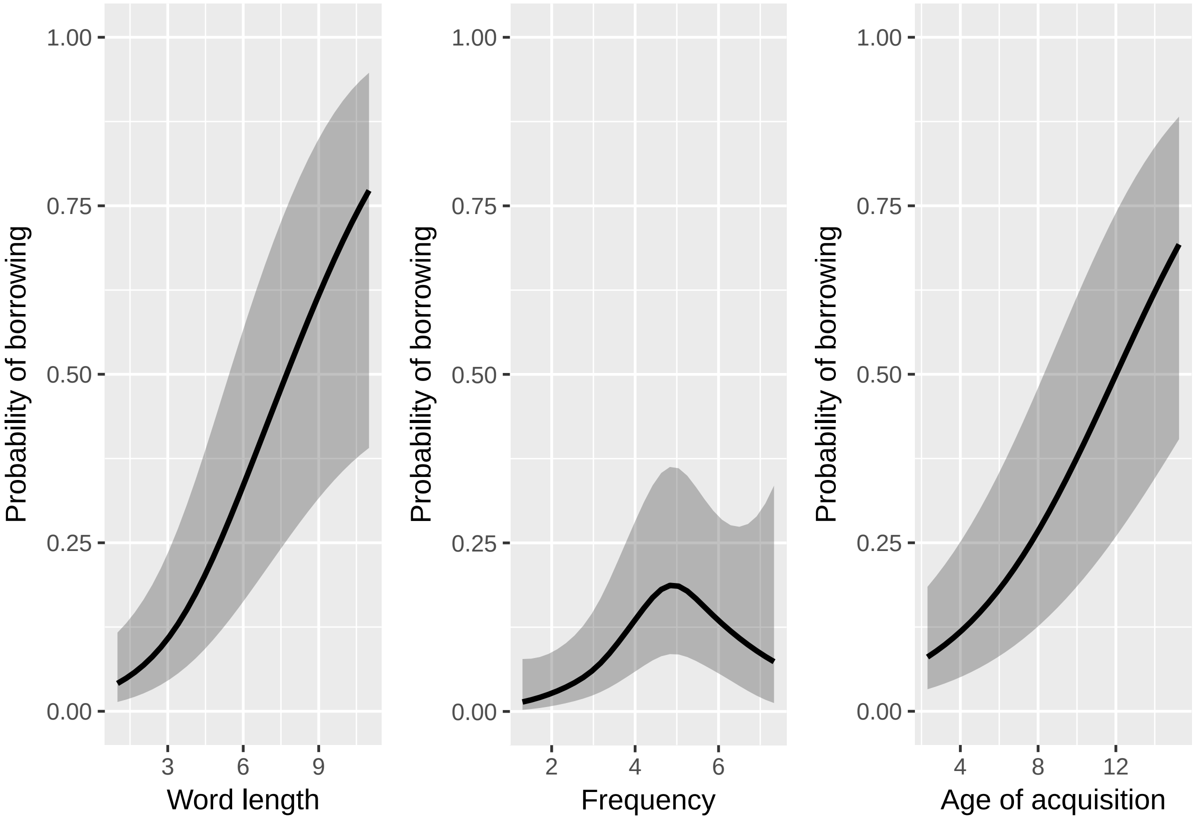
Plot the model estimates, changing the dependent scale to probability and the independent variables to their original scales. This code is hidden, but you can view it in the Rmd file.


English
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
Concreteness


English
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
Word Length
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
Frequency
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
14
Age of acquisition

```
## pdf
##    2
```

We can also plot the effects when removing the random effects (using the library `itsadug`). These are essentially the same, though not as easy to understand.

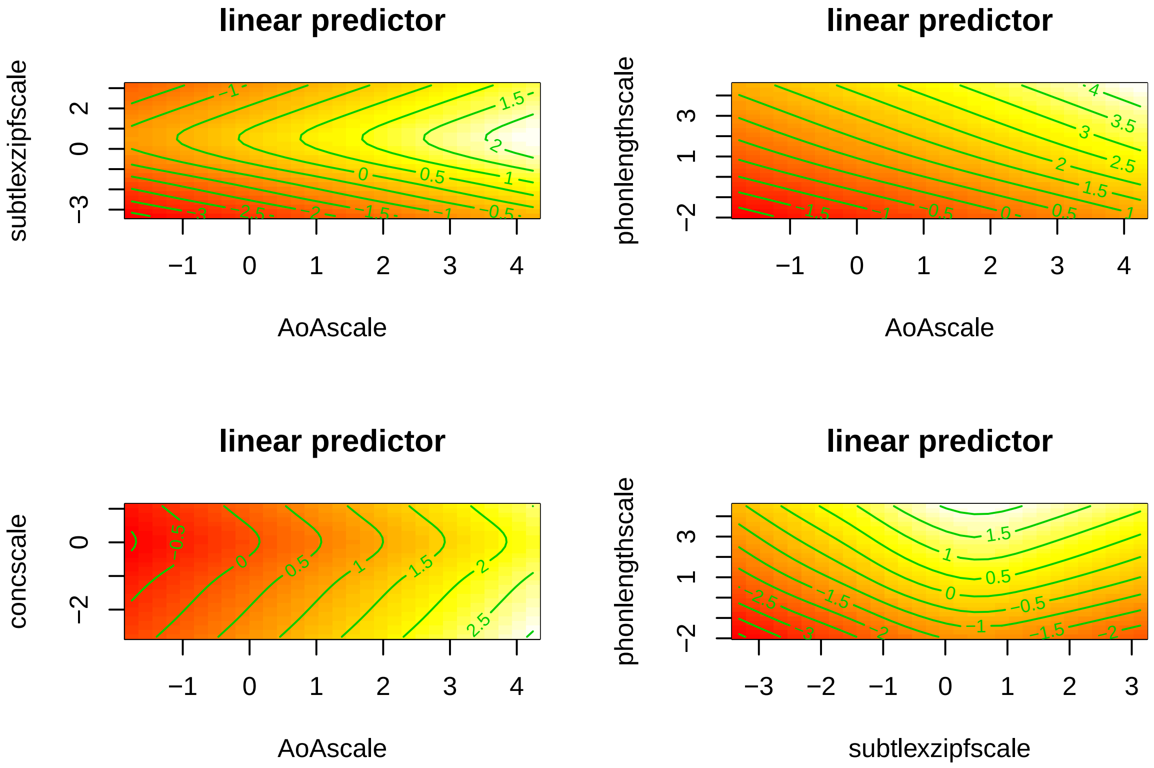


```
## pdf
##    2
```

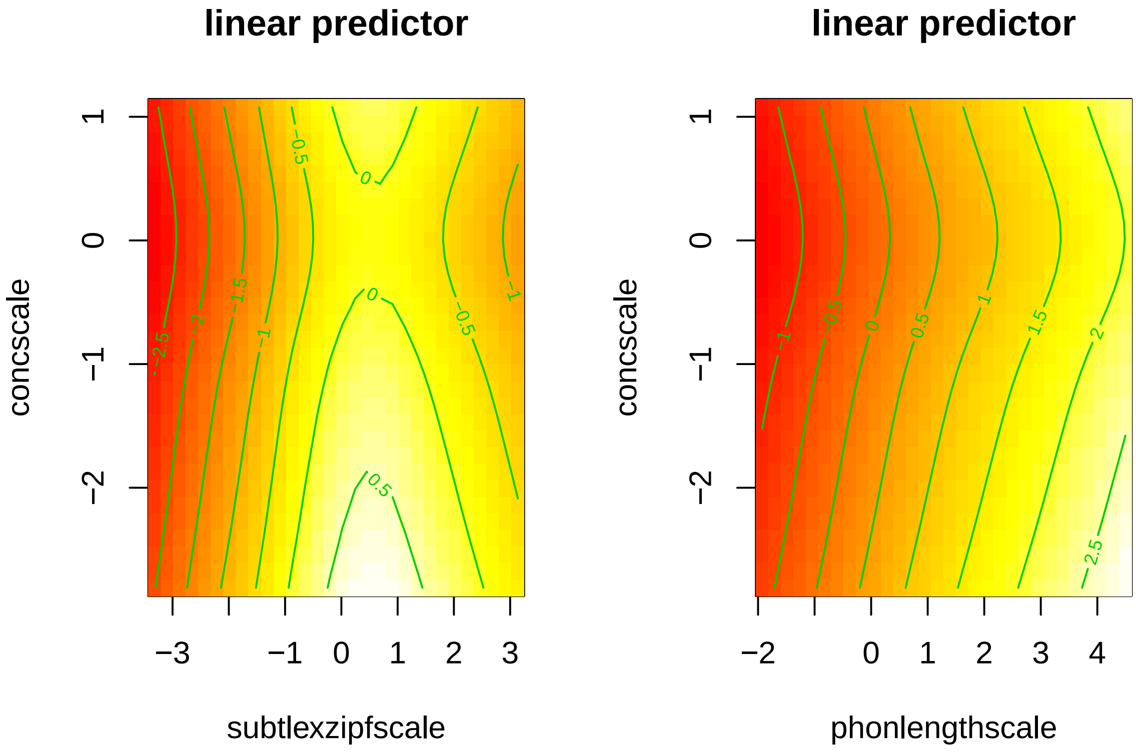
Contour plots

Visualise relationships between variables.

```
par(mfrow=c(2,2))
vis.gam(m0,view = c("AoAscale","subtlexzipfscale"),plot.type = "contour")
vis.gam(m0,view = c("AoAscale","phonlengthscale"),plot.type = "contour")
vis.gam(m0,view = c("AoAscale","concscale"),plot.type = "contour")
vis.gam(m0,view = c("subtlexzipfscale","phonlengthscale"),plot.type = "contour")
```



```
par(mfrow=c(1,2))
vis.gam(m0,view = c("subtlexzipfscale","concscale"),plot.type = "contour")
vis.gam(m0,view = c("phonlengthscale","concscale"),plot.type = "contour")
```



```r
par(mfrow=c(1,1))
```

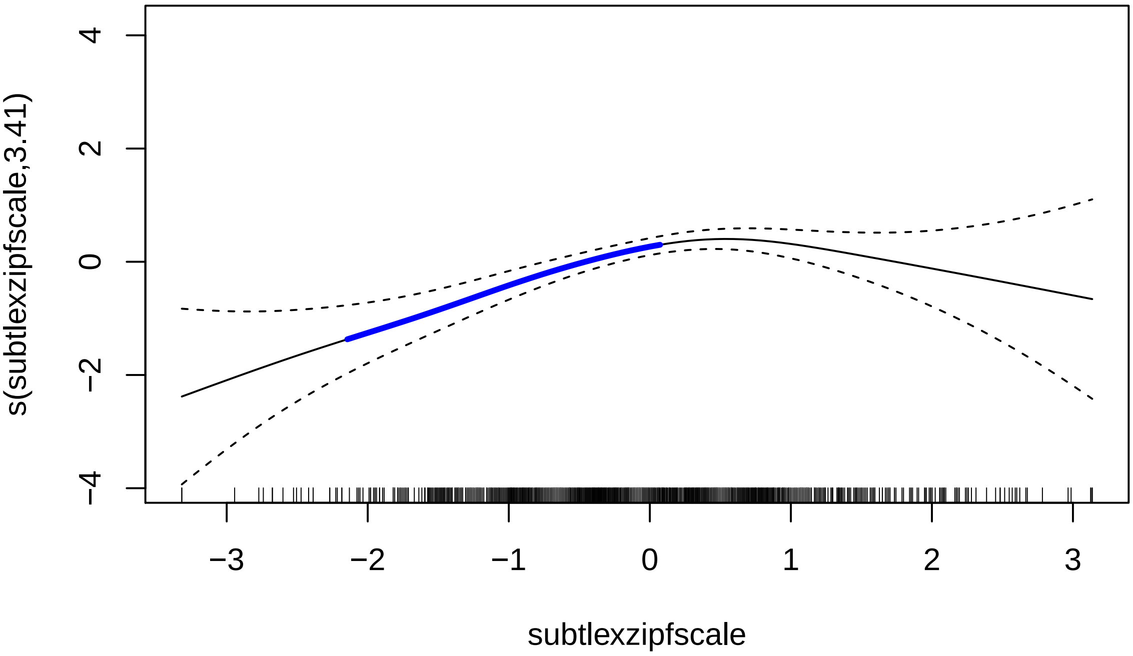
Significant trends

The plots below highlight which sections of the GAM splines are significantly increasing or decreasing. This method comes from https://www.fromthebottomoftheheap.net/2014/05/15/identifying-periods-of-change-with-gams/. The basic idea is to calculate the derivatives of the slope (how much the slope is increasing or decreasing) and then compute confidence intervals for the derivatives from their standard errors. If the confidence intervals of the derivatives do not overlap zero, then they are considered significant.

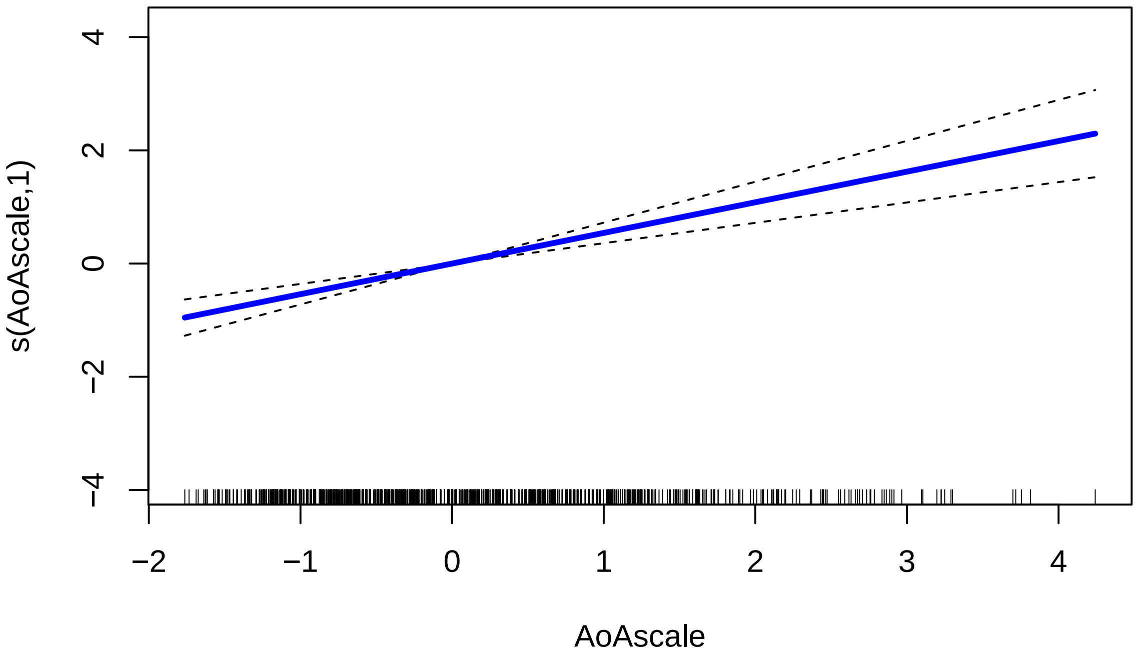
The results suggest:

- Frequency only significantly increasing for a sub-section of the spline
- AoA and length significantly increasing for essentially the whole range
- Concreteness not significantly increasing nor decreasing in any part of the curve.

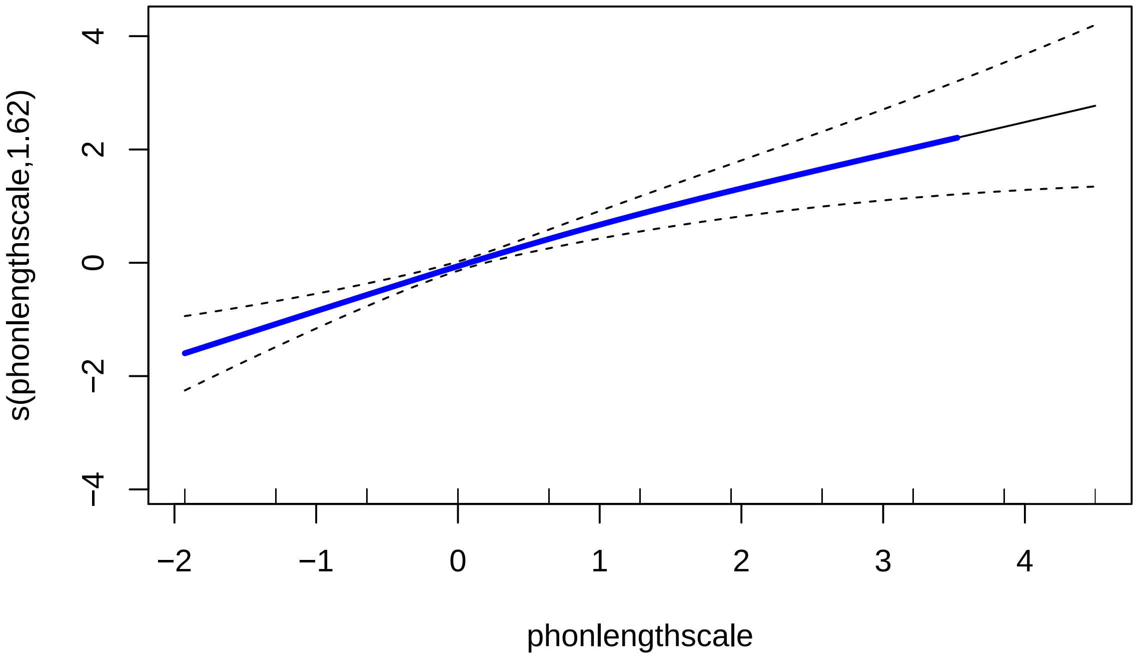
```
pSigFreq = plotGAMSignificantSlopes(m0,"subtlexzipfscale","Frequency")
```



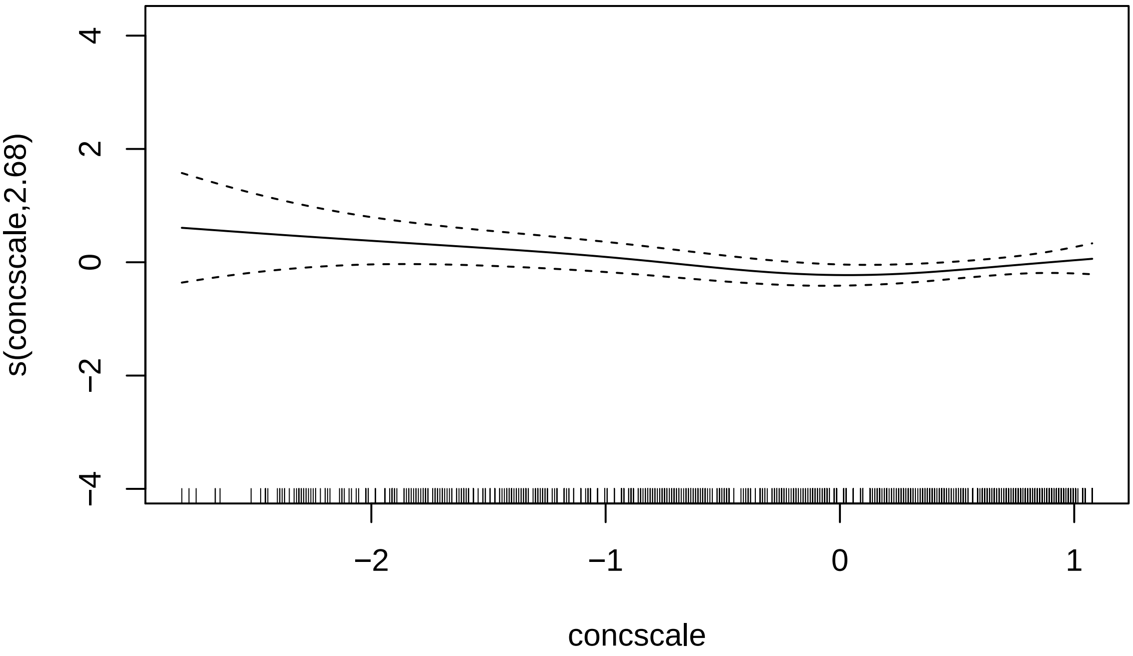
```
pSigAoA =  plotGAMSignificantSlopes(m0,"AoAscale","AoA")
```



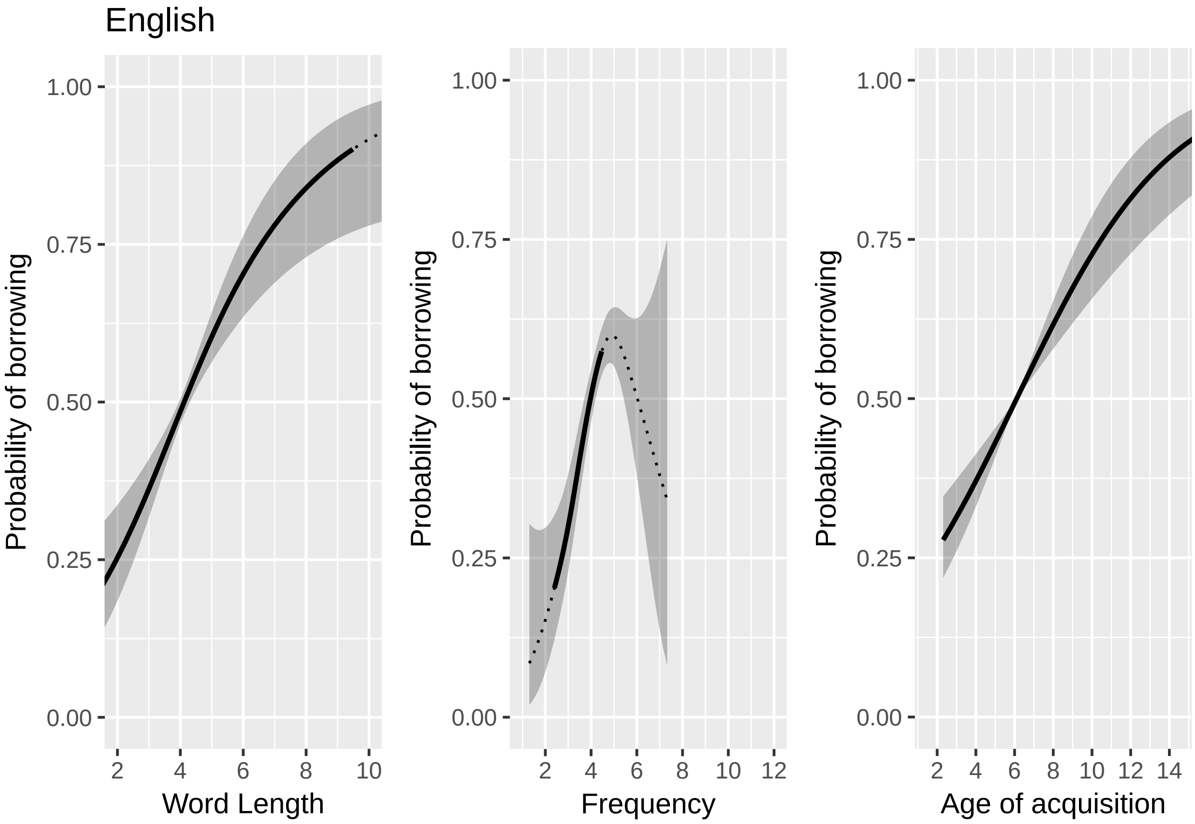
```
pSigLen =  plotGAMSignificantSlopes(m0,"phonlengthscale","length")
```



```
pSigConc = plotGAMSignificantSlopes(m0,"concscale","Concreteness")
```



```
## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).

## Warning: Removed 65 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).
```



```
## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).

## Warning: Removed 65 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).

## Warning: Removed 100 rows containing missing values (geom_path).

## pdf
##   2
```

Objective measures of AoA

Below we run the same model, but with objective, test-based AoA from Brysbaert et al. (2017). Note that the values for objective AoA are only whole numbers, so there are not as many unique values and we have to limit the number of knots that the model uses.

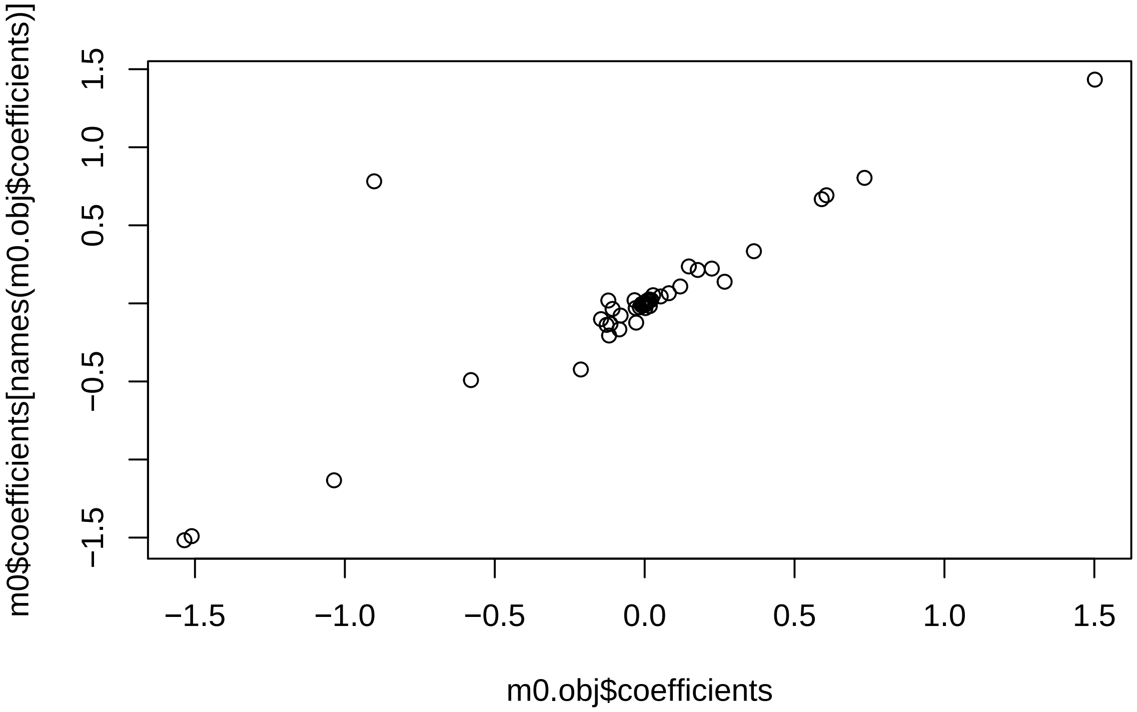
```r
m0.obj = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoA_objscaled, k=3) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoA_objscaled,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2[!is.na(dataloan2$AoA_objscaled),],
    family='binomial')
```

```r
summary(m0.obj)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale) + s(AoA_objscaled, k = 3) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoA_objscaled, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.511      0.439  -3.442 0.000577 ***
## ---
## 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(phonlengthscale)      2.119e+00  2.694 34.748 2.13e-07 ***
## s(AoA_objscaled)        1.762e+00  1.940 28.249 3.47e-06 ***
## s(subtlexzipfscale)     3.437e+00  4.373 25.330 7.77e-05 ***
## s(concscale)            2.220e+00  2.773  7.034   0.0428 *
## s(cat)                  5.869e+00 11.000 46.705 1.79e-10 ***
## s(cat,phonlengthscale)  1.171e+00 11.000  3.044   0.0670 .
## s(cat,AoA_objscaled)    1.879e-06 11.000  0.000   0.7107
## s(cat,subtlexzipfscale) 8.312e-06 11.000  0.000   0.7457
## s(cat,concscale)        8.632e-06 11.000  0.000   0.7968
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.181   Deviance explained = 15.7%
## fREML = 1834.3  Scale est. = 1         n = 1296
```

Very similar results. For example, almost all coefficients are the same:

```
plot(m0.obj$coefficients, m0$coefficients[names(m0.obj$coefficients)])
```



The outlier is the coefficient for `subtlexzipfscale`.

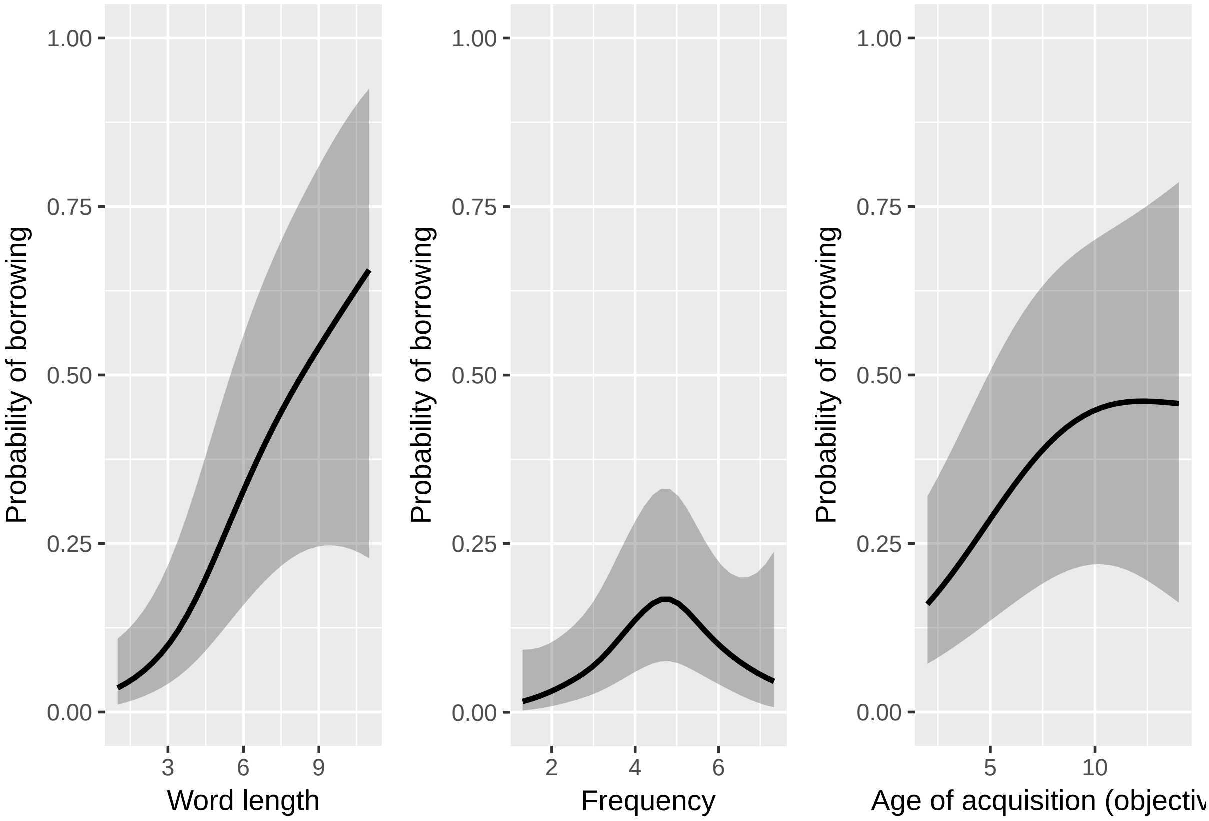
And chi squared terms are similar:

```
m0S = summary(m0)
m0.objS = summary(m0.obj)
cbind(m0=m0S$chi.sq,m0.obj=m0.objS$chi.sq)[1:4,]
```

```
##                            m0    m0.obj
## s(phonlengthscale)  32.336043 34.748289
## s(AoAscale)         35.555290 28.248903
## s(subtlexzipfscale) 32.599058 25.329646
## s(concscale)         7.639604  7.033877
```

Objective AoA: Model plots

Visualise the model smooth terms, independent of influence of random effects. The code is hidden, but you can view it in the Rmd file.



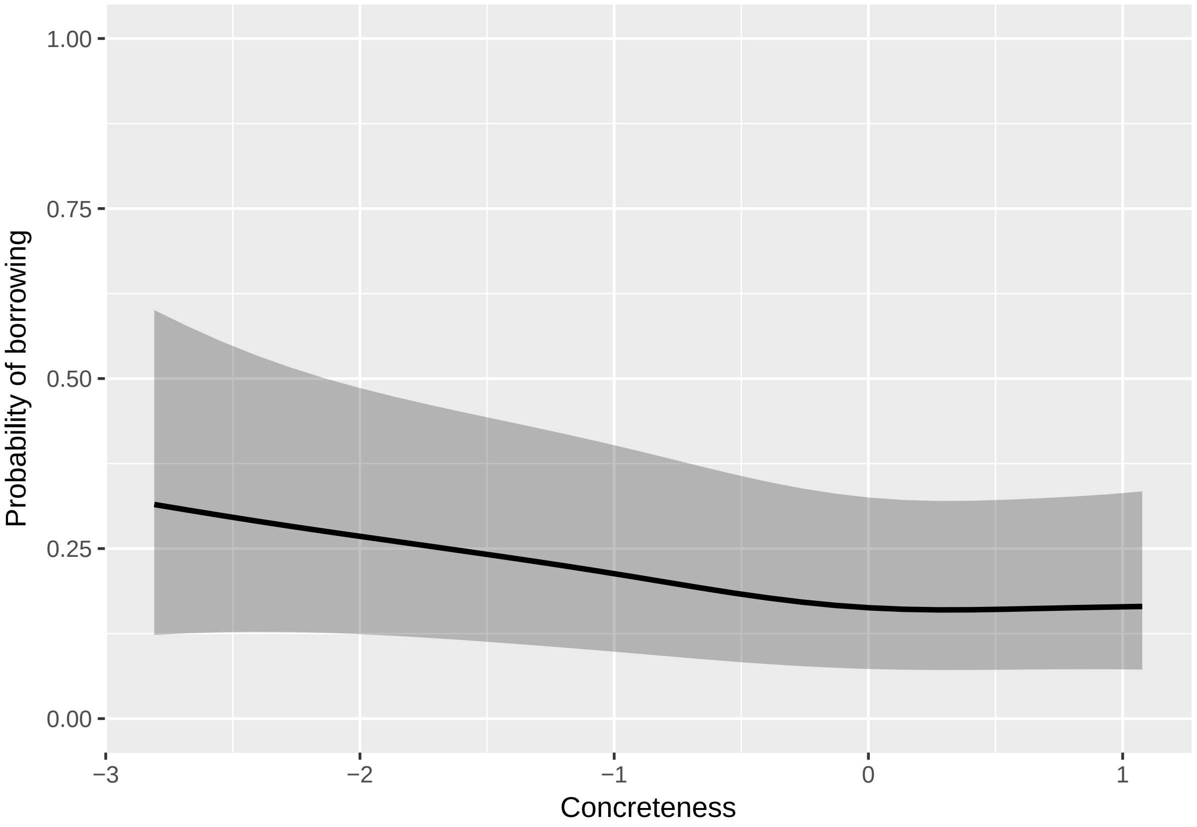
```
## pdf
##   2
```

Note that concreteness is marginally significant in this model. However, the trend is weak, and the decrease is only significant (according to the derivitives test) for a small section of the range:

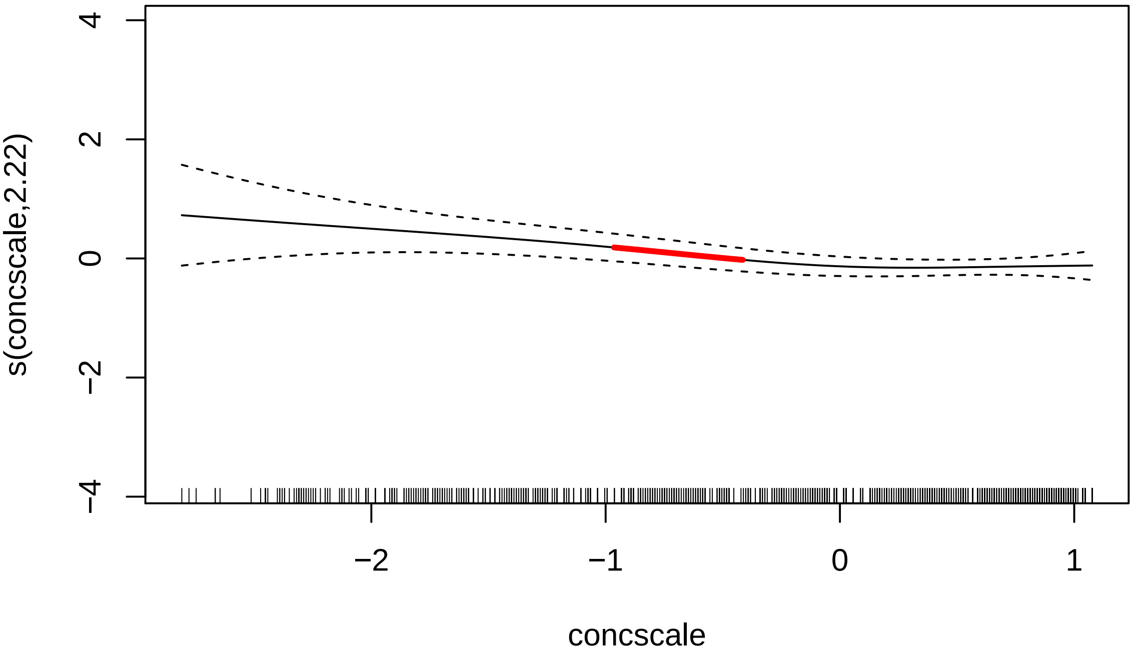
```
px = plot_smooth(m0.obj,view="concscale", rm.ranef = T, print.summary=F)
```

```
px$fv$fit = logit2per(px$fv$fit)
px$fv$ul = logit2per(px$fv$ul)
px$fv$ll = logit2per(px$fv$ll)
px$fv$phonlengthscale = px$fv$concscale * conc.scale + conc.center
```

```
gConc0 = ggplot(px$fv, aes(x=concscale,y=fit)) +
  geom_ribbon(aes(ymin=ll,ymax=ul), alpha=0.3) +
  geom_line(size=1) +
  ylab("Probability of borrowing") +
  xlab("Concreteness")+
  coord_cartesian(ylim=c(0,1))
gConc0
```



```
plotGAMSignificantSlopes(m0.obj,'concscale','Concreteness',aoaLab = "AoA_objscaled")
```

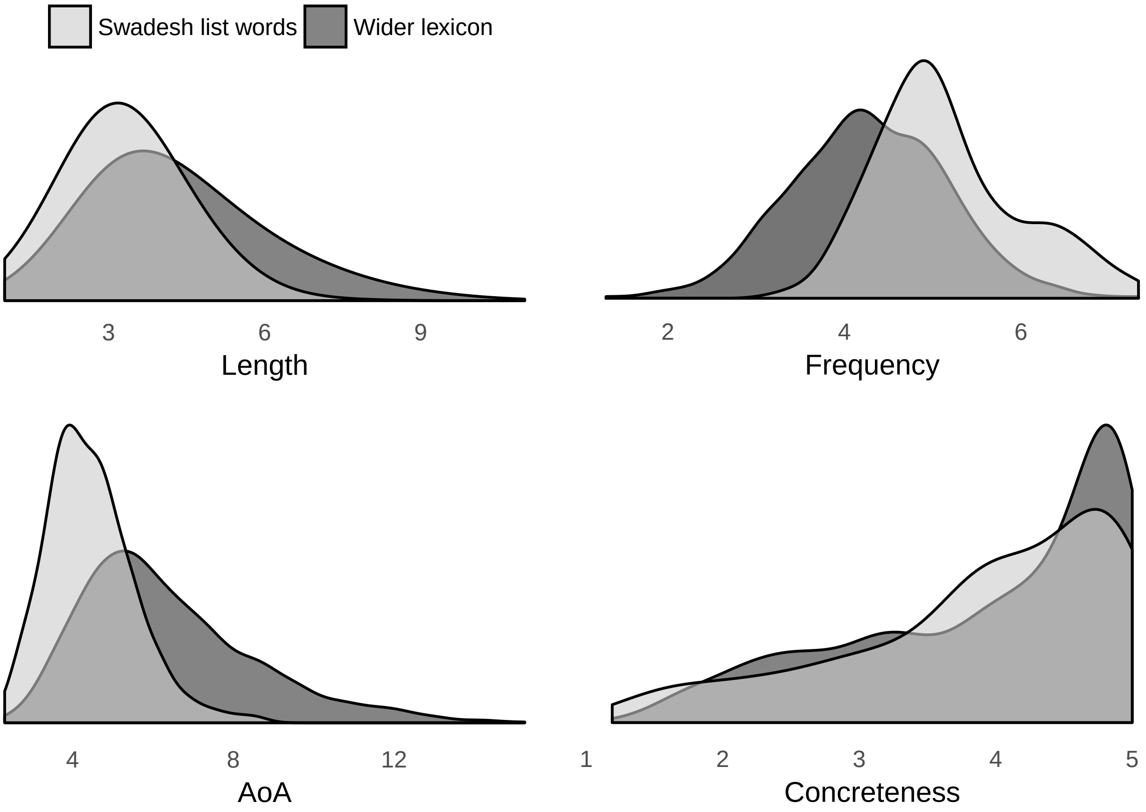


Comparison to Swadesh list words

Match up words with presence or absence in Swedesh list

```
dx = dataloan2[,      c("subtlexzipf",
                        "AoA",
                        "phonlength",
                        "conc",
                        "Swadesh"),]
names(dx) = c("Frequency","AoA","Length","Concreteness","Swadesh")
dx$Swadesh = c("No","Yes")[1+as.numeric(dx$Swadesh)]
dx = dx[complete.cases(dx),]
```

Plot the distributions (code hidden but available in the Rmd file):

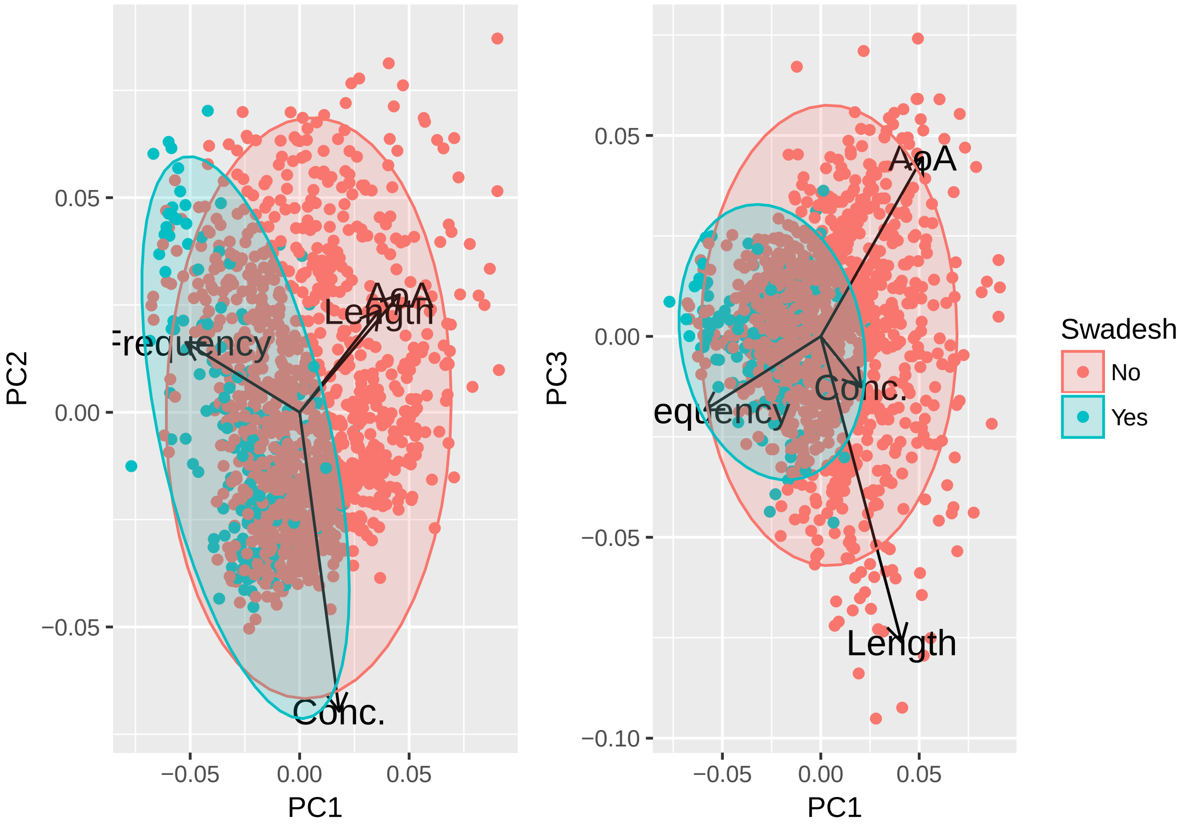


```
## pdf
##   2
```

Try a PCA plot:

```
dx = dataloan2[,      c("subtlexzipfscale",
                        "AoAscale",
                        "phonlengthscale",
                        "concscale",
                        "Swadesh"),]
names(dx) = c("Frequency","AoA","Length","Conc.","Swadesh")
dx$Swadesh = c("No","Yes")[1+as.numeric(dx$Swadesh)]
dx = dx[complete.cases(dx),]
```

```
pc = prcomp(dx[, 1:4])

gpc1 = autoplot(pc, data = dx, colour = 'Swadesh',
         loadings = TRUE, loadings.colour = 'black', loadings.label.colour='black',
         loadings.label = TRUE, loadings.label.size = 5,frame = TRUE, frame.type = 'norm') +
  theme(legend.position = 'none')

gpc2 = autoplot(pc, data = dx, x = 1,y = 3,
         colour = 'Swadesh',
         loadings = TRUE, loadings.colour = 'black', loadings.label.colour='black',
         loadings.label = TRUE, loadings.label.size = 5,frame = TRUE, frame.type = 'norm')
gxpc123 = grid.arrange(gpc1, gpc2, nrow=1, widths=c(0.8,1))
```



```
pdf("../results/graphs/English_Swadesh_PCA.pdf",
    width = 10, height= 4.5)
plot(gxpc123)
dev.off()
```

```
## pdf
##   2
```

Borrowing factors in Swadesh vs non-Swadesh words

First, we make a new model with a fixed effect for whether the word is in the Swadesh list:

```
dataloan2$Swadesh=factor(dataloan2$Swadesh)
mSwadesh = update(m0,~.+Swadesh)
```

Now add an interaction for a word's length and whether it's in the Swadesh list. Then test whether the fit to the data significantly improves.

```
mSwadeshLen = update(mSwadesh,~.+
      s(phonlengthscale,by=Swadesh))
lrtest(mSwadesh,mSwadeshLen)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh + s(phonlengthscale,
##     by = Swadesh)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 20.444 -739.58
## 2 23.091 -737.39 2.6469 4.3889     0.2224
```

Same for AoA:

```
mSwadeshAoA = update(mSwadesh,~.+
      s(AoAscale,by=Swadesh))
lrtest(mSwadesh,mSwadeshAoA)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh + s(AoAscale,
##     by = Swadesh)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 20.444 -739.58
## 2 23.111 -739.30 2.6668 0.5537     0.9069
```

Same for Frequency:

```
mSwadeshFreq = update(mSwadesh,~.+
      s(subtlexzipfscale,by=Swadesh))
lrtest(mSwadesh,mSwadeshFreq)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh + s(subtlexzipfscale,
##     by = Swadesh)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 20.444 -739.58
## 2 22.825 -739.43 2.3813 0.3038     0.8591
```

Same for Concreteness:

```
mSwadeshConc = update(mSwadesh,~.+
      s(concscale,by=Swadesh))
lrtest(mSwadesh,mSwadeshConc)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + Swadesh + s(concscale,
##     by = Swadesh)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 20.444 -739.58
## 2 21.589 -739.06 1.1448 1.0433      0.307
```

The model is not improved by adding an interaction between any predictor and whether a word is in the Swadesh list. That is, the relationship betwen the predictors and borrowing is not significantly different for Swadesh words versus non-Swadesh words.

Sensitivity analyses

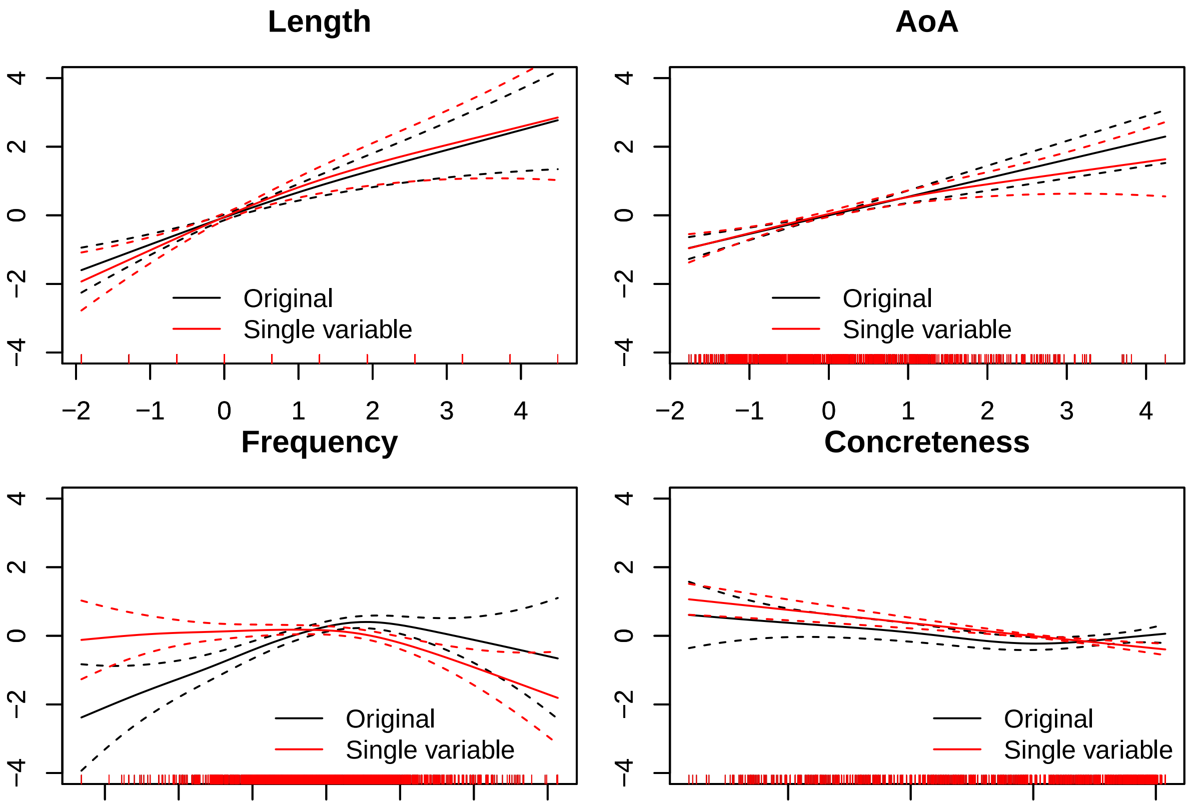
Individual models for each variable

Check that the response function is similar when including a variable alone in a model. Run seperate models with single predictors:

```
m0.length = bam(bor15.cat ~
      s(phonlengthscale)+
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re'),
    data = dataloan2,
    family='binomial')

m0.AoA = bam(bor15.cat ~
      s(AoAscale)+
      s(cat,bs='re')+
      s(cat,AoAscale,bs='re'),
    data = dataloan2,
    family='binomial')

m0.frequency = bam(bor15.cat ~
      s(subtlexzipfscale)+
      s(cat,bs='re')+
      s(cat,subtlexzipfscale,bs='re'),
    data = dataloan2,
    family='binomial')

m0.conc = bam(bor15.cat ~
      s(concscale)+
      s(cat,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2,
    family='binomial')
```

Plot the original curves against the single-variable model curves:

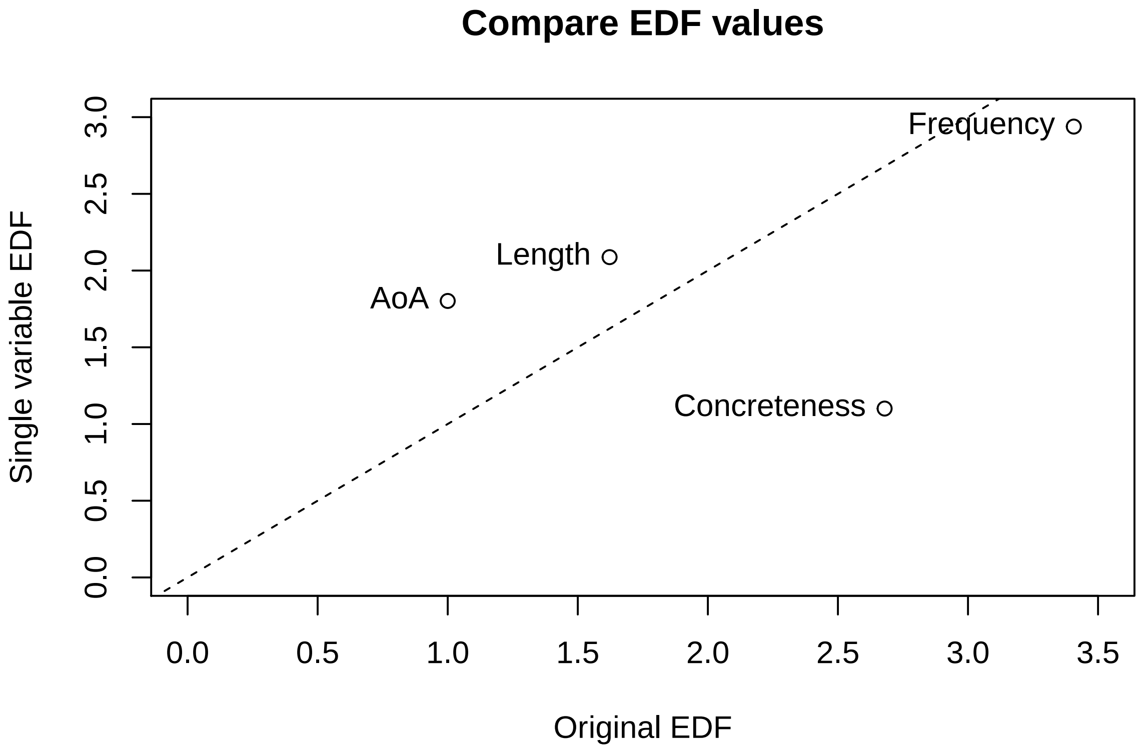
```
par(mfrow=c(2,2), mar=c(1,2,3,1))
mx = list(m0.length, m0.AoA, m0.frequency,m0.conc)
mx.labels = c("Length","AoA","Frequency","Concreteness")
for(i in 1:4){
plot(m0,select=i,main=mx.labels[i],ylim=c(-4,4))
par(new=T)
plot(mx[[i]],select=1,ylim=c(-4,4), col=2,ylab="")
legend(-1,-1.5,legend = c("Original","Single variable"),col=1:2,lty=1, bty='n')
}
```



```
par(mfrow=c(1,1))
```

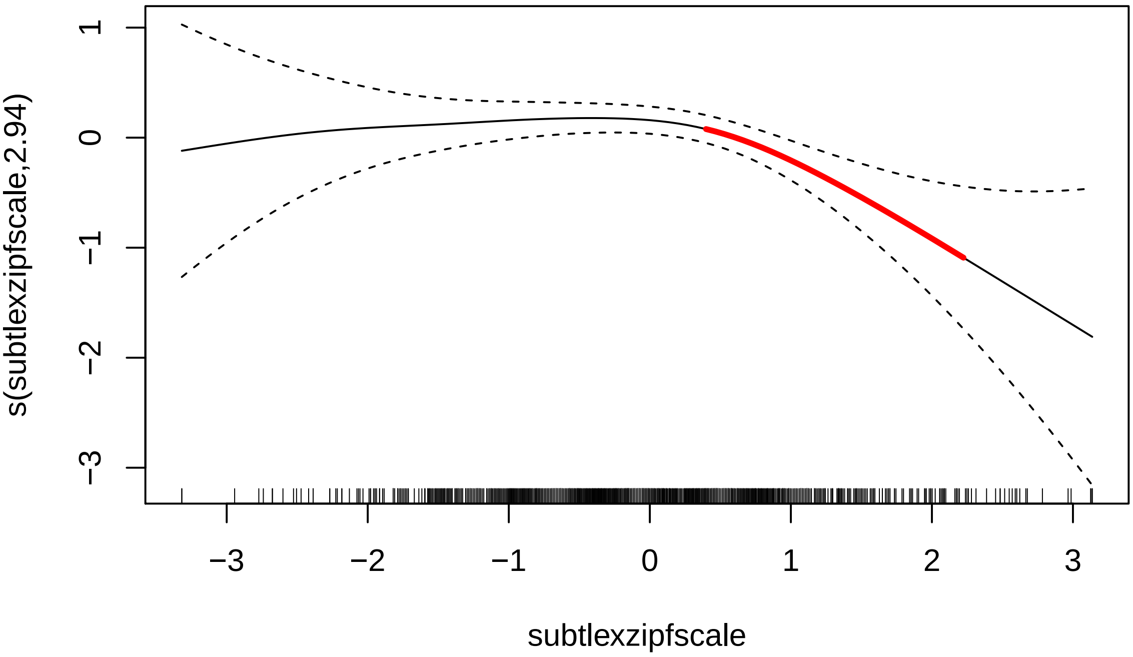
Compare the EDF values.

```
orig.edf = summary(m0)$edf[1:4]
single.edf = lapply(mx,function(X){summary(X)$edf[1]})
plot(orig.edf,
     single.edf,
     xlab="Original EDF",
     ylab="Single variable EDF",
     ylim=c(0,3),xlim=c(0,3.5),
     main="Compare EDF values")
text(orig.edf,single.edf,mx.labels,pos=2)
abline(0,1,lty=2)
```



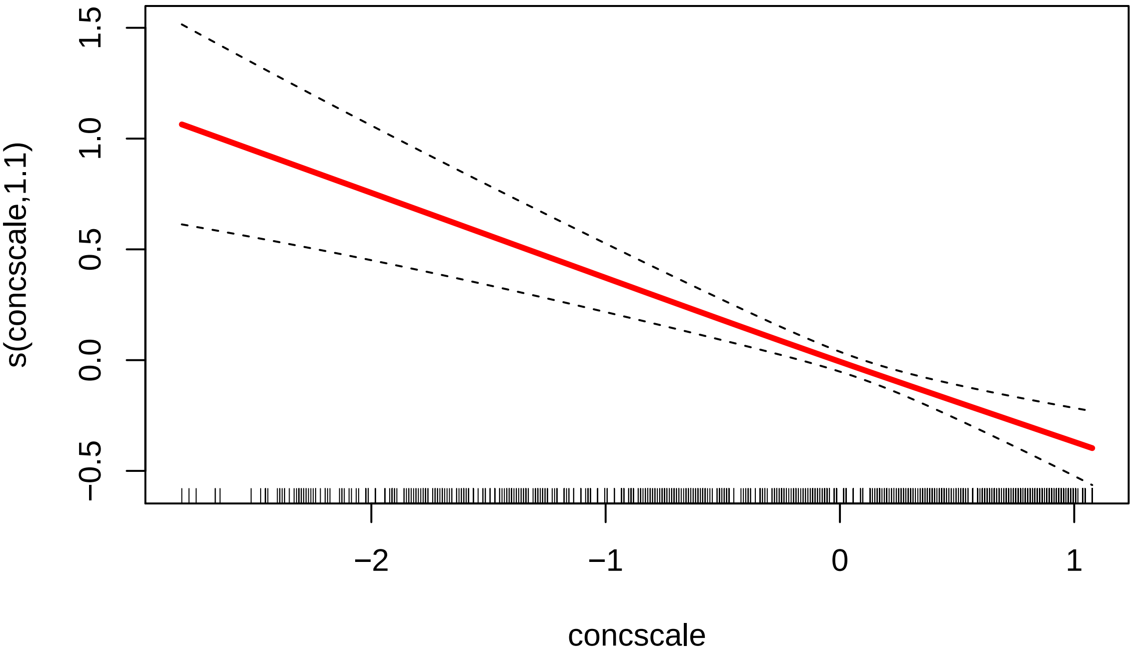
The results for length and AoA are almost identical. The results for frequency are similar (non-linear relationship with a peak in the middle), although the significant slope is now for the higher values:

```
plotGAMSignificantSlopes(m0.frequency,"subtlexzipfscale","Frequency")
```



By itself, concreteness is a signficant predictor.

```
plotGAMSignificantSlopes(m0.conc,"concscale","Concreteness")
```

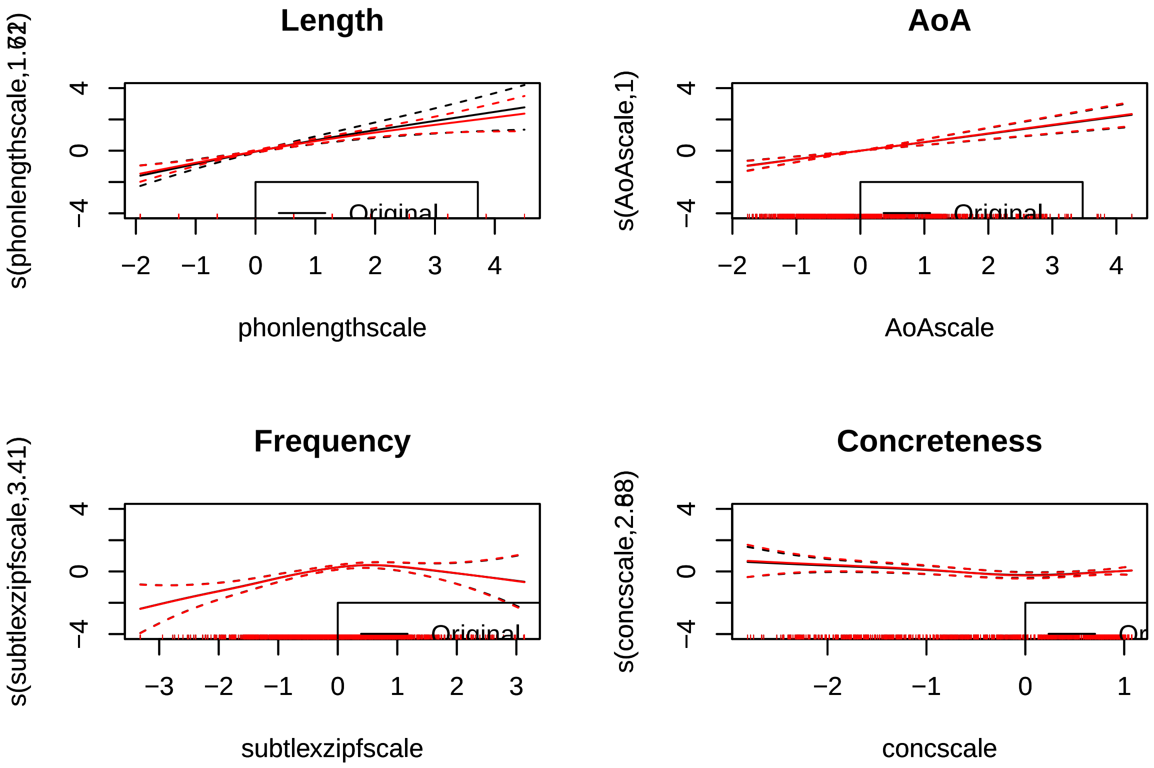


GAM with PoS as fixed effects

Part of speech was modelled above as a random effect. Here we show that the estimates differ very little if we treat part of speech as a fixed effect:

```
m0.posFE = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      cat,
    data = dataloan2,
    family='binomial')
```

```
vars = c("phonlengthscale","AoAscale","subtlexzipfscale","concscale")
varLabels = c("Length","AoA","Frequency","Concreteness")
par(mfrow=c(2,2))
for(i in 1:4){
  plot(m0,select=i,main=varLabels[i],ylim=c(-4,4))
  par(new=T)
  plot(m0.posFE,select=i,ylim=c(-4,4), col=2)
  legend(0,-2,legend = c("Original","PoS Fixed"),col=1:2,lty=1)
}
```



```
par(mfrow=c(1,1))
```

Test the inclusion of interactions between cat and other fixed effects by log ratio tests:

```
m0.posFE.catByLen = update(m0.posFE,~.+s(phonlengthscale,by=cat))
```

```
## Warning in bgam.fit(G, mf, chunk.size, gp, scale, gamma, method = method, :
## fitted probabilities numerically 0 or 1 occurred
```

```
lrtest(m0.posFE,m0.posFE.catByLen)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat)
##      #Df  LogLik    Df  Chisq Pr(>Chisq)
## 1 20.846 -749.18
## 2 28.247 -738.19 7.401 21.984   0.002557 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adding an interaction between part of speech and length results in a better fit. However, since there is a realtively small range of unique length values, and that many part of speech categories have very limited length ranges, then perfect seperation occurs and the model has poor convergence. The results of the model suggest that length has a bigger effect size for some parts of speech than others. This is better captured by a mixed effects model. The estimates for other fixed effects are not qualitatively different in this model.

cat x AoA is not significant:

```
m0.posFE.catByAoA = update(m0.posFE.catByLen,~.+s(AoAscale,by=cat))
```

```
## Warning in bgam.fit(G, mf, chunk.size, gp, scale, gamma, method = method, :
## fitted probabilities numerically 0 or 1 occurred
```

```
lrtest(m0.posFE.catByLen,m0.posFE.catByAoA)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat)
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat) + s(AoAscale,
##     by = cat)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 28.247 -738.19
## 2 37.276 -733.99 9.0289 8.3887     0.4955
```

cat x frequency is not significant:

```
m0.posFE.catByFreq = update(m0.posFE.catByLen,~.+s(subtlexzipfscale,by=cat))
```

```
## Warning in bgam.fit(G, mf, chunk.size, gp, scale, gamma, method = method, :
## fitted probabilities numerically 0 or 1 occurred
```

```
lrtest(m0.posFE.catByLen,m0.posFE.catByFreq)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat)
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat) + s(subtlexzipfscale,
##     by = cat)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 28.247 -738.19
## 2 36.150 -731.51 7.9032 13.359     0.1001
```

cat x concreteness is not significant:

```
m0.posFE.catByConc = update(m0.posFE.catByLen,~.+s(concscale,by=cat))
```

```
## Warning in bgam.fit(G, mf, chunk.size, gp, scale, gamma, method = method, :
## fitted probabilities numerically 0 or 1 occurred
```

```
lrtest(m0.posFE.catByLen,m0.posFE.catByConc)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat)
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + cat + s(phonlengthscale, by = cat) + s(concscale,
##     by = cat)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 28.247 -738.19
## 2 39.926 -730.60 11.679 15.176      0.232
```

Controlling for source language length

We attempted to obtain estimates for the distribution of word lengths in source languages (for the English dataset). This is very difficult, because there are 25 source languages from across the world, and very few of these have large word list sources. The only possibility is the Automatic Similarity Judgement Program database, which contains a small standard list of words for many languages. We obtained these words for the source languages (subsituting phylogenetically close languages for those with no data, words which were not borrowed were matched to modern English data from the ASJP). We calculated the average word length for each source language. See the file `processing/GetSourceLanguageWordLengths.R` for details.

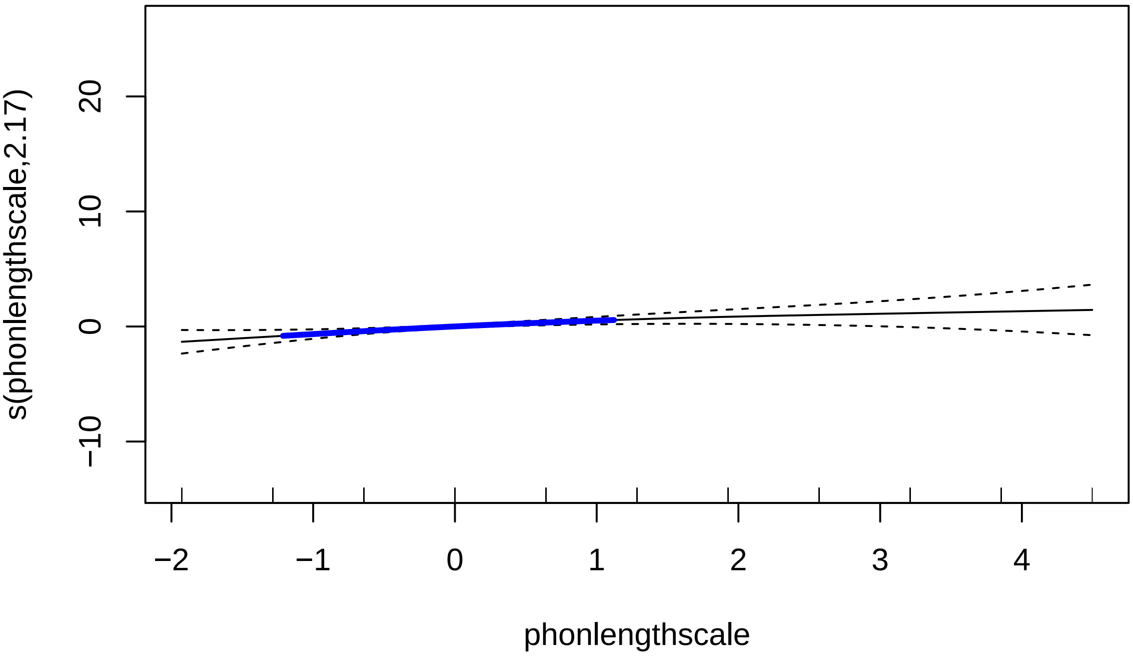
Compare the original model with a model with control for average donor langauge word length.

```
mMeanLenControl = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
        s(SLMWL,k=3) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2,
    family='binomial')

summary(mMeanLenControl)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(SLMWL, k = 3) + s(cat, bs = "re") + s(cat,
##     phonlengthscale, bs = "re") + s(cat, AoAscale, bs = "re") +
##     s(cat, subtlexzipfscale, bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.233      0.779   1.582    0.114
##
## Approximate significance of smooth terms:
##                             edf Ref.df Chi.sq p-value
## s(phonlengthscale)      2.1654  2.747 10.783 0.01064 *
## s(AoAscale)             2.0591  2.623  9.652 0.01719 *
## s(subtlexzipfscale)     3.0172  3.865 14.632 0.00511 **
## s(concscale)            2.3464  2.955  1.157 0.72110
## s(SLMWL)                1.9352  1.995 92.038 < 2e-16 ***
## s(cat)                  4.0991 11.000 20.104 0.10109
## s(cat,phonlengthscale)  0.8335 11.000  0.756 0.44015
## s(cat,AoAscale)         0.8726 11.000  2.489 0.36014
## s(cat,subtlexzipfscale) 2.3598 11.000 10.795 0.23925
## s(cat,concscale)        1.7455 11.000  1.086 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.702   Deviance explained = 46.6%
## fREML = 9.3391e+10  Scale est. = 1         n = 1317
```

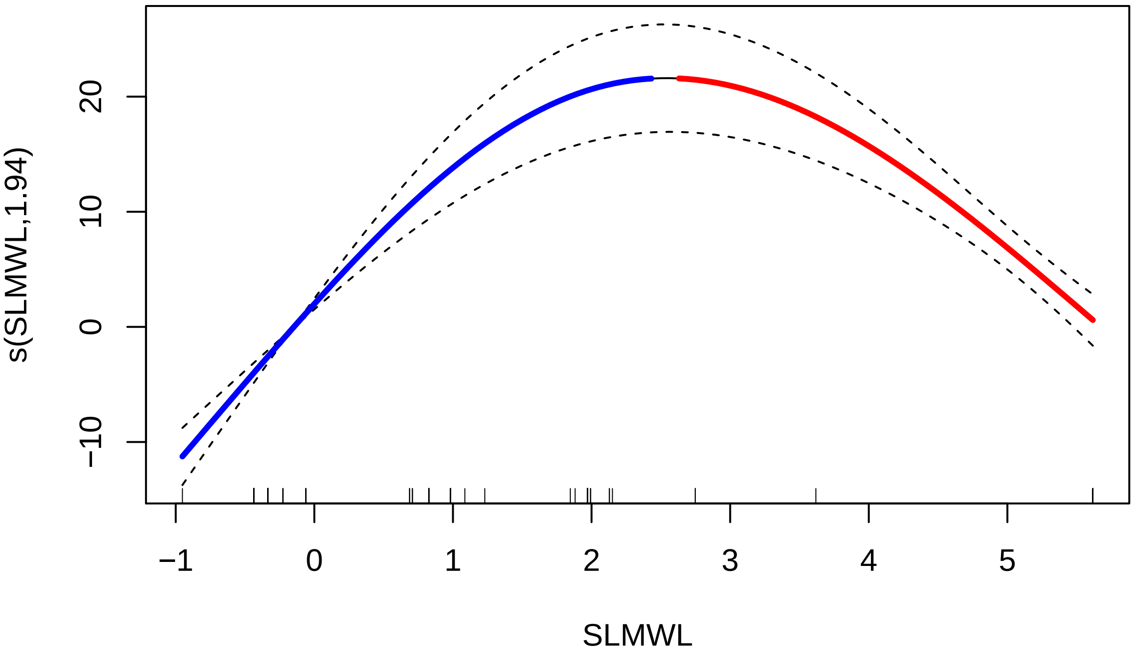
The original length term is still significant. Original length term curve:

```
plotGAMSignificantSlopes(mMeanLenControl,
            "phonlengthscale",'phonlengthscale')
```

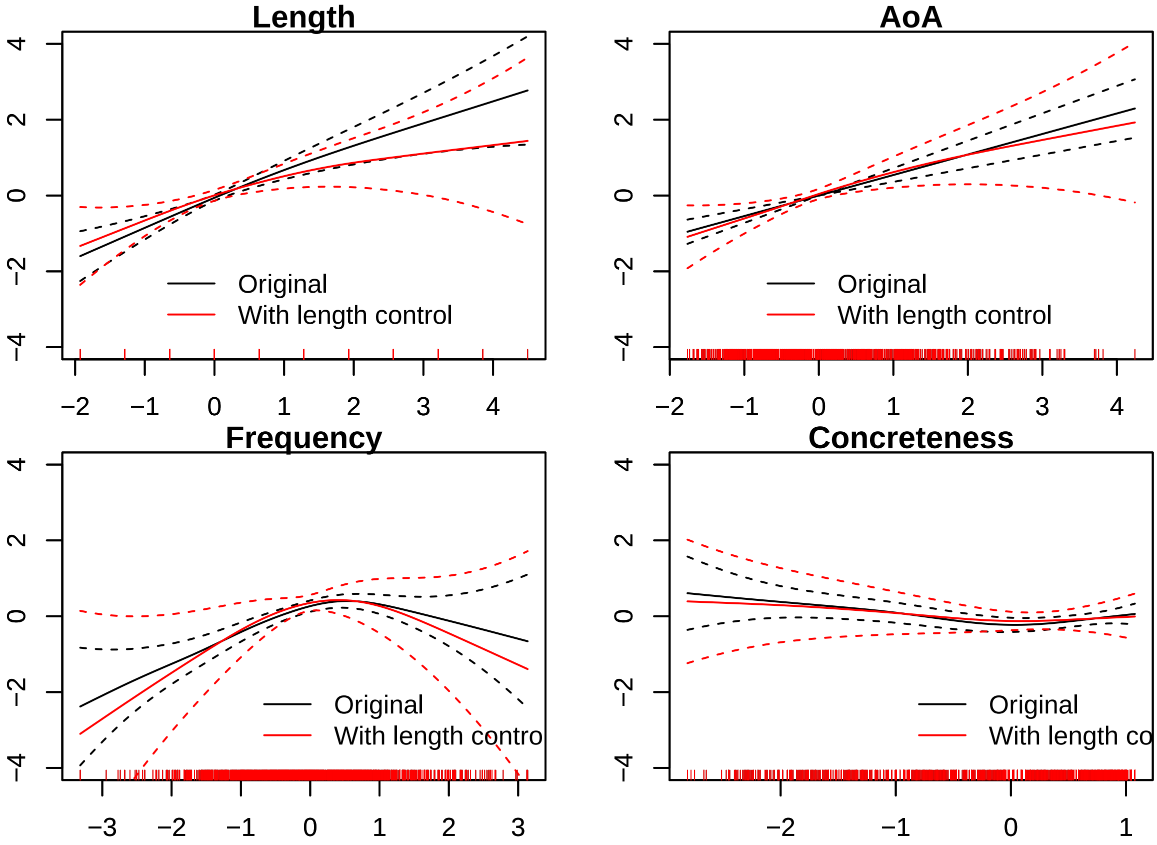


Relationship between mean word length in donor language and probaility of borrowing:

```
plotGAMSignificantSlopes(mMeanLenControl,
            "SLMWL",'SLMWL')
```



```
mx.labels = c("Length","AoA","Frequency","Concreteness")
par(mfrow=c(2,2),mar=c(2,2,1,2))
for(i in 1:4){
plot(m0,select=i,main=mx.labels[i],ylim=c(-4,4))
par(new=T)
plot(mMeanLenControl,select=i,ylim=c(-4,4), col=2,ylab="")
legend(-1,-1.5,legend = c("Original","With length control"),col=1:2,lty=1, bty='n')
}
```



```r
par(mfrow=c(1,1))
```

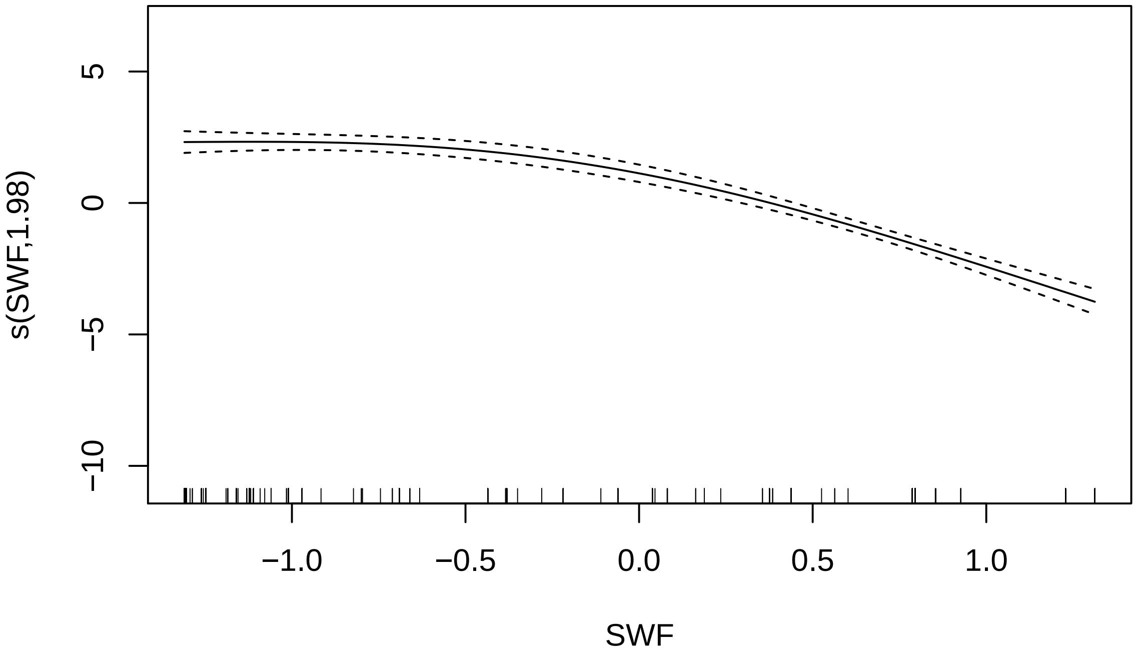
We also attempted to calculate the measure that the reviewer suggested: the frequency of the length of the given borrowed word in each donor language. This was done with the ASJP data. For each borrowed word, we calculated the frequency of having a word of that length in the source language. The word lists are small, so not all word lengths are represented. Missing values were imputed by regression. Frequency estimates for non-borrowed words came from estimates from the ASJP for modern English.

Use the source word length frequency measure to control for borrowing from langauges with longer words.

```
mLenFreqControl = update(m0, ~.+s(SWF,k=3))
summary(mLenFreqControl)
```

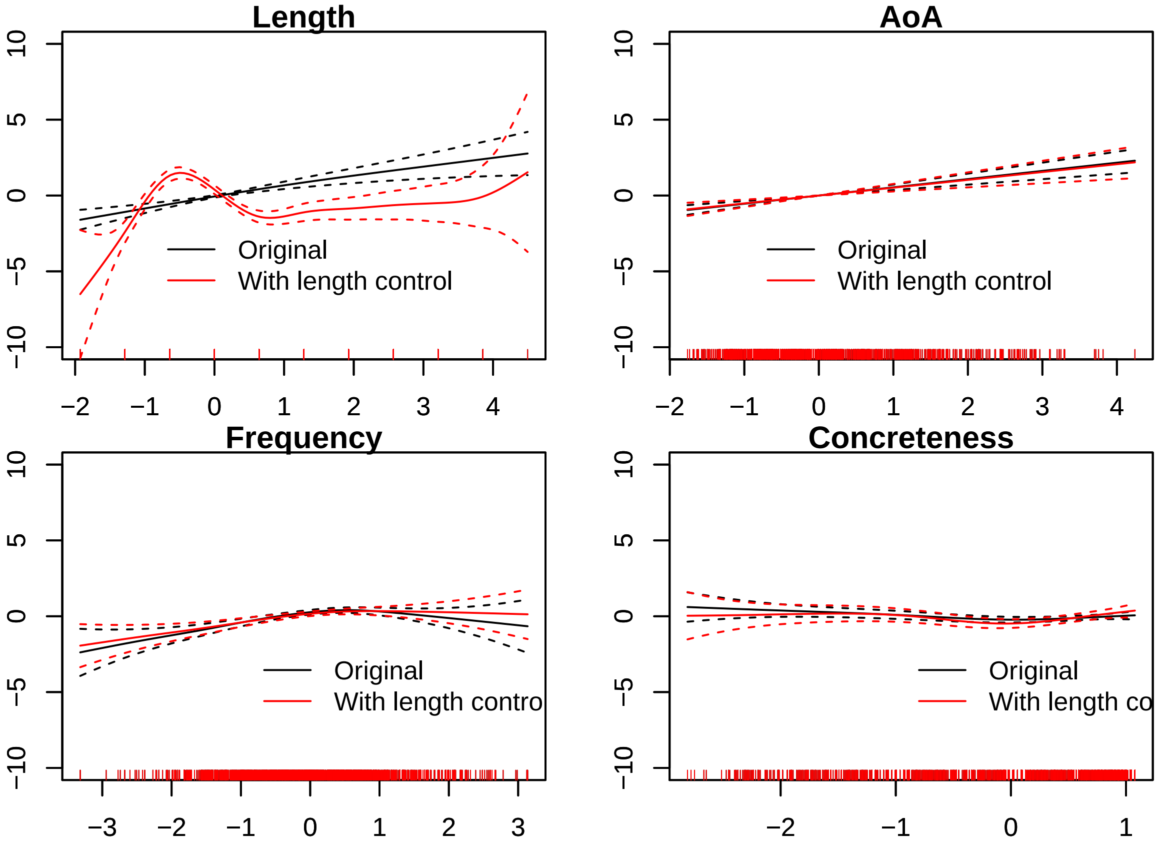
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(SWF, k = 3)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3972     0.4272  -3.271  0.00107 **
## ---
## 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(phonlengthscale)      6.811e+00  7.538  97.653  < 2e-16 ***
## s(AoAscale)             1.000e+00  1.000  17.561 2.78e-05 ***
## s(subtlexzipfscale)     2.614e+00  3.356  16.784 0.001305 **
## s(concscale)            3.541e+00  4.393  11.260 0.035509 *
## s(cat)                  5.220e+00 11.000  24.773 0.000622 ***
## s(cat,phonlengthscale)  9.873e-01 11.000   2.630 0.115578
## s(cat,AoAscale)         3.301e-01 11.000   0.418 0.287609
## s(cat,subtlexzipfscale) 4.823e-05 11.000   0.000 0.576417
## s(cat,concscale)        1.001e+00 11.000   1.930 0.364679
## s(SWF)                  1.979e+00  1.999 250.988  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.513   Deviance explained = 41.5%
## fREML = 3953.9  Scale est. = 1         n = 1317
```

```
plot(mLenFreqControl,select=10)
```



Compare the original model with a model with control for donor language length.

```
mx.labels = c("Length","AoA","Frequency","Concreteness")
par(mfrow=c(2,2),mar=c(2,2,1,2))
for(i in 1:4){
plot(m0,select=i,main=mx.labels[i],ylim=c(-10,10))
par(new=T)
plot(mLenFreqControl,select=i,ylim=c(-10,10), col=2,ylab="")
legend(-1,-1.5,legend = c("Original","With length control"),col=1:2,lty=1, bty='n')
}
```



```
par(mfrow=c(1,1))
```

Number of morphemes

For the Dutch data, we found that the effect of length differed dramatically according to whether the word was monomorphemic or not. Here we run the same analysis for English. Data is from CELEX.

```
dataloan2$monomorphemic = "other"
dataloan2$monomorphemic[dataloan2$morph_code %in% c("M","Z")] = "monomorphemic"
dataloan2$monomorphemic = factor(dataloan2$monomorphemic)
```

```
mMorph = bam(bor15.cat ~
      s(phonlengthscale, by=monomorphemic) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2,
    family='binomial')
lrtest(m0,mMorph)
```

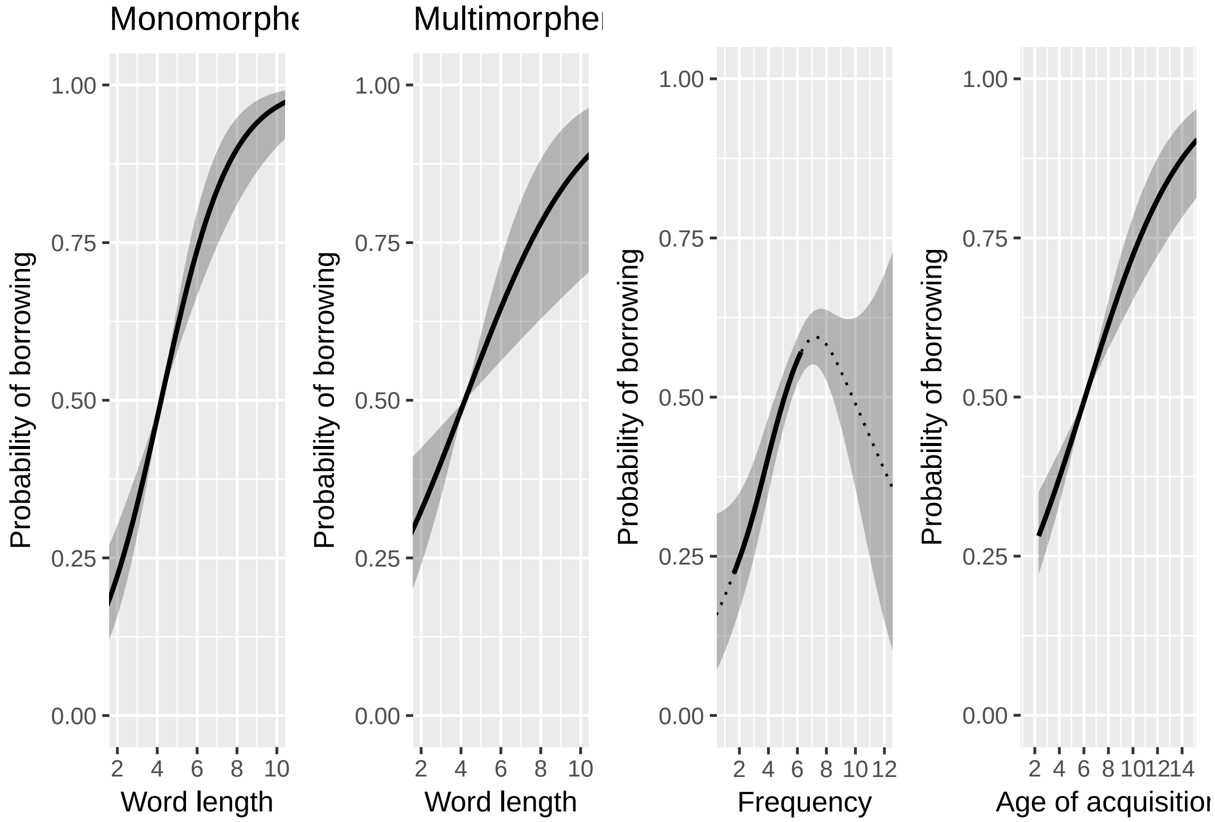
```
## Likelihood ratio test
```

```
## 
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) + 
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale, 
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale, 
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, by = monomorphemic) + s(AoAscale) + 
##     s(subtlexzipfscale) + s(concscale) + s(cat, bs = "re") + 
##     s(cat, phonlengthscale, bs = "re") + s(cat, AoAscale, bs = "re") + 
##     s(cat, subtlexzipfscale, bs = "re") + s(cat, concscale, bs = "re")
##      #Df  LogLik       Df  Chisq Pr(>Chisq)  
## 1 19.758 -749.22                             
## 2 19.243 -746.84 -0.51516 4.7547    0.02922 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mMorph)
```

```
## 
## Family: binomial 
## Link function: logit 
## 
## Formula:
## bor15.cat ~ s(phonlengthscale, by = monomorphemic) + s(AoAscale) + 
##     s(subtlexzipfscale) + s(concscale) + s(cat, bs = "re") + 
##     s(cat, phonlengthscale, bs = "re") + s(cat, AoAscale, bs = "re") + 
##     s(cat, subtlexzipfscale, bs = "re") + s(cat, concscale, bs = "re")
## 
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)   
## (Intercept)   -1.419      0.448  -3.167  0.00154 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##                                                     edf Ref.df Chi.sq
## s(phonlengthscale):monomorphemicmonomorphemic 1.000e+00  1.000 35.890
## s(phonlengthscale):monomorphemicother         1.000e+00  1.000 11.733
## s(AoAscale)                                   1.000e+00  1.000 34.173
## s(subtlexzipfscale)                           3.341e+00  4.250 29.195
## s(concscale)                                  2.657e+00  3.315  7.772
## s(cat)                                        5.931e+00 11.000 39.685
## s(cat,phonlengthscale)                        1.237e+00 11.000  3.508
## s(cat,AoAscale)                               3.157e-06 11.000  0.000
## s(cat,subtlexzipfscale)                       8.406e-06 11.000  0.000
## s(cat,concscale)                              1.113e-05 11.000  0.000
##                                                p-value    
## s(phonlengthscale):monomorphemicmonomorphemic 2.09e-09 ***
## s(phonlengthscale):monomorphemicother         0.000614 ***
## s(AoAscale)                                   5.04e-09 ***
## s(subtlexzipfscale)                           1.13e-05 ***
## s(concscale)                                  0.066836 .  
## s(cat)                                        1.36e-08 ***
## s(cat,phonlengthscale)                        0.054578 .  
## s(cat,AoAscale)                               0.733605    
## s(cat,subtlexzipfscale)                       0.739814    
## s(cat,concscale)                                 0.642216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.193   Deviance explained = 16.5%
## fREML = 1865.4  Scale est. = 1         n = 1317
```

Plot the results from the new model..



```
## pdf
##   2
```

```
## pdf
##   2
```

3 Dutch data (study 2)

Cognitive influences in language evolution: Dutch data

Introduction

This is the model code for Monaghan & Roberts, “Cognitive influences in language evolution: Psycholinguistic predictors of loan word borrowing”. It takes data from the WOLD database of borrowing for Dutch and tries to predict whether a word has been borrowed or not according to various psycholinguitic measures.

Load libraries

```
library(mgcv)
library(sjPlot)
library(lattice)
library(ggplot2)
library(gplots)
library(dplyr)
library(party)
library(lmtest)
library(gridExtra)
library(itsadug)
library(car)
library(caret)
library(scales)
library(binom)


logit2per = function(X){
  return(exp(X)/(1+exp(X)))
}

rescaleGam = function(px, n, xvar, xlab="",breaks=NULL,xlim=NULL){
  y = logit2per(px[[n]]$fit)
  x = px[[n]]$x *attr(xvar,"scaled:scale") + attr(xvar,"scaled:center")
  se.upper = logit2per(px[[n]]$fit+px[[n]]$se)
  se.lower = logit2per(px[[n]]$fit-px[[n]]$se)
  dx = data.frame(x=x,y=y,ci.upper=se.upper,ci.lower=se.lower)
  plen = ggplot(dx, aes(x=x,y=y))+
    geom_ribbon(aes(ymin=ci.lower,ymax=ci.upper), alpha=0.3)+
    geom_line(size=0.5,linetype=3) +
    xlab(xlab)+
    ylab("Probability of borrowing")
  if(!is.null(breaks)){
    plen = plen + scale_x_continuous(breaks = breaks)
  }
  if(!is.null(xlim)){
   plen = plen + coord_cartesian(ylim = c(0,1),xlim=xlim)
  } else{
    plen = plen + coord_cartesian(ylim = c(0,1))
  }
  return(plen)
}

source("GAM_derivaties.R")
```

Load data

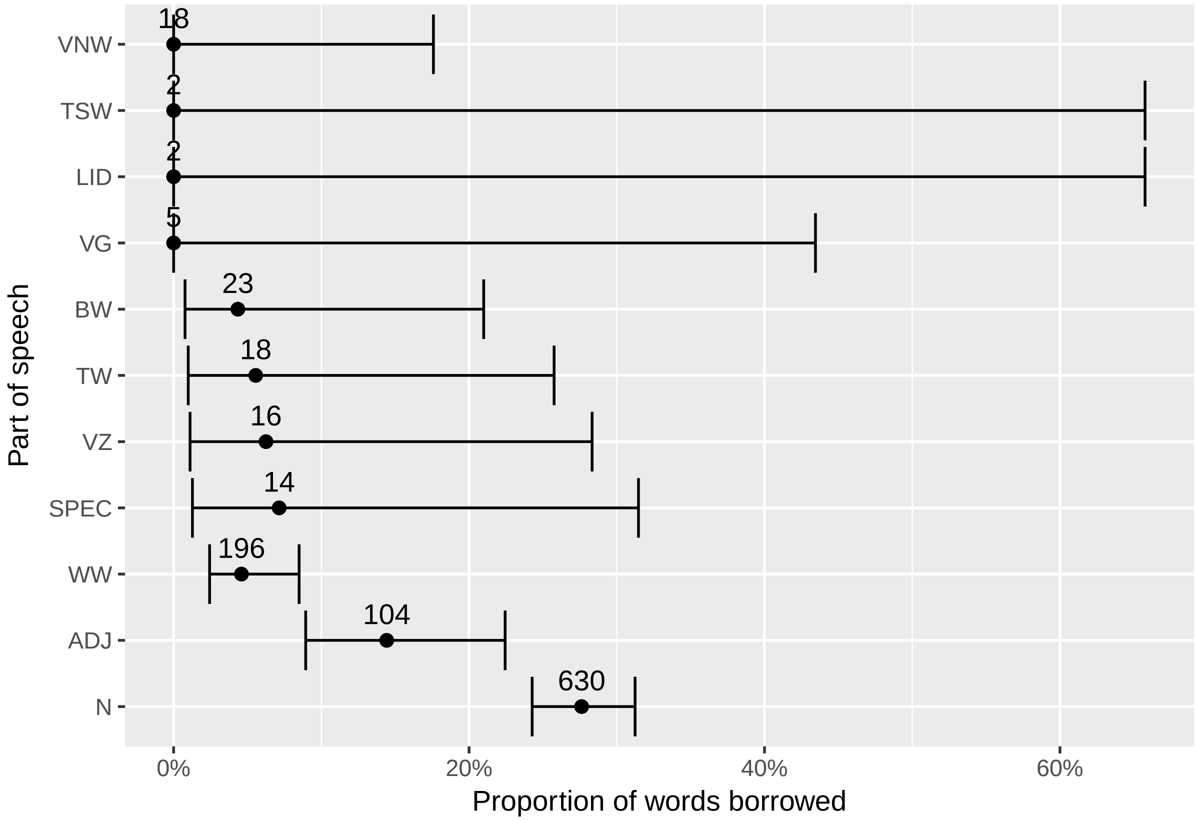
The Dutch data is processed very similarly to the English data. The full process can be found in the `processing` folder, but here we just load the final prepared data frame:

```
load("../data/loanwords_Dutch.Rdat")
```

Part of speech

We calculate the means, but the number of observations is very different for each category. We estimate confidence intervals around the mean with Wilson's binomial confidence interval method.

```
catx = data.frame(
  PoS = tapply(dutch$cat, dutch$cat, function(X){as.character(X[1])}),
  mean = tapply(dutch$bor15, dutch$cat, mean),
  n = tapply(dutch$bor15, dutch$cat, length),
  confint = binom.confint(
    tapply(dutch$bor15, dutch$cat, sum),
    tapply(dutch$bor15, dutch$cat, length),
    methods="wilson"
  )
)
catx = catx[order(catx$confint.lower, decreasing = T),]
catx$PoS = factor(catx$PoS, levels = catx[order(catx$confint.lower, decreasing = T),]$PoS)

posg = ggplot(catx, aes(x=mean, y=PoS)) +
  geom_point(size=2) +
  ylab("Part of speech") +
  xlab("Proportion of words borrowed")+
  scale_x_continuous(labels=percent_format()) +
  geom_text(aes(label=n), nudge_y=0.4) +
  geom_errorbarh(aes(xmin=confint.lower, xmax=confint.upper))
posg
```



```
pdf("../results/graphs/POS_Borrowing_Dutch.pdf",
    width = 6,
    height = 4)
posg
dev.off()
```

```
## pdf
##   2
```

```
catx$mean= catx$mean*100
catx$confint.lower= catx$confint.lower*100
catx$confint.upper= catx$confint.upper*100
write.csv(catx[,c("PoS","mean",
                  'n','confint.lower','confint.upper')],
    "../results/Dutch_POS_BorrowingProportions.csv",
    row.names = F)
```

GAM model

Dutch data has 1028 datapoints.

The range of the length variable limits the number of knots that the gam model can fit. We use k=4, suggested by the rule of thumb in Winter & Wieling (2016) being less than half of the number of unique values.

```
m0.dutch = bam(bor15.cat ~
      s(phonlengthscale, k=4) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dutch,
    family='binomial')
```

```
summary(m0.dutch)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3492     0.3773  -6.226 4.78e-10 ***
## ---
## 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(phonlengthscale)      2.644e+00  2.904 16.919 0.000425 ***
## s(AoAscale)             1.386e+00  1.683 11.311 0.006505 **
## s(subtlexzipfscale)     3.667e+00  4.606 12.048 0.021505 *
## s(concscale)            1.653e+00  2.046  3.036 0.232210
## s(cat)                  3.715e+00 10.000 38.590 2.86e-08 ***
## s(cat,phonlengthscale)  6.887e-05 10.000  0.000 0.364693
## s(cat,AoAscale)         7.494e-06 10.000  0.000 0.966115
## s(cat,subtlexzipfscale) 1.346e-05 10.000  0.000 0.668251
## s(cat,concscale)        1.233e+00 10.000  3.255 0.094333 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.115   Deviance explained = 13.7%
## fREML = 1431.8  Scale est. = 1         n = 1028
```

Interactions

Test whether an interaction between AoA and frequency is warranted:

```
m1.dutch = bam(bor15.cat ~
      s(phonlengthscale, k=3) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(AoAscale,subtlexzipfscale),
    data = dutch,
    family='binomial')

lrtest(m0.dutch,m1.dutch)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, k = 3) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(AoAscale,
##     subtlexzipfscale)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 19.272 -439.79
## 2 20.348 -439.66 1.0755 0.2535     0.6146
```

No significant improvement.

Test whether an interaction between AoA and length is warranted:

```
m2.dutch = bam(bor15.cat ~
      s(phonlengthscale, k=3) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(AoAscale,phonlengthscale),
    data = dutch,
    family='binomial')

lrtest(m0.dutch,m2.dutch)
```

```
## Likelihood ratio test
```

```
##
## Model 1: bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, k = 3) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(AoAscale,
##     phonlengthscale)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 19.272 -439.79
## 2 21.220 -439.07 1.9473 1.4249     0.4904
```

There is no improvement in log likelihood.

Test whether an interaction between Frequency and length is warranted:

```
m3.dutch = bam(bor15.cat ~
      s(phonlengthscale, k=3) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      te(subtlexzipfscale,phonlengthscale),
    data = dutch,
    family='binomial')

lrtest(m0.dutch,m3.dutch)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, k = 3) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + te(subtlexzipfscale,
##     phonlengthscale)
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 19.272 -439.79
## 2 22.040 -437.60 2.7677 4.3719      0.224
```

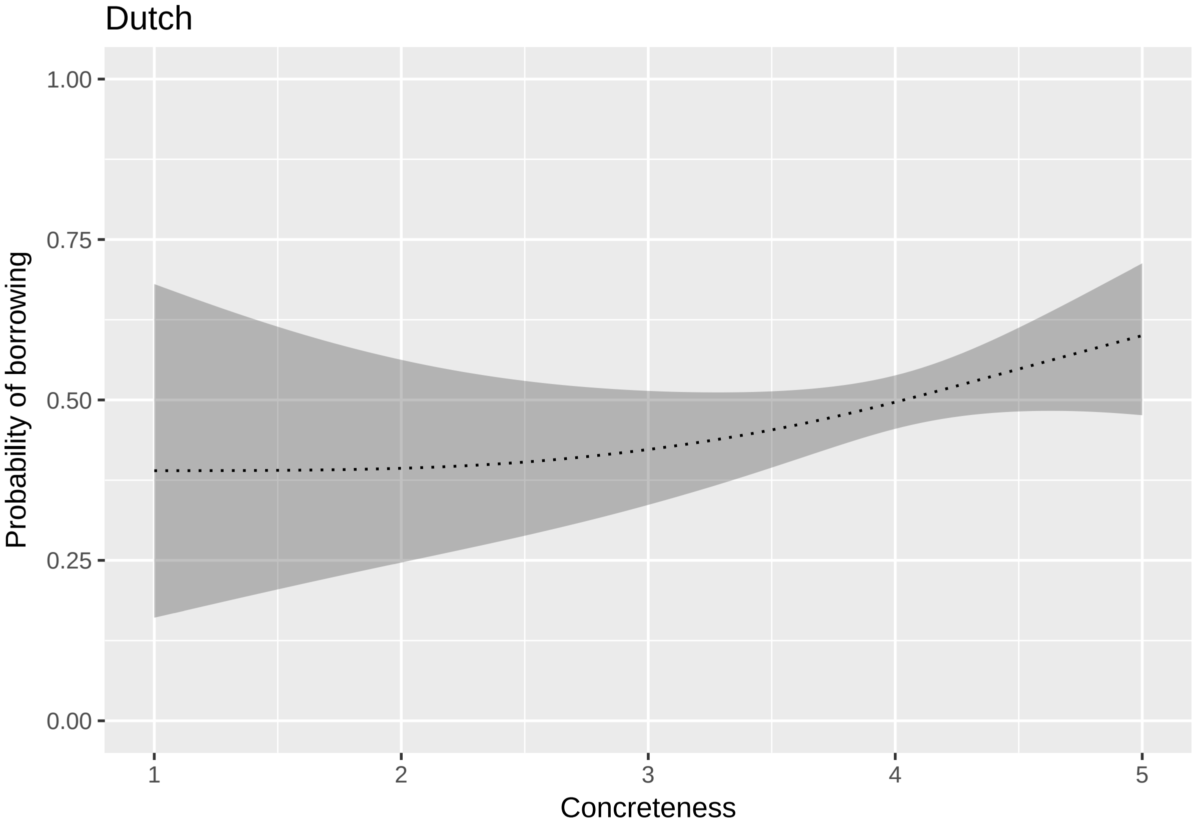
No significant improvement.

So no interactions are necessary.

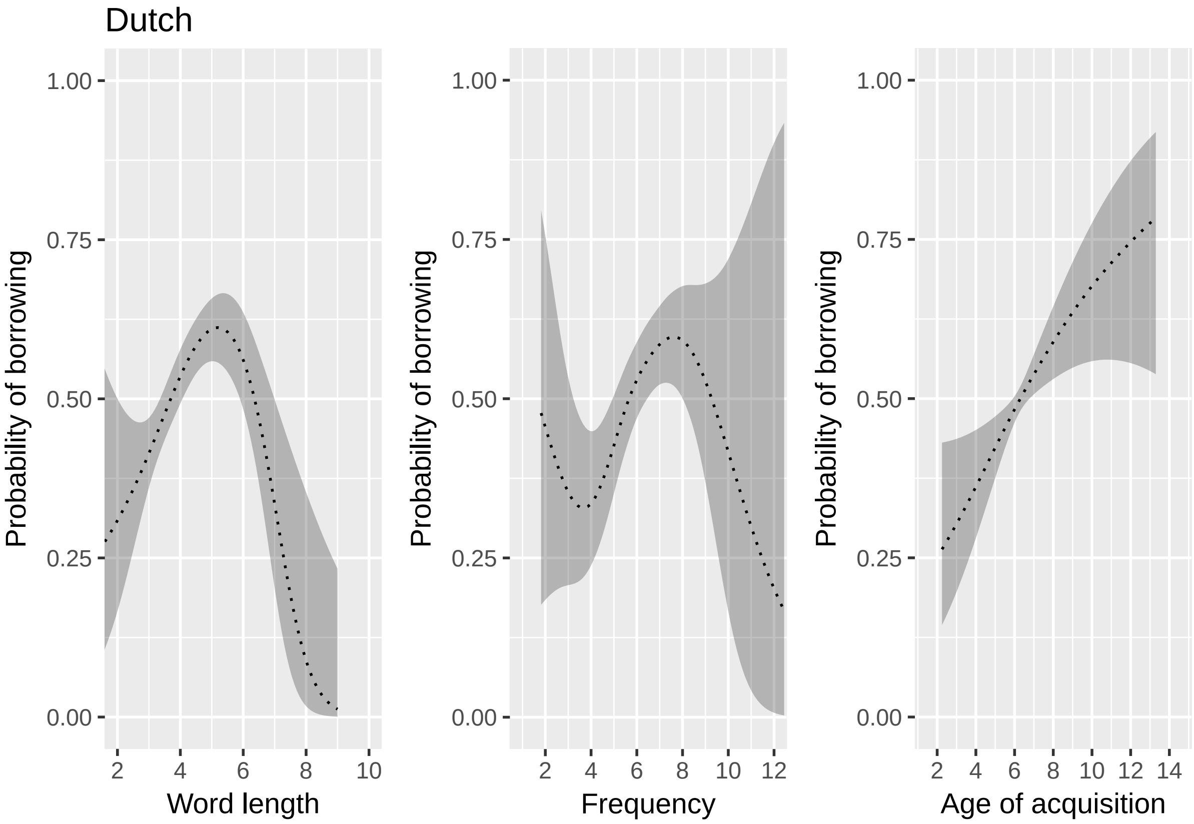
Model estiamtes

Plot the model estimates, changing the dependent scale to probability and the independent variables to their original scales (code is hidden, but available in the Rmd file).

```
plen = rescaleGam(px, 1, dutch$phonlengthscale,"Word length",
                  xlim = c(2,10), breaks = c(2,4,6,8,10)) + ggtitle("Dutch")
paoa = rescaleGam(px, 2, dutch$AoAscale,"Age of acquisition",
                  xlim=c(1.5,14.5),breaks=c(2,4,6,8,10,12,14)) + ggtitle("")
pfreq = rescaleGam(px, 3, dutch$AoAscale,"Frequency",
                   xlim = c(1,12), breaks=c(2,4,6,8,10,12)) + ggtitle("")
pconc = rescaleGam(px, 4, dutch$concscale,"Concreteness") + ggtitle("Dutch")

pconc
```



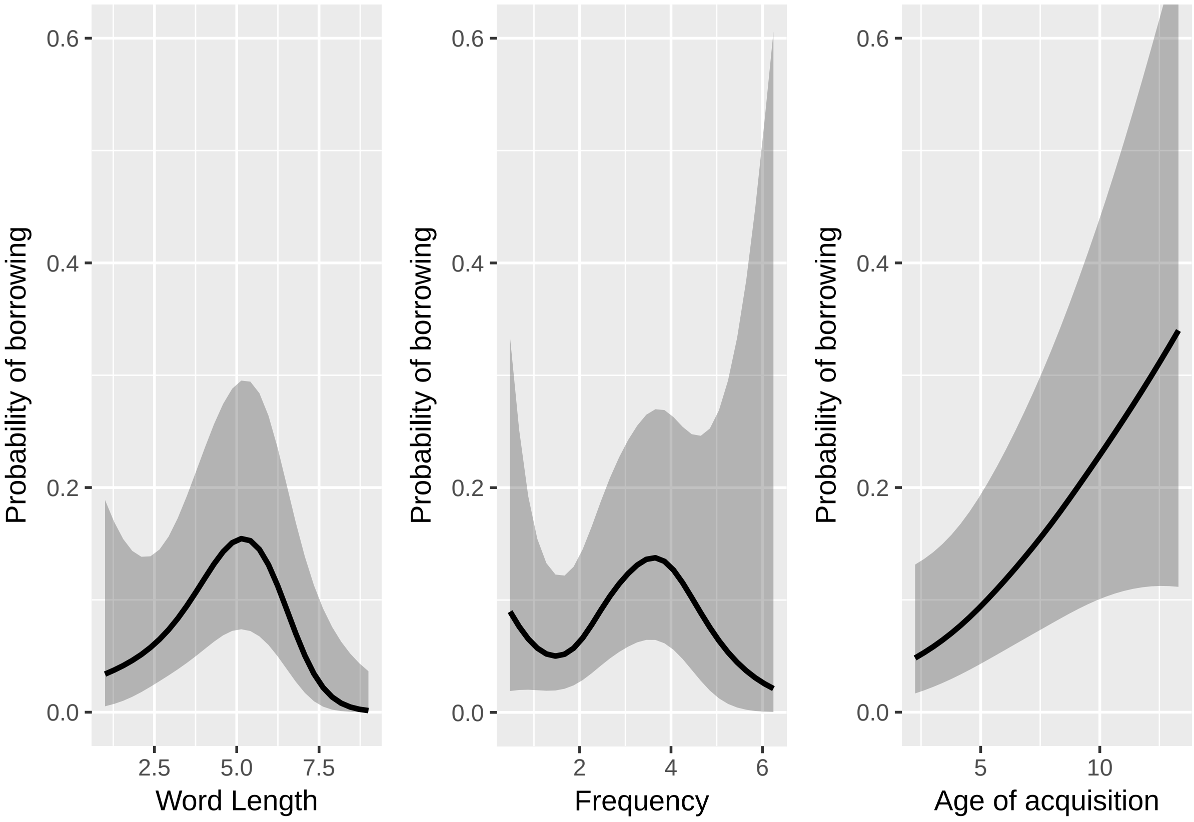
```
grid.arrange(plen,pfreq,paoa, nrow=1)
```



```
pdf(file='../results/graphs/Dutch_ModelResults.pdf',
    height =3,width = 8)
grid.arrange(plen,pfreq,paoa, nrow=1)
dev.off()
```

```
## pdf
##   2
```

Plot the model estimates, removing the influence of the random effects using the library `itsadug` (code is hidden, but available in the Rmd file).



```
## pdf 
##   2
```

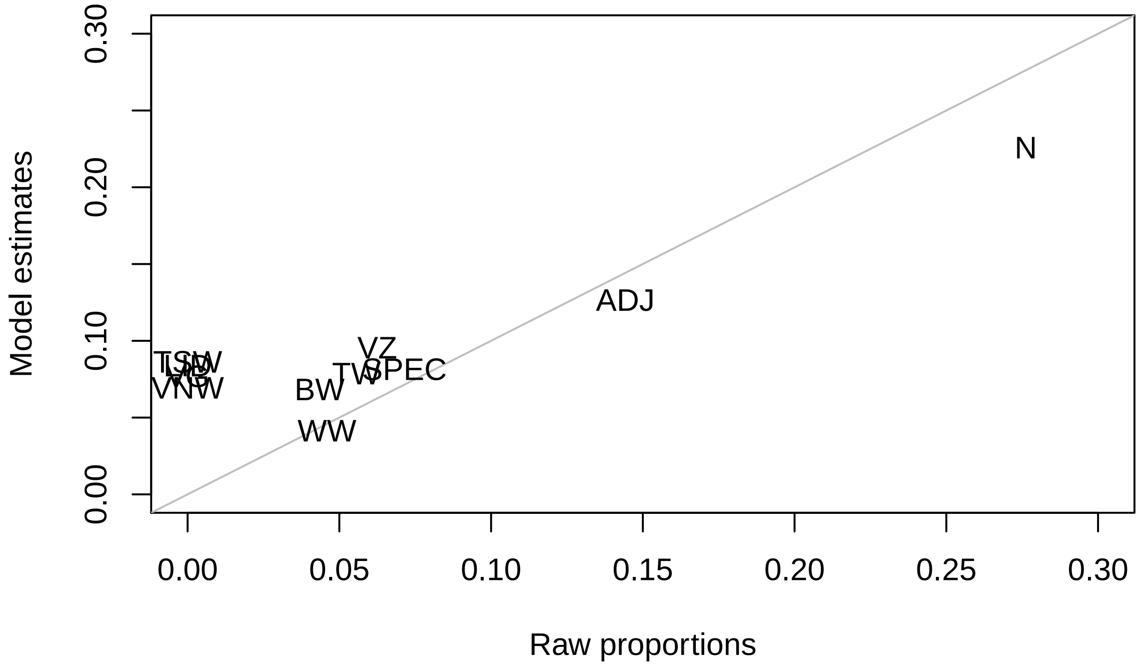
Random effects for Part of speech

```
mc = m0.dutch$coefficients
mc[grepl("s\\(cat\\)",names(mc))]
```

```
##     s(cat).1     s(cat).2     s(cat).3     s(cat).4     s(cat).5 
##  1.118766823  0.418169254 -0.258757264 -0.038467724 -0.069267143 
##     s(cat).6     s(cat).7     s(cat).8     s(cat).9    s(cat).10 
## -0.007058009 -0.107870765 -0.132337453 -0.242556253  0.104322001 
##    s(cat).11 
## -0.784943467
```

```
raw = tapply(dutch$bor15,dutch$cat,mean)
model.est = logit2per(m0.dutch$coefficients[1] +
              mc[grepl("s\\(cat\\)",names(mc))])

plot(raw, model.est,
  xlab="Raw proportions",
  ylab="Model estimates",
  col="white",
     ylim=c(0,0.3),
     xlim=c(0,0.3))
abline(0,1,col='gray')
```

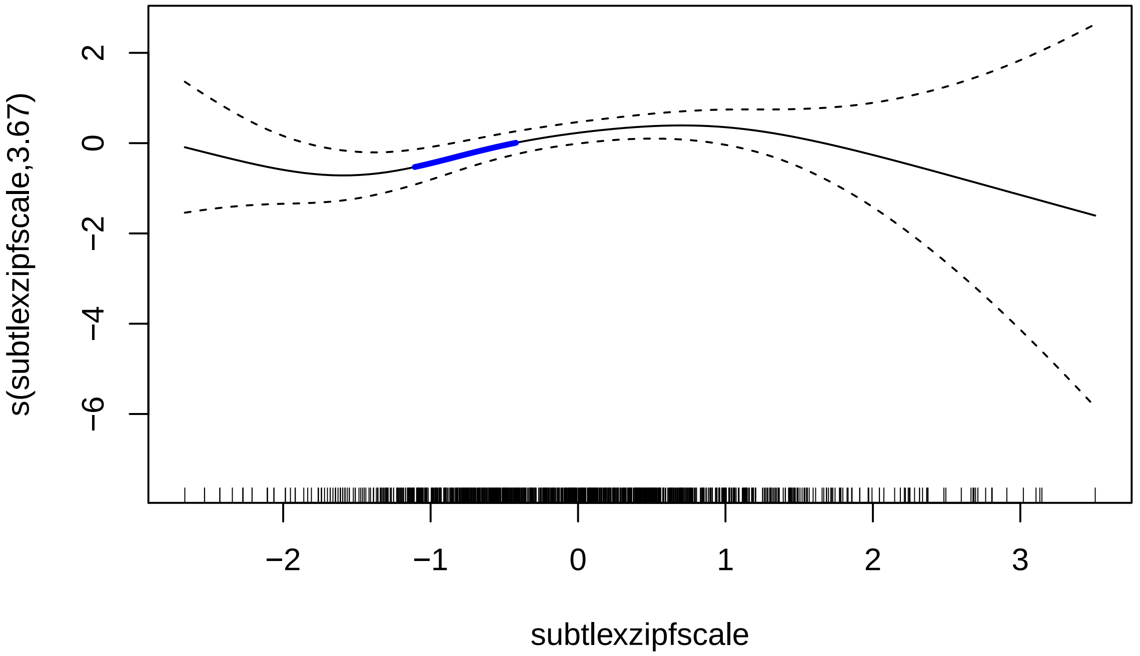
```
text(raw, model.est, names(raw))
```



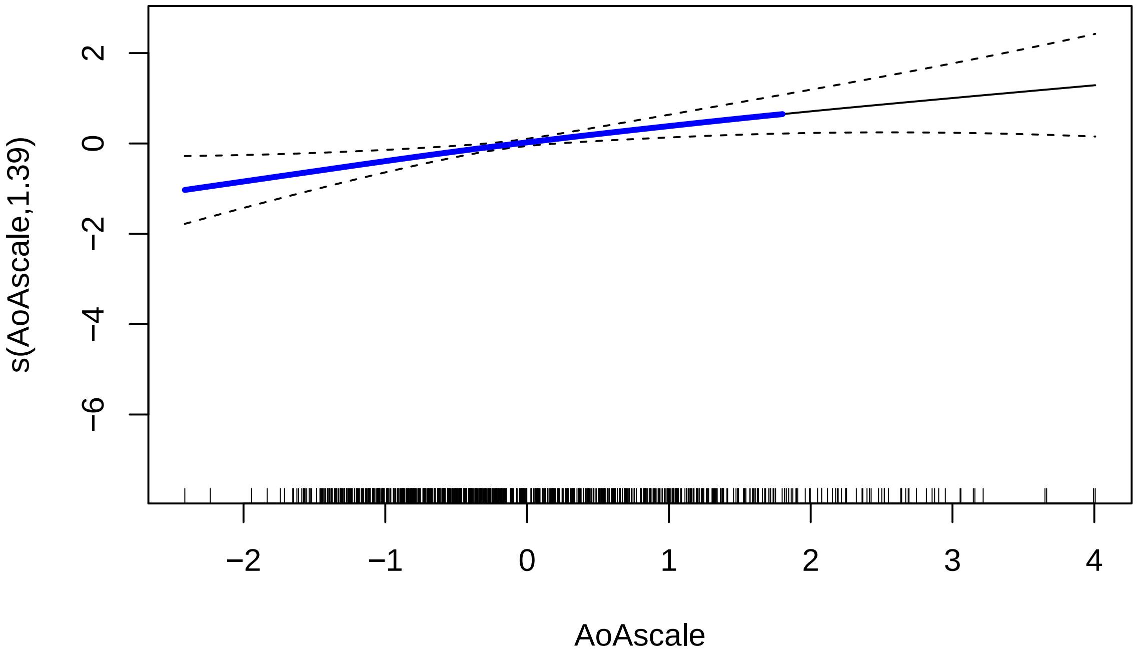
Derivatives test

The plots below highlight which sections of the GAM splines are significantly increasing or decreasing.

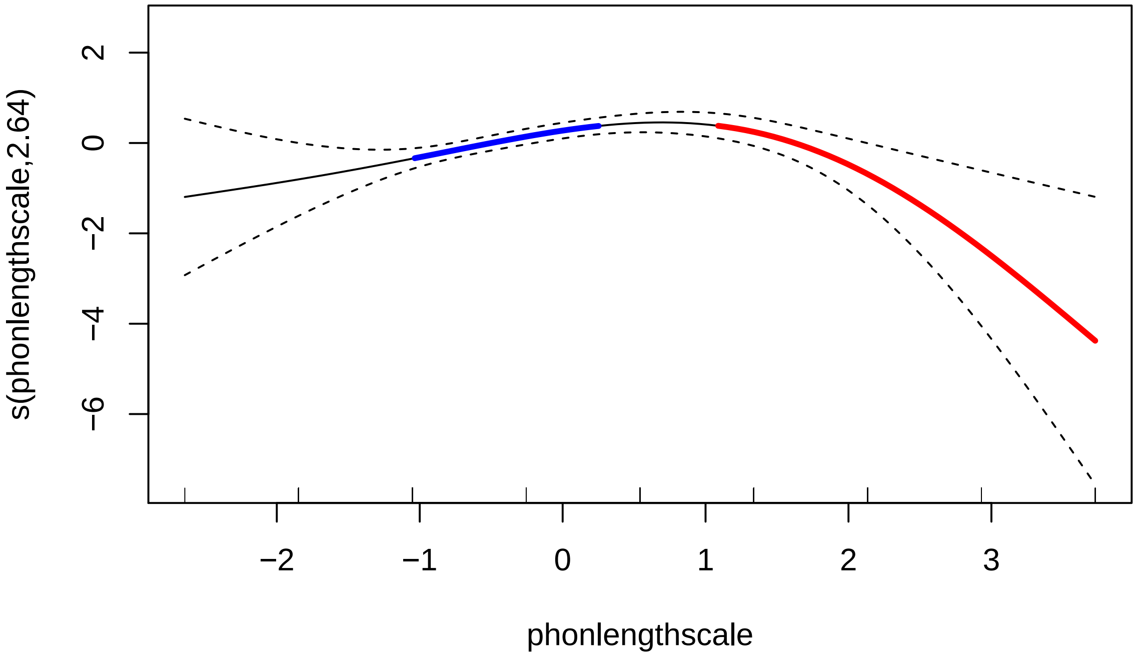
```
pSigFreq = plotGAMSignificantSlopes(m0.dutch,"subtlexzipfscale","Frequency")
```



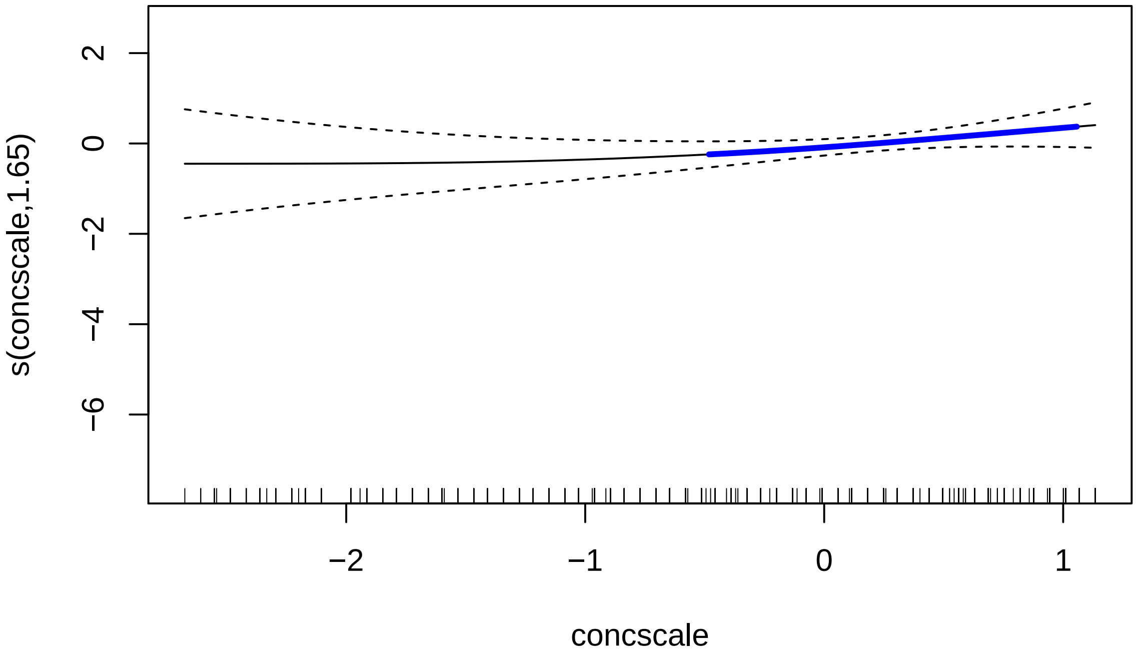
```
pSigAoA = plotGAMSignificantSlopes(m0.dutch,"AoAscale","AoA")
```

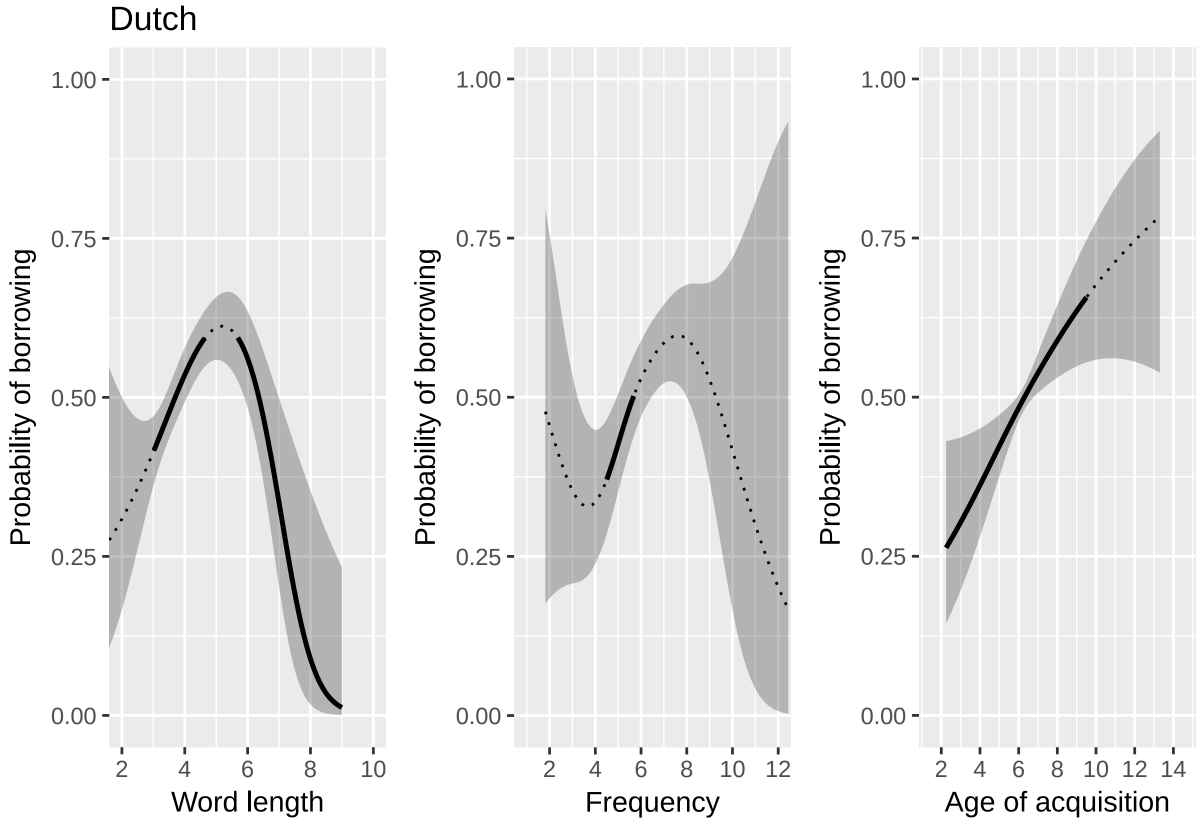


```
pSigLen = plotGAMSignificantSlopes(m0.dutch,"phonlengthscale","length")
```



```
pSigConc = plotGAMSignificantSlopes(m0.dutch,"concscale","Concreteness")
```


2
0
−2
−4
−6
s(concscale,1.65)
−2
−1
0
1
concscale


Dutch
1.00
0.75
0.50
0.25
0.00
Probability of borrowing
2
4
6
8
10
Word length
1.00
0.75
0.50
0.25
0.00
Probability of borrowing
2
4
6
8
10
12
Frequency
1.00
0.75
0.50
0.25
0.00
Probability of borrowing
2
4
6
8
10
12
14
Age of acquisition

```
## pdf
##   2
```

Number of morphemes

The big difference between English and Dutch is the effect of word length. This could reflect the role of agglutination in Dutch, where longer words may be composed of multiple morphemes. We can allow the effect of length to vary by whether the word is monomorphemic or not.

```
dutch$morphology = "other"
dutch$morphology[grepl("monomorphemic",dutch$celex.morphology)] = "monomorphemic"
#dutch$morphology[grepl("complex",dutch$celex.morphology)] = "complex"
dutch$morphology = factor(dutch$morphology)
```

```
mMorph.dutch = bam(bor15.cat ~
      s(phonlengthscale, by=morphology, k=4) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dutch,
    family='binomial')
lrtest(m0.dutch,mMorph.dutch)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, by = morphology, k = 4) + s(AoAscale) +
##     s(subtlexzipfscale) + s(concscale) + s(cat, bs = "re") +
##     s(cat, phonlengthscale, bs = "re") + s(cat, AoAscale, bs = "re") +
##     s(cat, subtlexzipfscale, bs = "re") + s(cat, concscale, bs = "re")
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 19.272 -439.79
## 2 20.382 -435.02 1.1094 9.5326   0.002019 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

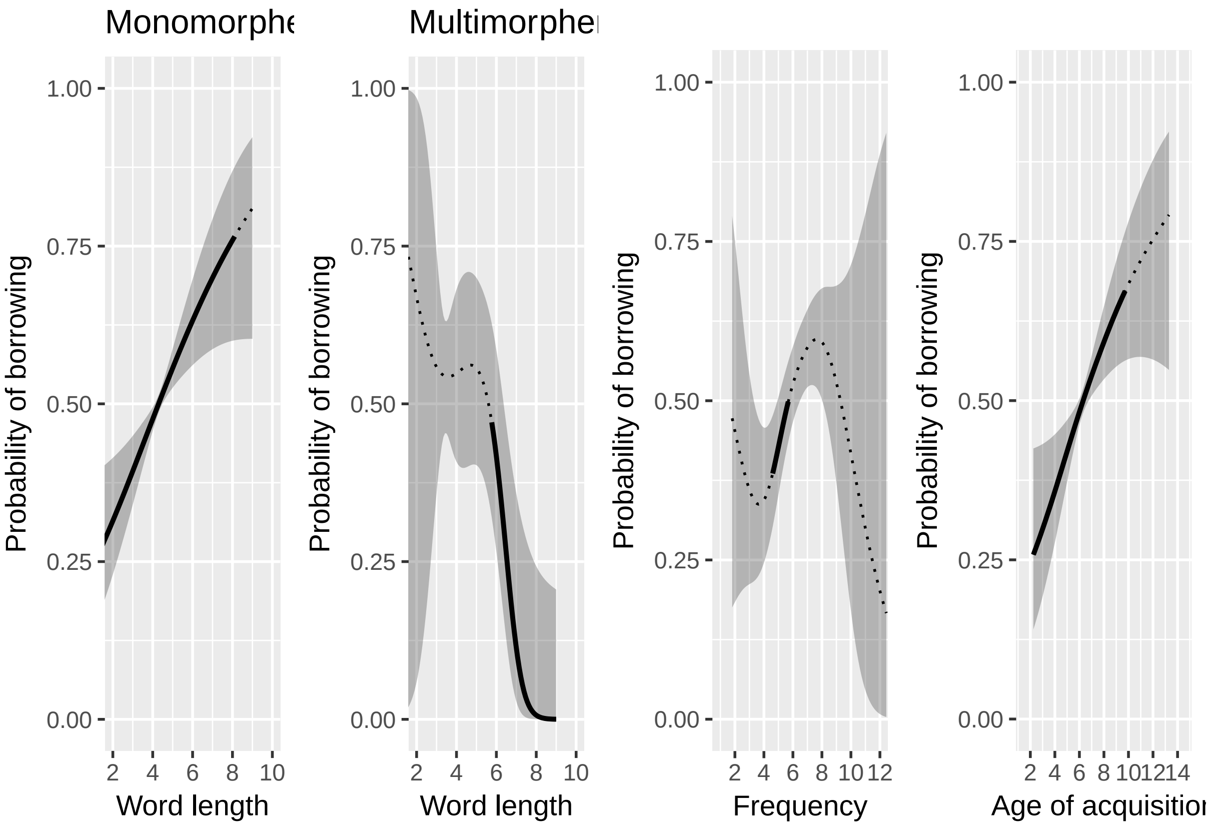
Adding an interaction with morphology signficantly imporves the fit of the model. The model stats are very similar to the model above:

```
summary(mMorph.dutch)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale, by = morphology, k = 4) + s(AoAscale) +
##     s(subtlexzipfscale) + s(concscale) + s(cat, bs = "re") +
##     s(cat, phonlengthscale, bs = "re") + s(cat, AoAscale, bs = "re") +
##     s(cat, subtlexzipfscale, bs = "re") + s(cat, concscale, bs = "re")
##
```

```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.2209     0.3819  -5.815 6.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                              edf Ref.df Chi.sq
## s(phonlengthscale):morphologymonomorphemic 1.087e+00  1.167 14.815
## s(phonlengthscale):morphologyother        2.202e+00  2.516  9.202
## s(AoAscale)                               1.384e+00  1.679 11.954
## s(subtlexzipfscale)                       3.579e+00  4.504 11.253
## s(concscale)                              1.664e+00  2.063  2.368
## s(cat)                                    3.689e+00 10.000 39.371
## s(cat,phonlengthscale)                    1.246e-05 10.000  0.000
## s(cat,AoAscale)                           3.387e-06 10.000  0.000
## s(cat,subtlexzipfscale)                   7.712e-06 10.000  0.000
## s(cat,concscale)                          1.568e+00 10.000  5.177
##                                            p-value
## s(phonlengthscale):morphologymonomorphemic 0.000313 ***
## s(phonlengthscale):morphologyother        0.058796 .
## s(AoAscale)                               0.004962 **
## s(subtlexzipfscale)                       0.028609 *
## s(concscale)                              0.330308
## s(cat)                                     6.2e-08 ***
## s(cat,phonlengthscale)                    0.475597
## s(cat,AoAscale)                           1.000000
## s(cat,subtlexzipfscale)                   0.625217
## s(cat,concscale)                          0.054537 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.12   Deviance explained = 14.6%
## fREML = 1420.9  Scale est. = 1         n = 1028
```

Plot the results from the new model. Monomorphemic words have an almost linear positive relationship between word length and probability of borrowing, just like English.


Monomorphe
Multimorphe
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
2 4 6 8 10
Word length
2 4 6 8 10
Word length
2 4 6 8 10 12
Frequency
2 4 6 8 10 12 14
Age of acquisitio

```
## pdf
##   2
```

```
## pdf
##   2
```

4 Combined English and Dutch analysis

Combined English and Dutch Analysis

```
library(mgcv)
library(lmtest)
# Code for assessing significance of GAM slopes
source("GAM_derivaties.R")
```

Introduction

Below are the main results for English and Dutch. This analysis tests whether the differences are significant by combining the data and adding a random effect for language.

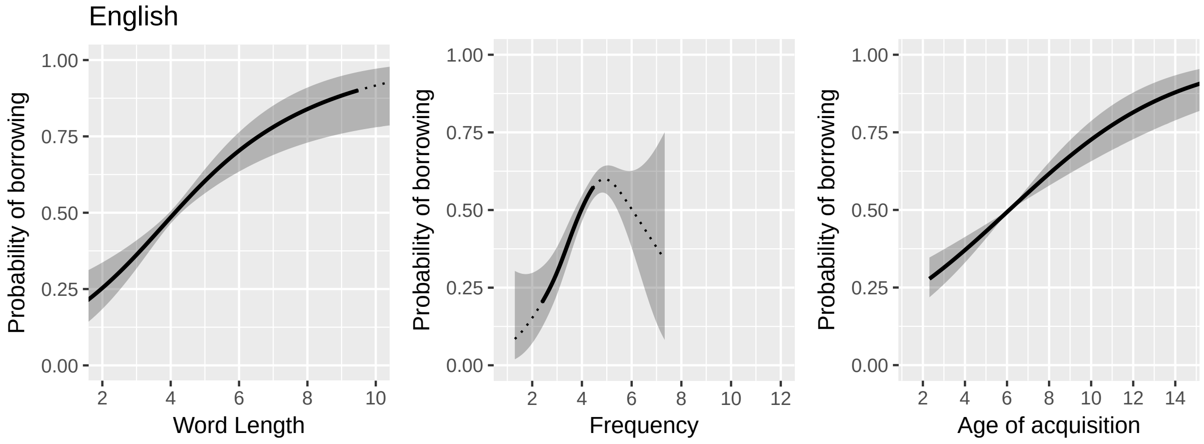


Figure 1: English

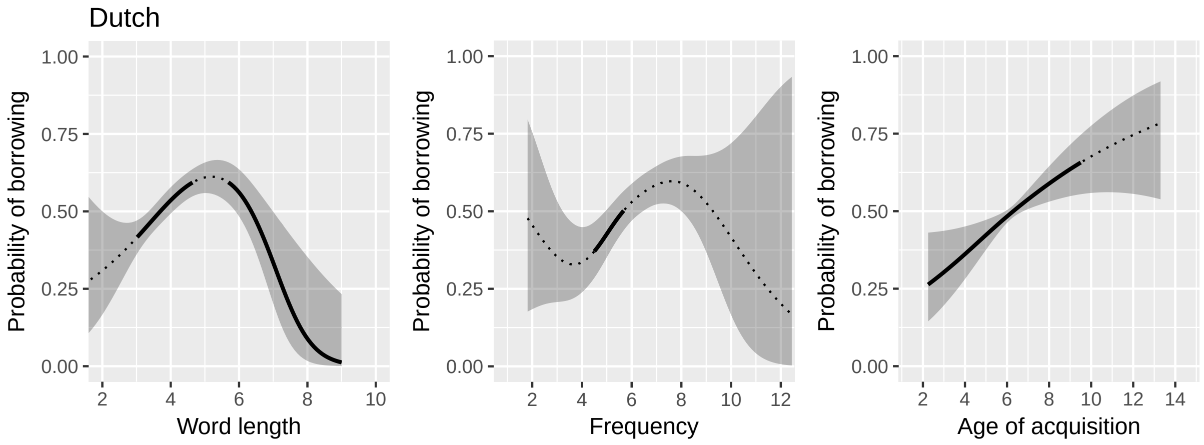


Figure 2: Dutch

Load data

Dutch data:

```
load("../data/loanwords_Dutch.Rdat")
dutch$subtlexzipf =  log10(as.numeric(dutch$subtlex.dominant.pos.frequency))
```

English data:

```
dataloan <- read.csv("../data/loanword12.csv",stringsAsFactors = F)
dataloan$bor15 <- ifelse(dataloan$borrowing==1,1, ifelse(dataloan$borrowing==5,0,NA))
dataloan$bor15.cat <- factor(dataloan$bor15)
dataloan$conc = as.numeric(dataloan$conc)
dataloan$AoA = as.numeric(dataloan$AoA)
dataloan$subtlexzipf = as.numeric(dataloan$subtlexzipf)
```

Make Dutch data frame compatible with English:

```
dataloan$language = "English"
dutch$language = "Dutch"

names(dutch)[names(dutch)=="spelling"] = "word"
names(dutch)[names(dutch)=="length"] = "phonlength"
names(dutch)[names(dutch)=="aoa"] = "AoA"
names(dutch)[names(dutch)=="concreteness"] = "conc"

dutchToEngPos = c(
  N="Noun",
  ADJ="Adjective",
  WW="Verb",
  SPEC="Name",
  VZ="Preposition",
  TW="Number",
  BW="Adverb",
  VG="Cobjuction",
  LID="Interjection",
  TSW="Article",
  VNW="Pronoun")

dutch$cat = dutchToEngPos[as.character(dutch$cat)]
dutch$cat = factor(dutch$cat)
dutch$cat = relevel(dutch$cat,"Noun")
```

Combine data:

```
cols = c("word","bor15.cat","phonlength",
         "AoA", "subtlexzipf",
         "conc","cat","language")

EnglishAndDutch =
  rbind(
    dutch[,cols],
    dataloan[,cols]
  )
EnglishAndDutch$language = factor(EnglishAndDutch$language)
```

```
EnglishAndDutch = EnglishAndDutch[complete.cases(EnglishAndDutch),]
```

Scale and center (note that the data is now centered according to the combined mean):

```
EnglishAndDutch$AoAscale <- scale(EnglishAndDutch$AoA)

EnglishAndDutch$subtlexzipfscale <- scale(EnglishAndDutch$subtlexzipf)

phonlength.center = median(EnglishAndDutch$phonlength)
EnglishAndDutch$phonlengthscale <-
  EnglishAndDutch$phonlength - phonlength.center
phonlength.scale = sd(EnglishAndDutch$phonlengthscale)
EnglishAndDutch$phonlengthscale = EnglishAndDutch$phonlengthscale/phonlength.scale

attr(EnglishAndDutch$phonlengthscale,"scaled:scale") = phonlength.scale
attr(EnglishAndDutch$phonlengthscale,"scaled:center") = phonlength.center

EnglishAndDutch$concscale <- scale(EnglishAndDutch$conc)
conc.scale = attr(EnglishAndDutch$concscale,"scaled:scale")
conc.center = attr(EnglishAndDutch$concscale,"scaled:center")

EnglishAndDutch$cat = relevel(EnglishAndDutch$cat,"Noun")
```

Test of differences between languages

Baseline model with random effect for language:

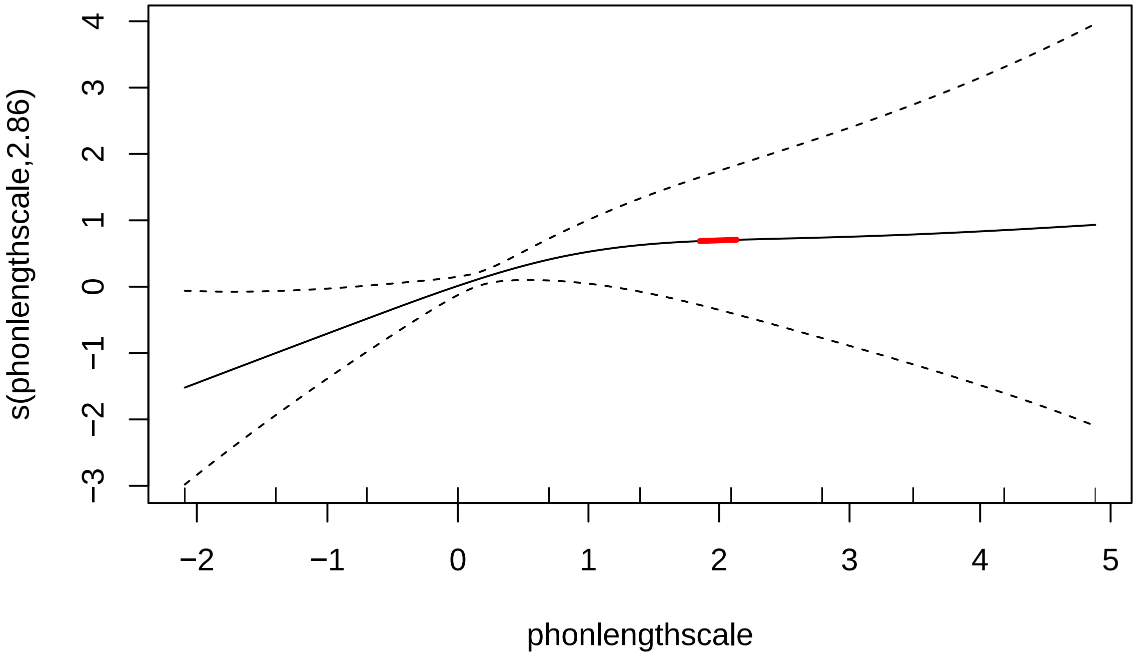
```
m0 = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re') +
      s(language,bs='re'),
    data = EnglishAndDutch,
    family='binomial')
```

Word Length

```
mLen = update(m0, ~. + s(language, phonlengthscale,bs="re"))
lrtest(m0,mLen)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re") +
##     s(language, phonlengthscale, bs = "re")
##      #Df  LogLik     Df Chisq Pr(>Chisq)
## 1 29.566 -1229.5
## 2 30.585 -1220.3 1.0183 18.54  1.663e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plotGAMSignificantSlopes(mLen,"phonlengthscale","Length")
```

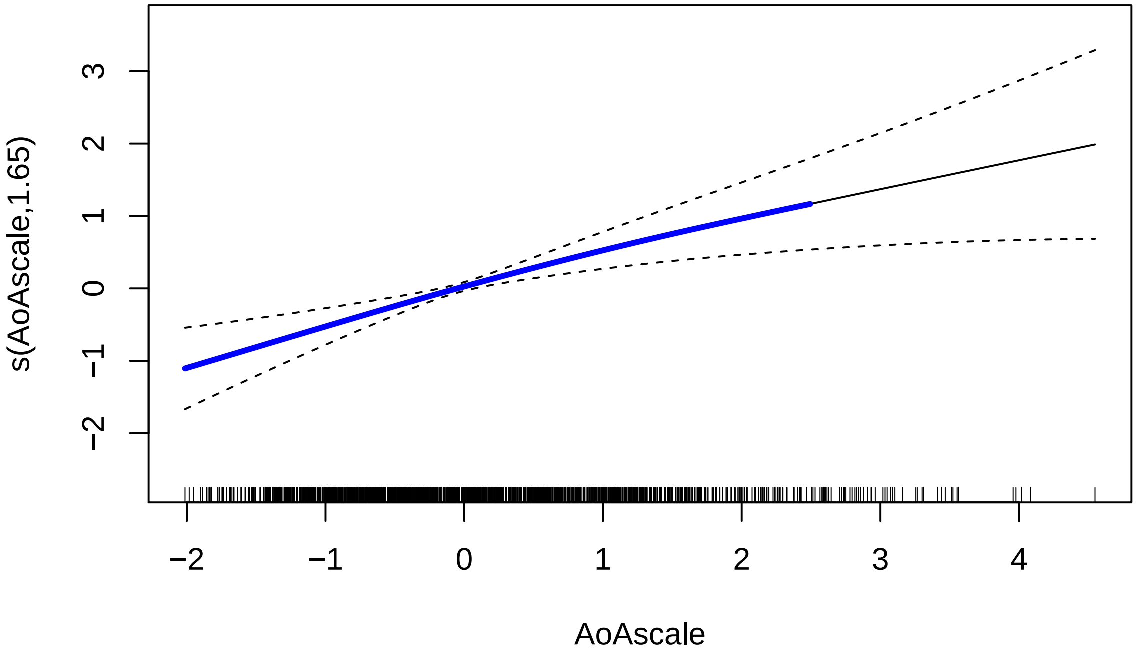


AoA

```
mAoA = update(m0, ~. + s(language, AoAscale,bs="re"))
lrtest(m0,mAoA)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re") +
##     s(language, AoAscale, bs = "re")
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 29.566 -1229.5
## 2 31.272 -1229.1 1.7054 0.8659     0.6486
```

```
plotGAMSignificantSlopes(mAoA,"AoAscale","AoA")
```

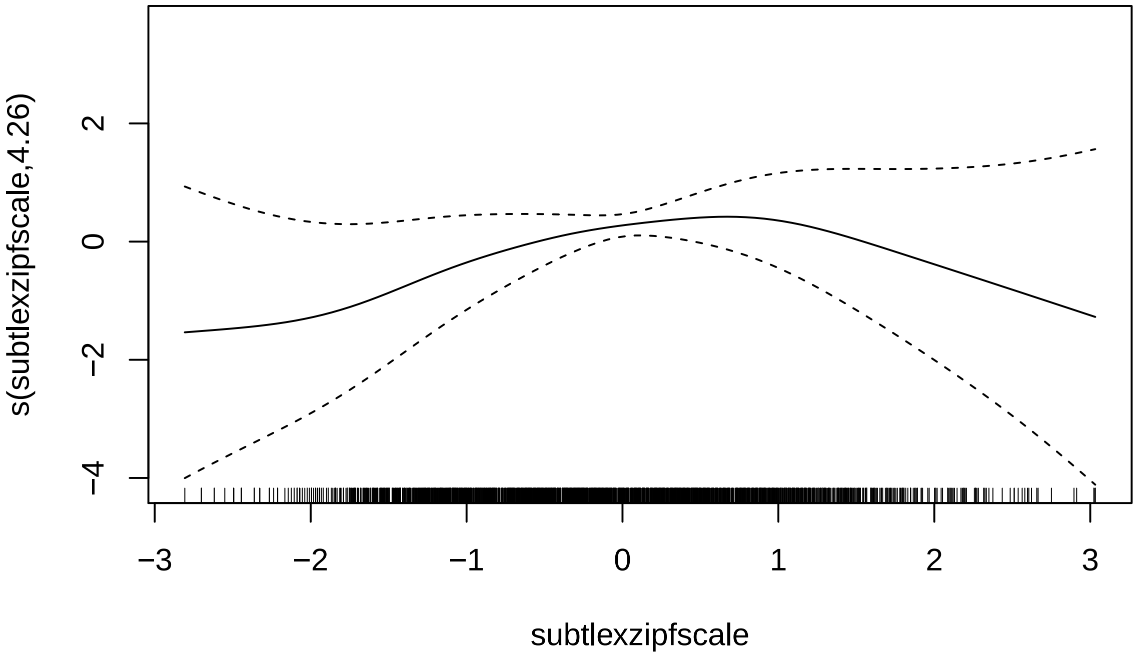


Frequency

```
mFreq = update(m0, ~. + s(language, subtlexzipfscale,bs="re"))
lrtest(m0,mFreq)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re") +
##     s(language, subtlexzipfscale, bs = "re")
##      #Df  LogLik       Df  Chisq Pr(>Chisq)
## 1 29.566 -1229.5
## 2 29.301 -1224.9 -0.26555 9.2611  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot.gam(mFreq,select=3)
```

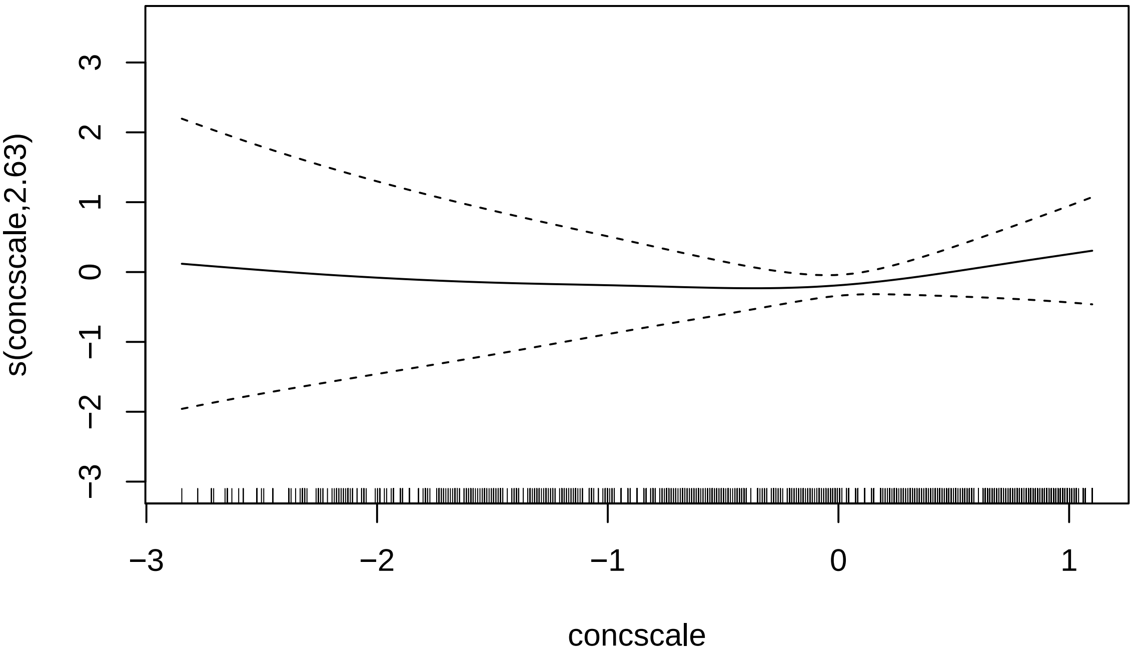


Concreteness

```
mConc = update(m0, ~. + s(language, concscale,bs="re"))
lrtest(m0,mConc)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re") + s(language, bs = "re") +
##     s(language, concscale, bs = "re")
##      #Df  LogLik        Df  Chisq Pr(>Chisq)
## 1 29.566 -1229.5
## 2 29.522 -1215.0 -0.043985 29.048  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot.gam(mConc,select=4)
```



Summary

The effect of age of acquisition on borrowing does not differ by language. The effect of word length and frequency on borrowing does differ by language. Adding an interaction between language and concreteness significantly improves the model, suggesting that the relationship between concreteness and borrowing differs by language. However, in the latter case, one should note that there was no significant main effect of concreteness in the main analysis for each language.

5 Rates of change (study 3)

Cognitive influences in language evolution: Rates of change

Contents

Introduction

We test whether rates of lexical replacement can be predicted by age of acuisition. The key variables in the dataset `pagel/loanword9.csv` are:

- word: orthographic form
- borrowing: whether the word is borrowed into English, according to WOLD
- phonology: phonological form, according to CELEX
- phonlength: number of phonological segments
- AoA: age of acuqisition, according to Kuperman et al. (2012)
- subtlexzipf: frequency from SUBTLEX
- pagel_rate: rate of lexical replacement according to Pagel, Atkinson & Meade (2007)
- cat: part of speech of the word

Pagel, Atkinson & Meade find that lexical replacement rates are higher for less frequent words, though the baseline rate differs by part of speech. We use mixed effects modelling to investigate the effects of length, age of acquisition and concreteness, with random intercepts for part of speech.

Load libraries

```r
library(lme4)
library(sjPlot)
library(ggplot2)
library(gplots)
library(gridExtra)
library(mgcv)
```

Load data

Scale and center all variables:

```
d = read.csv("../data/pagel/loanword9.csv", stringsAsFactors = F)
d$pagel_rate = as.numeric(d$pagel_rate)
```

```
## Warning: NAs introduced by coercion
```

```
d$AoA = as.numeric(d$AoA)
```

```
## Warning: NAs introduced by coercion
```

```
d$subtlexzipf = as.numeric(d$subtlexzipf)
```

```
## Warning: NAs introduced by coercion
```

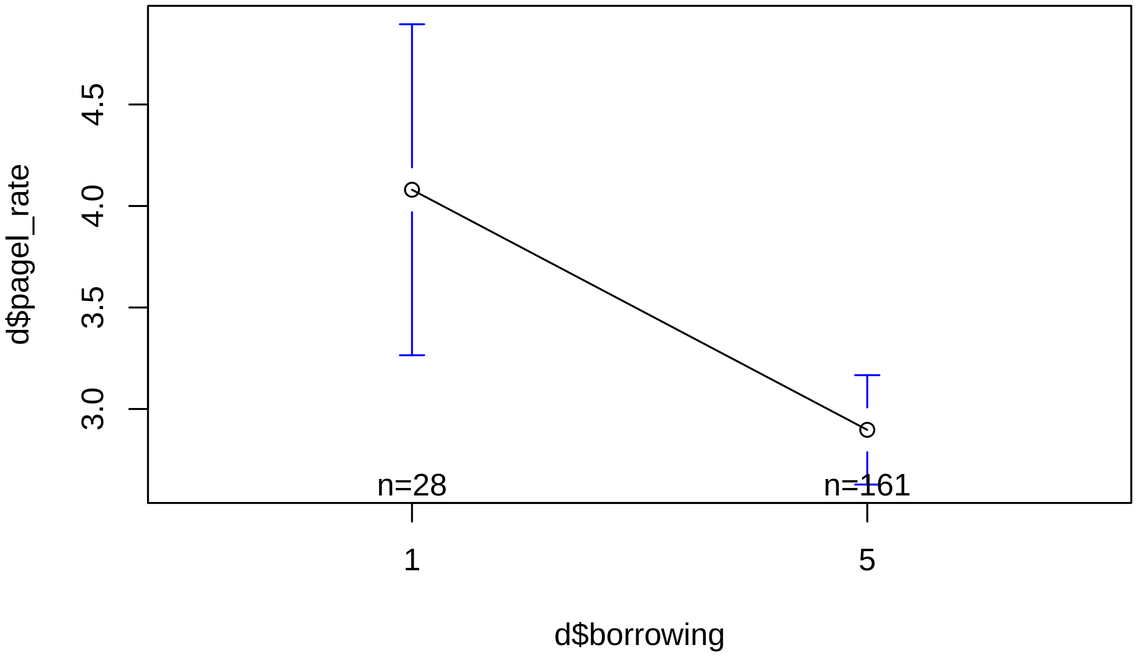
```
d$phonlength = as.numeric(d$phonlength)
d$conc = as.numeric(d$conc)
```

```
## Warning: NAs introduced by coercion
```

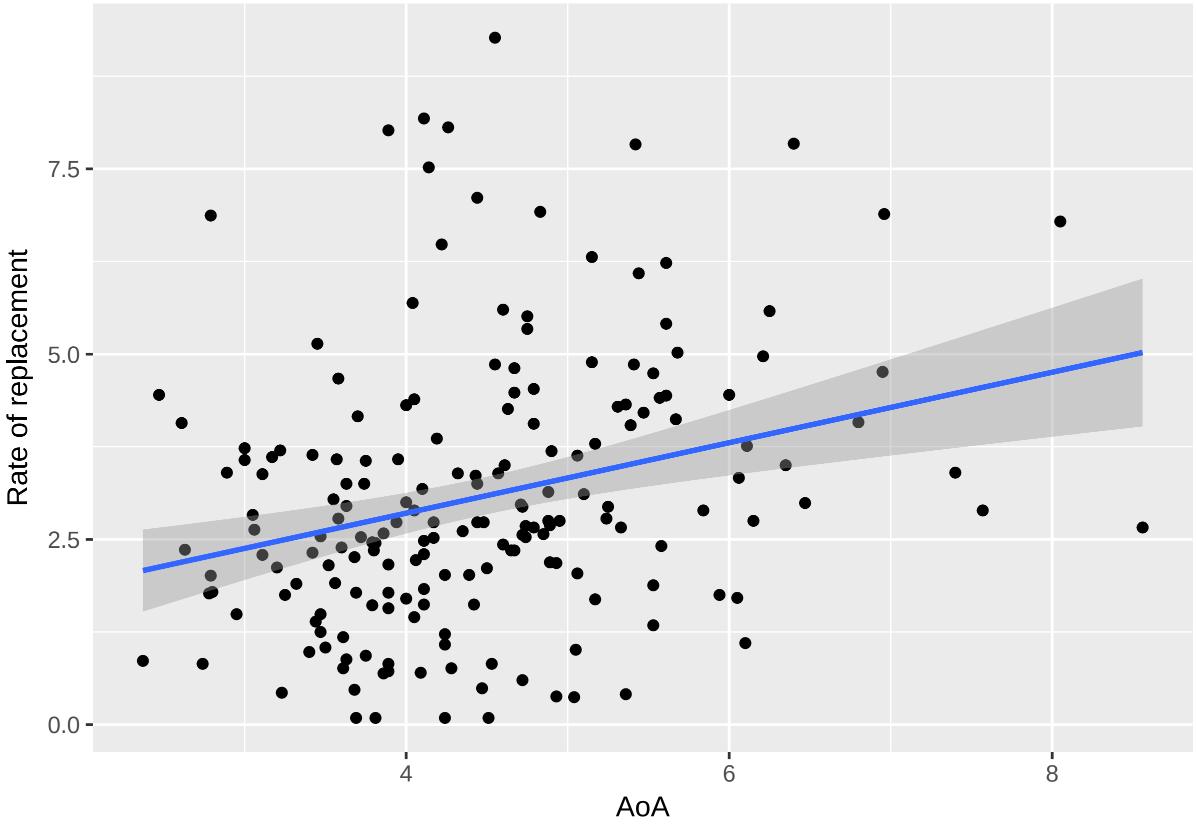
```
# group borrowing confidence
d$borrowing[d$borrowing==2] = 1
d$borrowing[d$borrowing==3] = NA
d$borrowing[d$borrowing==4] = 5
d$borrowing[d$borrowing==7] = NA

# complete cases only
d = d[complete.cases(d[,c("borrowing",'AoA','phonlength','conc','pagel_rate')]),]
```

Plot raw data:

```
plotmeans(d$pagel_rate~d$borrowing)
```



```
ggplot(d, aes(AoA, as.numeric(pagel_rate))) +
  geom_point() + stat_smooth(method="lm") +
  ylab("Rate of replacement")
```



Scale variables for analysis:

```
d$pagel_rate = scale(d$pagel_rate)
d$AoA.scale = scale(d$AoA)
d$subtlexzipf.scale = scale(d$subtlexzipf)
d$phonlength.scale = scale(d$phonlength)
d$conc.scale = scale(d$conc)

# most frequent category as intercept
d$borrowing = factor(d$borrowing,levels=c(5,1),labels = c("no","yes"))
d$borrowing.num = as.numeric(d$borrowing)-1
```

Modelling

Null model:

```
m0 = lmer(pagel_rate ~ 1 + (1|cat), data=d)
```

As expected, borrowed words have a higer rate of change:

```
m1 = lmer(pagel_rate ~
          borrowing +
          (1|cat),
         data = d)
anova(m0,m1)
```

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

## Data: d
## Models:
## m0: pagel_rate ~ 1 + (1 | cat)
## m1: pagel_rate ~ borrowing + (1 | cat)
##    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0  3 528.17 537.90 -261.09   522.17
## m1  4 518.80 531.77 -255.40   510.80 11.371      1   0.000746 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The original model in Pagel et al. found relationships between rate of change, frequency and length:

```
m2 = lmer(pagel_rate ~
          borrowing +
          subtlexzipf.scale +
          (1 | cat),
        data = d)
m3 = lmer(pagel_rate ~
          borrowing +
          subtlexzipf.scale +
          phonlength.scale +
          (1 | cat),
        data = d)
anova(m1,m2,m3)
```

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

## Data: d
## Models:
## m1: pagel_rate ~ borrowing + (1 | cat)
## m2: pagel_rate ~ borrowing + subtlexzipf.scale + (1 | cat)
## m3: pagel_rate ~ borrowing + subtlexzipf.scale + phonlength.scale +
## m3:     (1 | cat)
##    Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m1  4 518.80 531.77 -255.40   510.80
## m2  5 509.09 525.30 -249.55   499.09 11.7106      1  0.0006214 ***
## m3  6 505.04 524.49 -246.52   493.04  6.0557      1  0.0138616 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: pagel_rate ~ borrowing + subtlexzipf.scale + phonlength.scale +
##     (1 | cat)
##    Data: d
##
## REML criterion at convergence: 502.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0268 -0.6162 -0.2251  0.5129  2.7396
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cat      (Intercept) 0.2484   0.4984  
##  Residual             0.7513   0.8668  
## Number of obs: 189, groups:  cat, 10
## 
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)        0.01825    0.20291   0.090
## borrowingyes       0.49831    0.18476   2.697
## subtlexzipf.scale -0.24474    0.08498  -2.880
## phonlength.scale   0.16917    0.06898   2.452
## 
## Correlation of Fixed Effects:
##             (Intr) brrwng sbtlx.
## borrowingys -0.127              
## sbtlxzpf.sc -0.308  0.039       
## phnlngth.sc  0.050 -0.197  0.215
```

Indeed, both length and freuqnecy are significant predictors of rate of change.

We can test whether there are non-linear effects for frequency and length:

```
m3.freq2 = lmer(pagel_rate ~
          borrowing +
          subtlexzipf.scale +
          I(subtlexzipf.scale^2) +
          phonlength.scale +
          (1 | cat),
        data = d)
m3.len2 = lmer(pagel_rate ~
          borrowing +
          subtlexzipf.scale +
          I(phonlength.scale^2) +
          phonlength.scale +
          (1 | cat),
        data = d)
anova(m3, m3.freq2)
```

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

```
## Data: d
## Models:
## m3: pagel_rate ~ borrowing + subtlexzipf.scale + phonlength.scale + 
## m3:     (1 | cat)
## m3.freq2: pagel_rate ~ borrowing + subtlexzipf.scale + I(subtlexzipf.scale^2) + 
## m3.freq2:     phonlength.scale + (1 | cat)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m3        6 505.04 524.49 -246.52   493.04                         
## m3.freq2  7 507.02 529.72 -246.51   493.02 0.0115      1     0.9147
```

```
anova(m3, m3.len2)
```

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

```
## Data: d
## Models:
## m3: pagel_rate ~ borrowing + subtlexzipf.scale + phonlength.scale + 
## m3:     (1 | cat)
## m3.len2: pagel_rate ~ borrowing + subtlexzipf.scale + I(phonlength.scale^2) +
## m3.len2:     phonlength.scale + (1 | cat)
##          Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## m3        6 505.04 524.49 -246.52   493.04
## m3.len2   7 505.16 527.86 -245.58   491.16 1.871      1     0.1714
```

There is no significant non-linear (quadratic) effect of frequency or length.

We can now add age of acquisition:

```
m4 = lmer(pagel_rate ~
           borrowing +
           phonlength.scale +
           subtlexzipf.scale +
           AoA.scale +
          (1 |cat),
         data = d)
anova(m3,m4)
```

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

```
## Data: d
## Models:
## m3: pagel_rate ~ borrowing + subtlexzipf.scale + phonlength.scale +
## m3:     (1 | cat)
## m4: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
## m4:     AoA.scale + (1 | cat)
##    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m3  6 505.04 524.49 -246.52   493.04
## m4  7 500.55 523.24 -243.27   486.55 6.4894      1    0.01085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
##     AoA.scale + (1 | cat)
##    Data: d
##
## REML criterion at convergence: 500.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0850 -0.6595 -0.1920  0.4955  2.8777
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cat      (Intercept) 0.2093   0.4575
##  Residual             0.7338   0.8566
## Number of obs: 189, groups:  cat, 10
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)      -0.06110    0.19282  -0.317
## borrowingyes      0.50080    0.18250   2.744
## phonlength.scale  0.15176    0.06849   2.216
## subtlexzipf.scale -0.15846    0.08969  -1.767
## AoA.scale          0.17481    0.06942   2.518
##
## Correlation of Fixed Effects:
##             (Intr) brrwng phnln. sbtlx.
## borrowingys -0.132
## phnlngth.sc  0.067 -0.196
## sbtlxzpf.sc -0.345  0.038  0.163
## AoA.scale   -0.160  0.005 -0.103  0.372
```

Age of acquisition significantly improves the model and has an effect size similar to length (also, frequency has a much weaker effect).

Test the non-linear effect of age of acquisition:

```
m4b = lmer(pagel_rate ~
           borrowing +
           phonlength.scale +
           subtlexzipf.scale +
           AoA.scale +
           I(AoA.scale^2) +
          (1 |cat),
         data = d)
anova(m4,m4b)
```

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

```
## Data: d
## Models:
## m4: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
## m4:     AoA.scale + (1 | cat)
## m4b: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
## m4b:     AoA.scale + I(AoA.scale^2) + (1 | cat)
##     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m4   7 500.55 523.24 -243.27   486.55
## m4b  8 502.45 528.38 -243.22   486.45 0.1009      1     0.7507
```

No quadratic effect of age of acquisition.

Add concreteness:

```
m5 = lmer(pagel_rate ~
           phonlength.scale +
           subtlexzipf.scale +
           AoA.scale +
           borrowing +
           conc.scale +
            (1|cat),
         data = d)
anova(m4,m5)
```

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

```
## Data: d
## Models:
## m4: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
## m4:     AoA.scale + (1 | cat)
## m5: pagel_rate ~ phonlength.scale + subtlexzipf.scale + AoA.scale +
## m5:     borrowing + conc.scale + (1 | cat)
##    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m4  7 500.55 523.24 -243.27   486.55
## m5  8 501.03 526.96 -242.51   485.03 1.5177      1      0.218
```

Concreteness does not significantly improve the fit of the model.

Final model:

```
finalModel = m4
summary(finalModel)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: pagel_rate ~ borrowing + phonlength.scale + subtlexzipf.scale +
##     AoA.scale + (1 | cat)
##    Data: d
##
## REML criterion at convergence: 500.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0850 -0.6595 -0.1920  0.4955  2.8777
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cat      (Intercept) 0.2093   0.4575
##  Residual             0.7338   0.8566
## Number of obs: 189, groups:  cat, 10
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)       -0.06110    0.19282  -0.317
## borrowingyes       0.50080    0.18250   2.744
## phonlength.scale   0.15176    0.06849   2.216
## subtlexzipf.scale -0.15846    0.08969  -1.767
## AoA.scale          0.17481    0.06942   2.518
##
## Correlation of Fixed Effects:
##             (Intr) brrwng phnln. sbtlx.
## borrowingys -0.132
## phnlngth.sc  0.067 -0.196
## sbtlxzpf.sc -0.345  0.038  0.163
## AoA.scale   -0.160  0.005 -0.103  0.372
```

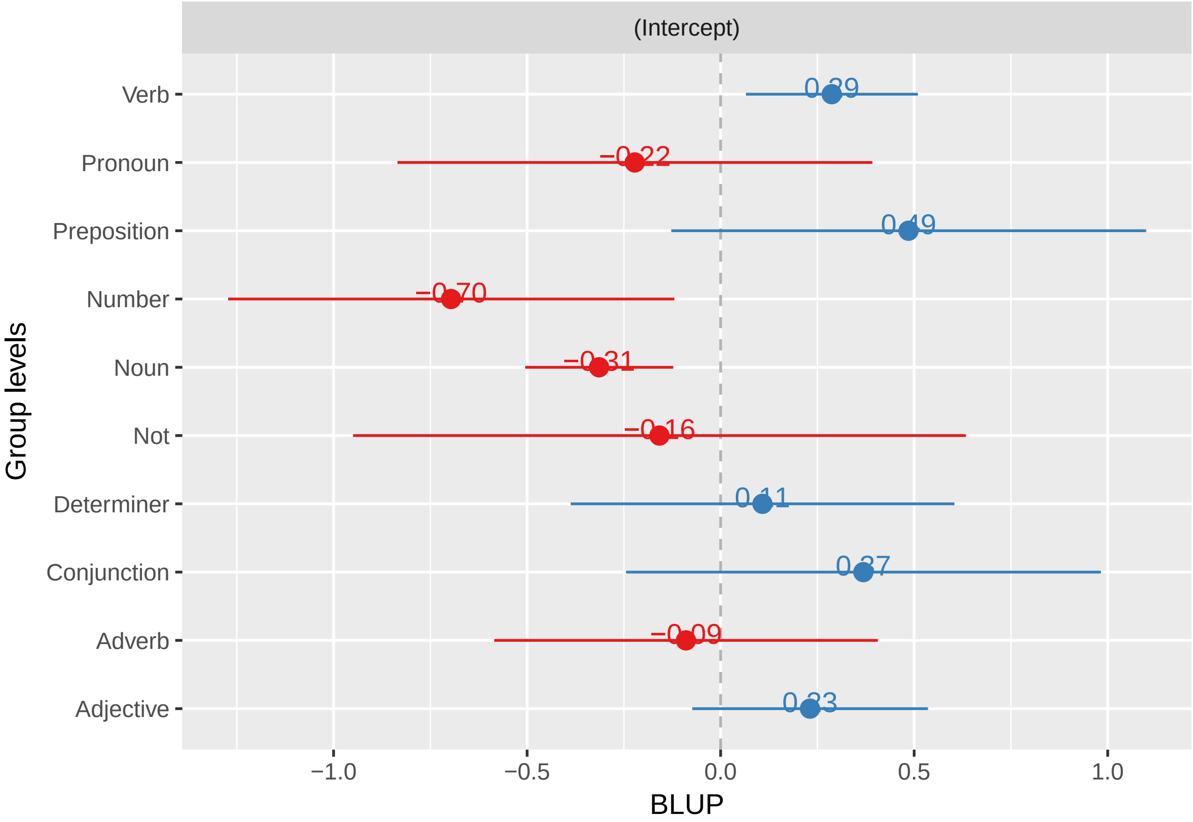
```
# Rough measure of model fit (R2):
cor(predict(finalModel),d$pagel_rate)^2
```

```
##           [,1]
## [1,] 0.3058029
```

Random intercepts for part of speech:

```
sjp.lmer(finalModel)
```

```
## Plotting random effects...
```



Model estimates

```
px =sjp.lmer(finalModel, 'eff', c("AoA.scale","phonlength.scale","subtlexzipf.scale"),
             show.ci = T, prnt.plot = F, facet.grid = F)
```

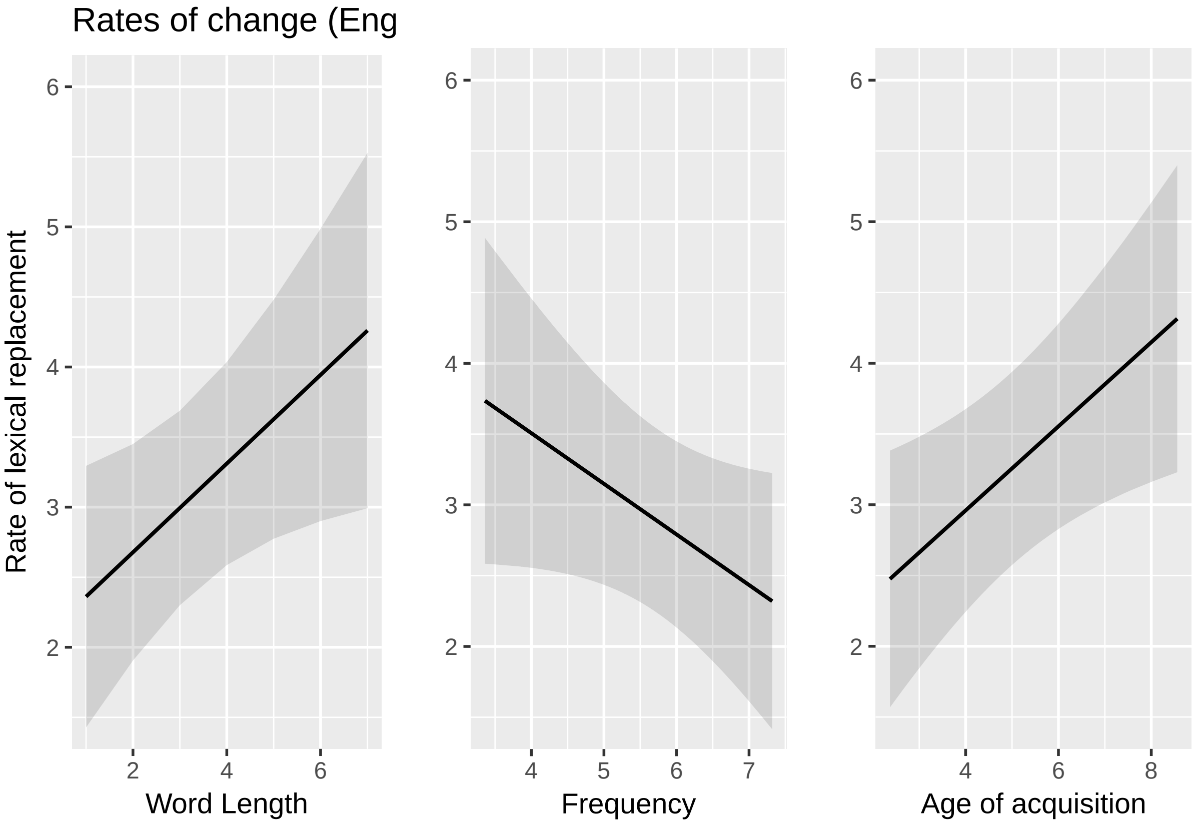
```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

```
pr.sc = attr(d$pagel_rate,"scaled:scale")
pr.cn = attr(d$pagel_rate,"scaled:center")
# length
p1 = px$plot.list[[1]]
p1$data$x = p1$data$x *attr(d$phonlength.scale,"scaled:scale") +
  attr(d$phonlength.scale,"scaled:center")
p1$data$y = p1$data$y * pr.sc + pr.cn
p1$data$lower = p1$data$lower * pr.sc + pr.cn
p1$data$upper = p1$data$upper * pr.sc + pr.cn
p1 = p1 + ggtitle("Rates of change (English)") +
  xlab("Word Length") +
  ylab("")+
  ylab("Rate of lexical replacement")+
  coord_cartesian(ylim=c(1.5,6))

# frequency
p2 = px$plot.list[[2]]
p2$data$x = p2$data$x *attr(d$subtlexzipf.scale,"scaled:scale") +
  attr(d$subtlexzipf.scale,"scaled:center")
```

```r
p2$data$y = p2$data$y * pr.sc + pr.cn
p2$data$lower = p2$data$lower * pr.sc + pr.cn
p2$data$upper = p2$data$upper * pr.sc + pr.cn
p2 = p2 + ggtitle("") +
  xlab("Frequency") +
  ylab("")+
  coord_cartesian(ylim=c(1.5,6))

# Age of acquisition
p3 = px$plot.list[[3]]
p3$data$x = p3$data$x *attr(d$AoA.scale,"scaled:scale") +
  attr(d$AoA.scale,"scaled:center")
p3$data$y = p3$data$y * pr.sc + pr.cn
p3$data$lower = p3$data$lower * pr.sc + pr.cn
p3$data$upper = p3$data$upper * pr.sc + pr.cn
p3 = p3 + ggtitle("") +
  xlab("Age of acquisition") +
  ylab("") +
  coord_cartesian(ylim=c(1.5,6))
pdf(file='../results/graphs/Pagel_RatesOfChange.pdf',
    height =3,width = 8)
grid.arrange(p1,p2,p3, nrow=1)
dev.off()
```

```
## pdf
##   2
```

```r
grid.arrange(p1,p2,p3, nrow=1)
```



Non-linear model

We can test a non-linear model

```
d$cat = factor(d$cat)
d$cat = relevel(d$cat,"Noun")
m0.GAM  = bam(pagel_rate~
    s(phonlength.scale, k=5) +
    s(subtlexzipf.scale) +
    s(AoA.scale) +
    s(cat,bs='re'),
    data = d)
summary(m0.GAM)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## pagel_rate ~ s(phonlength.scale, k = 5) + s(subtlexzipf.scale) +
##     s(AoA.scale) + s(cat, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.008816   0.192210   0.046    0.963
##
```

```
## Approximate significance of smooth terms:
##                          edf Ref.df     F  p-value    
## s(phonlength.scale)  1.236  1.434 6.022   0.0134 *  
## s(subtlexzipf.scale) 1.000  1.000 3.256   0.0728 .  
## s(AoA.scale)         1.000  1.000 6.177   0.0138 *  
## s(cat)               5.609  9.000 3.046 1.95e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.241   Deviance explained = 27.6%
## fREML = 252.99  Scale est. = 0.75926   n = 189
```

The results are very similar, and all edf scores are very close to 1 (linear). Given the small amount of data, we prefer the linear mixed effects model above.

Summary

There was a significant main effect of whether the word is borrowed (log likelihood difference = 5.7 , df = 1 , Chi Squared = 11.37 , p = 0.00075).

There was a significant main effect of frequency (log likelihood difference = 5.9 , df = 1 , Chi Squared = 11.71 , p = 0.00062).

There was a significant main effect of word length (log likelihood difference = 3 , df = 1 , Chi Squared = 6.06 , p = 0.014).

There was a significant main effect of age of acquisition (log likelihood difference = 3.2 , df = 1 , Chi Squared = 6.49 , p = 0.011).

There was no significant main effect of concreteness (log likelihood difference = 0.76 , df = 1 , Chi Squared = 1.52 , p = 0.22).

Rates of change are higher for:

- Borrowed words
- Less frequent words
- Longer words
- Words acquired later in childhood

Replication of Study 1

Predict borrowing by the various psychological predictors. Below we show that only length is significantly related to borrowing in this sample of data.

Linear model:

```
m0.Study1  = glmer(borrowing~
    phonlength.scale +
    subtlexzipf.scale +
    AoA.scale +
    conc.scale +
    (1|cat),
    family="binomial",
    data = d)
summary(m0.Study1)
```

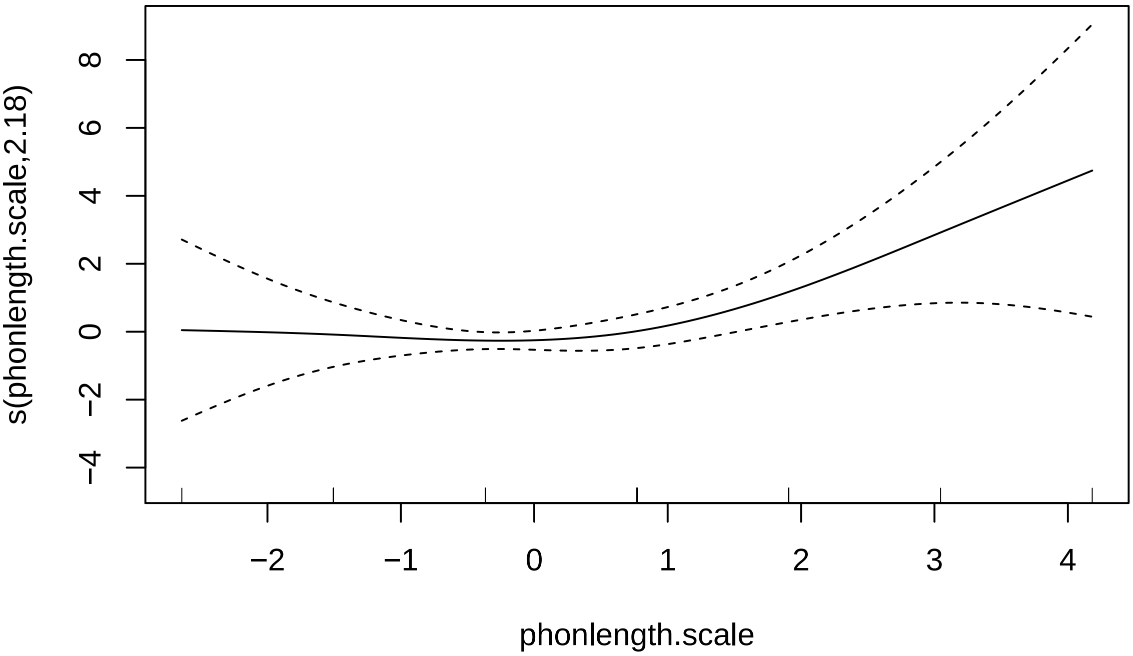
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: borrowing ~ phonlength.scale + subtlexzipf.scale + AoA.scale +
##     conc.scale + (1 | cat)
##    Data: d
##
##      AIC      BIC   logLik deviance df.resid
##    160.3    179.8    -74.2    148.3      183
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8041 -0.4475 -0.3556 -0.2513  5.2844
##
## Random effects:
##  Groups Name        Variance  Std.Dev.
##  cat    (Intercept) 9.371e-18 3.061e-09
## Number of obs: 189, groups:  cat, 10
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.90365    0.23270  -8.181 2.83e-16 ***
## phonlength.scale   0.51430    0.20837   2.468   0.0136 *
## subtlexzipf.scale -0.25450    0.34971  -0.728   0.4668
## AoA.scale         -0.06572    0.23832  -0.276   0.7827
## conc.scale         0.10224    0.33821   0.302   0.7624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) phnln. sbtlx. AA.scl
## phnlngth.sc -0.277
## sbtlxzpf.sc  0.163  0.041
## AoA.scale    0.065 -0.164  0.507
## conc.scale  -0.048 -0.133  0.641  0.473
```

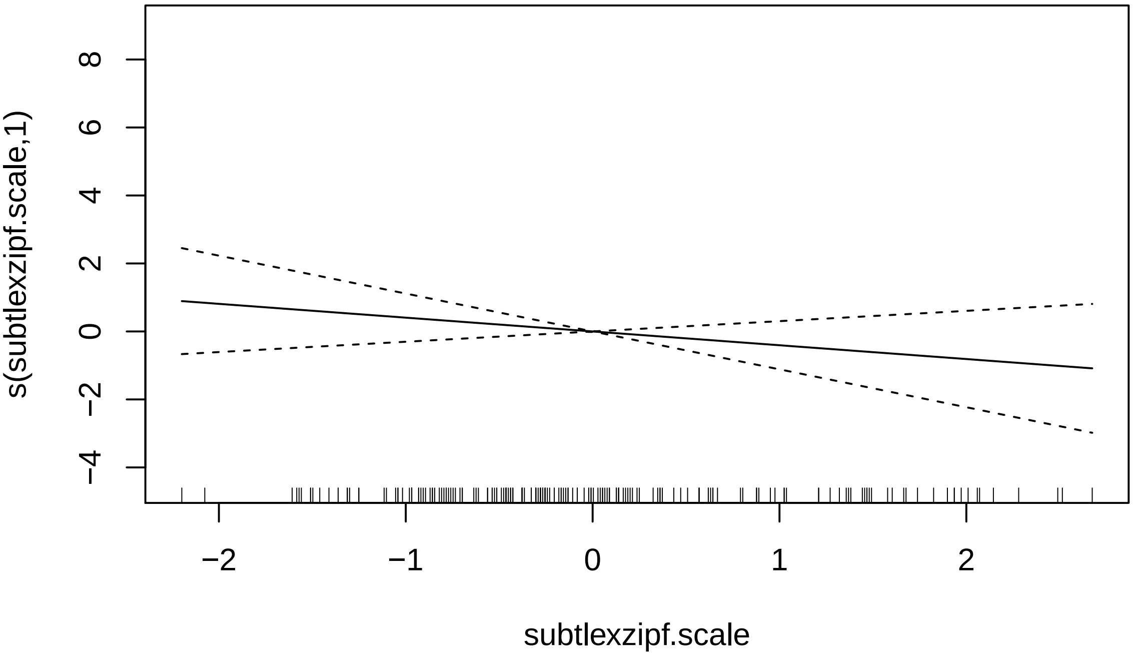
GAM:

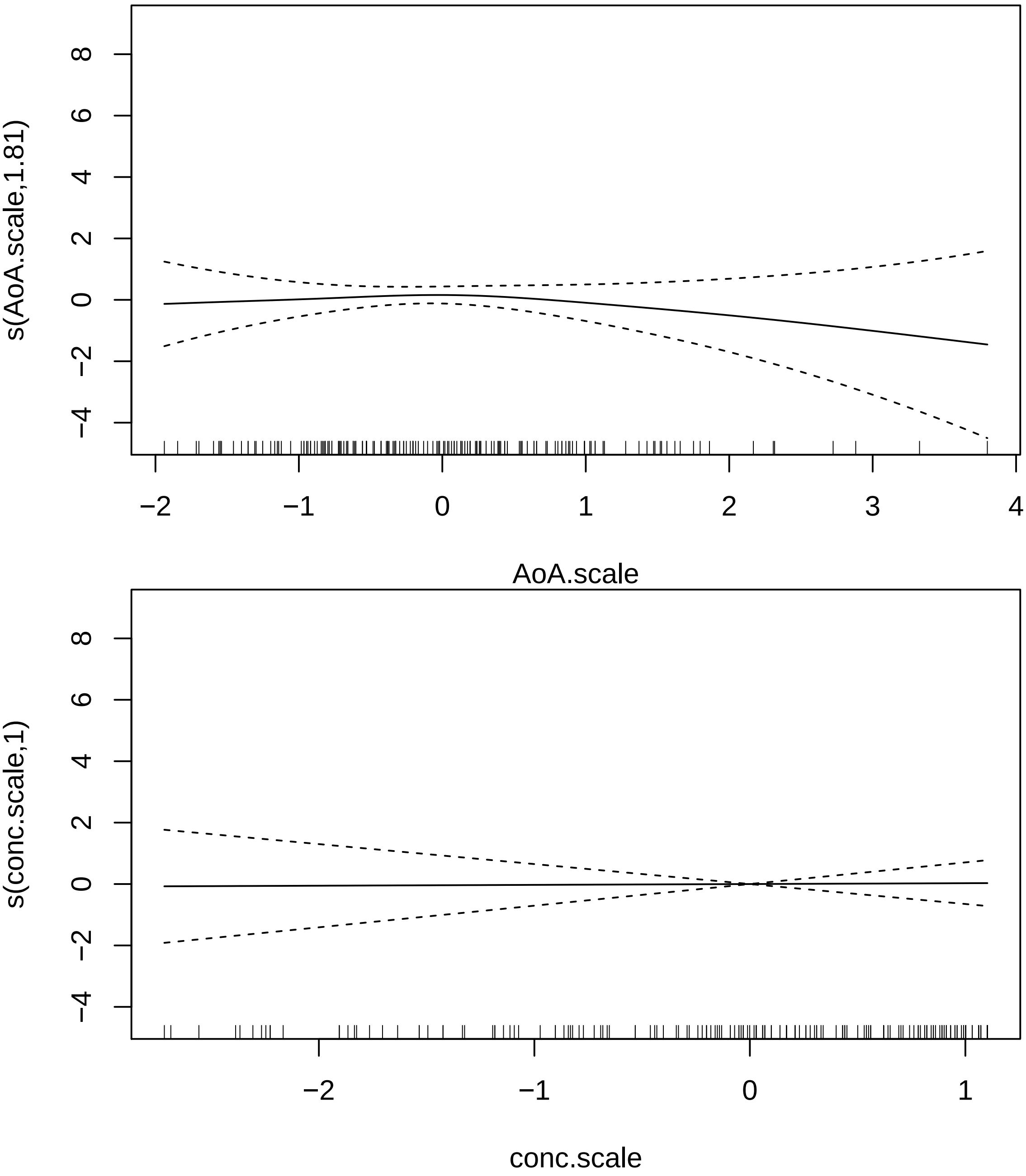
```
m0.Study1.gam  = bam(borrowing~
    s(phonlength.scale, k=5) +
    s(subtlexzipf.scale) +
    s(AoA.scale) +
    s(conc.scale) +
    s(cat,bs='re'),
    family="binomial",
    data = d)
summary(m0.Study1.gam)
```

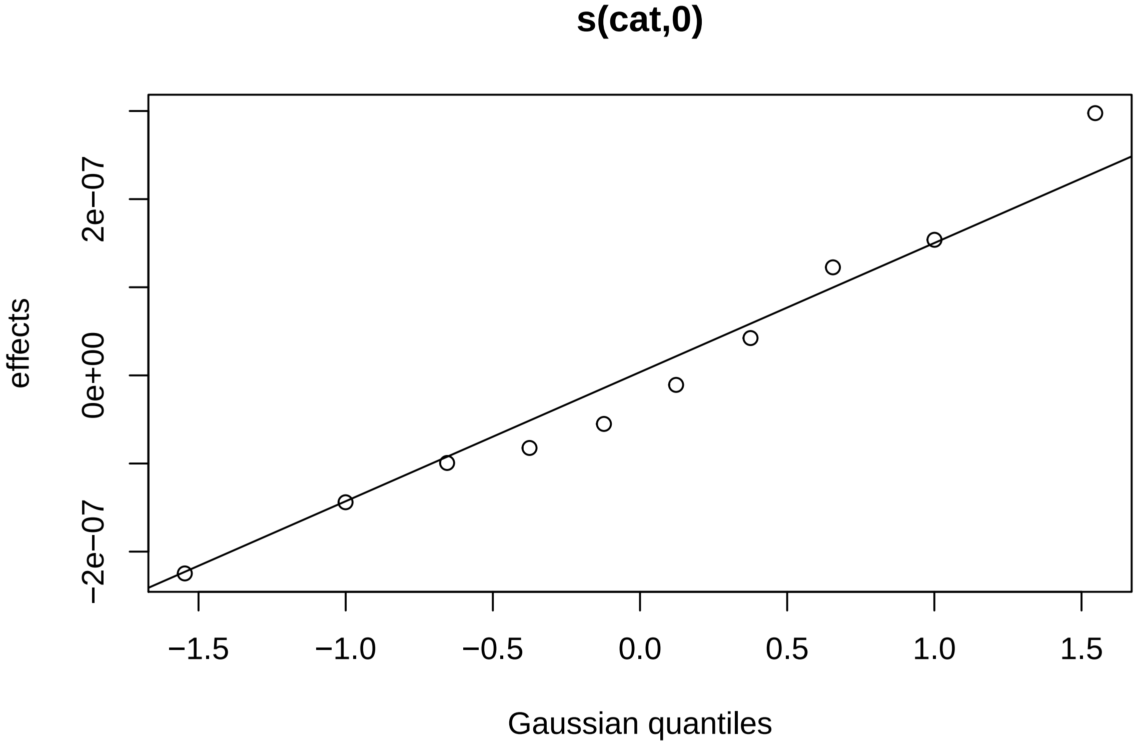
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## borrowing ~ s(phonlength.scale, k = 5) + s(subtlexzipf.scale) +
##     s(AoA.scale) + s(conc.scale) + s(cat, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.894      0.230  -8.236   <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(phonlength.scale)  2.177e+00  2.695  9.416  0.0272 *
## s(subtlexzipf.scale) 1.000e+00  1.000  1.312  0.2521
## s(AoA.scale)         1.805e+00  2.295  1.299  0.5725
## s(conc.scale)        1.000e+00  1.000  0.006  0.9360
## s(cat)               2.314e-06  9.000  0.000  0.6892
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0859   Deviance explained = 11.2%
## fREML =  265.6  Scale est. = 1         n = 189
```

```
plot(m0.Study1.gam)
```


s(phonlength.scale,2.18)
−4
−2
0
2
4
6
8
−2
−1
0
1
2
3
4
phonlength.scale


s(subtlexzipf.scale,1)
−4
−2
0
2
4
6
8
−2
−1
0
1
2
subtlexzipf.scale


s(AoA.scale,1.81)
8
6
4
2
0
−2
−4
−2
−1
0
1
2
3
4
AoA.scale
s(conc.scale,1)
8
6
4
2
0
−2
−4
−2
−1
0
1
conc.scale



References

Kuperman, V., Stadthagen-Gonzalez, H., & Brysbaert, M. (2012). Age-of-acquisition ratings for 30,000 English words. Behavior Research Methods, 44(4), 978-990.

Pagel, M., Atkinson, Q. D., & Meade, A. (2007). Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. Nature, 449(7163), 717-720.

6 Types of borrowing in Old English (study 4)

Cognitive influences in language evolution: Study 4

Contents

Load libraries

```r
library(mgcv)
library(sjPlot)
library(lattice)
library(ggplot2)
library(dplyr)
library(party)
library(lmtest)
library(gridExtra)
library(scales)
library(itsadug)
library(ggfortify)
library(factoextra)
library(gridExtra)
library(reshape2)
library(binom)
```

Extra helper functions:

```r
source("GAM_derivaties.R")
logit2per = function(X){
  return(exp(X)/(1+exp(X)))
}

rescaleGam = function(px, n, xvar, xlab="",breaks=NULL,xlim=NULL){
  y = logit2per(px[[n]]$fit)
  x = px[[n]]$x *attr(xvar,"scaled:scale") + attr(xvar,"scaled:center")
  se.upper = logit2per(px[[n]]$fit+px[[n]]$se)
  se.lower = logit2per(px[[n]]$fit-px[[n]]$se)
  dx = data.frame(x=x,y=y,ci.upper=se.upper,ci.lower=se.lower)
  plen = ggplot(dx, aes(x=x,y=y))+
    geom_ribbon(aes(ymin=ci.lower,ymax=ci.upper), alpha=0.3)+
    geom_line(size=1) +
    xlab(xlab)+
    ylab("Probability of borrowing")
  if(!is.null(breaks)){
    plen = plen + scale_x_continuous(breaks = breaks)
  }
  if(!is.null(xlim)){
   plen = plen + coord_cartesian(ylim = c(0,1),xlim=xlim)
  } else{
    plen = plen + coord_cartesian(ylim = c(0,1))
  }
  return(plen)
}
```

Load data

```
dataloan <- read.csv("../data/loanword12.csv",stringsAsFactors = F)
dataloan$bor15 <- ifelse(dataloan$borrowing==1,1, ifelse(dataloan$borrowing==5,0,NA))
dataloan$bor15.cat <- factor(dataloan$bor15)

dataloan$subtlexzipf = as.numeric(dataloan$subtlexzipf)
```

```
## Warning: NAs introduced by coercion
```

```
dataloan$AoA = as.numeric(dataloan$AoA)
```

```
## Warning: NAs introduced by coercion
```

```
dataloan$conc = as.numeric(dataloan$conc)
```

```
## Warning: NAs introduced by coercion
```

```
dataloan$old.english.length = as.numeric(dataloan$old.english.length)
aoaSD = sd(dataloan$AoA,na.rm = T)
aoaMean = mean(dataloan$AoA/aoaSD,na.rm=T)
dataloan$cat = factor(dataloan$cat)
dataloan$effect = factor(dataloan$effect)
```

Select only complete cases.

```
dataloan2 = dataloan[complete.cases(dataloan[,
               c("phonlength","AoA",
               "subtlexzipf", "cat",
               'conc','bor15')]),]
```

Scale and center:

```
dataloan2$AoAscale <- scale(dataloan2$AoA)
dataloan2$subtlexzipfscale <- scale(dataloan2$subtlexzipf)
phonlength.center = median(dataloan2$phonlength)
dataloan2$phonlengthscale <-
  dataloan2$phonlength - phonlength.center
phonlength.scale = sd(dataloan2$phonlengthscale)
dataloan2$phonlengthscale = dataloan2$phonlengthscale/phonlength.scale
attr(dataloan2$phonlengthscale,"scaled:scale") = phonlength.scale
attr(dataloan2$phonlengthscale,"scaled:center") = phonlength.center
dataloan2$concscale <- scale(dataloan2$conc)
dataloan2$cat = relevel(dataloan2$cat,"Noun")
dataloan2$AoA_objscaled = scale(dataloan2$AoA_obj)
```

GAM for Old English

With old.english length as length. The idea of this analysis is to investigate whether length prior to borrowing affected whether the word was borrowed, rather than length of the borrowing. For that we need to go back to how the language was before the borrowings

Select only complete cases with old english length also:

```
dataloan3 = dataloan2[complete.cases(dataloan2[,
                                                c("phonlength","AoA",
                                                  "subtlexzipf", "cat",
                                                  'conc','bor15','old.english.length')]),]
```

Scale length:

```
old.english.length.center = median(dataloan3$old.english.length)
dataloan3$old.english.lengthscale <-
  dataloan3$old.english.length - old.english.length.center
old.english.length.scale = sd(dataloan3$old.english.lengthscale)
dataloan3$old.english.lengthscale = dataloan3$old.english.lengthscale/old.english.length.scale
attr(dataloan3$old.english.lengthscale,"scaled:scale") =
  old.english.length.scale
attr(dataloan3$old.english.lengthscale,"scaled:center") = old.english.length.center
```

Run GAM:

```
mOoe = bam(bor15.cat ~
           s(old.english.lengthscale) +
           s(AoAscale) +
           s(subtlexzipfscale) +
           s(concscale) +
           s(cat,bs='re')+
           s(cat,old.english.lengthscale,bs='re')+
           s(cat,AoAscale,bs='re')+
           s(cat,subtlexzipfscale,bs='re')+
           s(cat,concscale,bs='re'),
         data = dataloan3,
         family='binomial')
summary(mOoe)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(old.english.lengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, old.english.lengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
```

```
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.673      0.449  -3.726 0.000195 ***
## ---
## 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(old.english.lengthscale)     1.000e+00  1.000  0.829 0.362568
## s(AoAscale)                    1.000e+00  1.000 37.763 7.99e-10 ***
## s(subtlexzipfscale)            3.641e+00  4.603 29.897 1.35e-05 ***
## s(concscale)                   3.001e+00  3.741 17.371 0.001776 **
## s(cat)                         5.253e+00 10.000 33.182 0.000708 ***
## s(cat,old.english.lengthscale) 2.366e+00 10.000 42.011 0.001721 **
## s(cat,AoAscale)                6.108e-06 10.000  0.000 0.726086
## s(cat,subtlexzipfscale)        1.536e-05 10.000  0.000 0.464215
## s(cat,concscale)               5.460e-06 10.000  0.000 0.831722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.144   Deviance explained = 12.8%
## fREML = 1615.1  Scale est. = 1         n = 1139
```

There is a significant interaction between part of speech (cat) and old english length, but no significant effect of old english length on its own.

Test effect of old english length without interactions:

```
m1oe = bam(bor15.cat ~
             s(old.english.lengthscale) +
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re'),
           data = dataloan3,
           family='binomial')
summary(m1oe)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(old.english.lengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6669     0.4378  -3.808  0.00014 ***
## ---
## 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(old.english.lengthscale) 1.000  1.000  11.14 0.000843 ***
## s(AoAscale)                1.000  1.000  38.49 5.52e-10 ***
## s(subtlexzipfscale)        3.649  4.615  29.56 1.62e-05 ***
## s(concscale)               2.710  3.384  15.14 0.003178 **
## s(cat)                     5.291 10.000  35.46 4.65e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.136   Deviance explained =   12%
## fREML = 1615.2  Scale est. = 1         n = 1139
```

Now Old English length is significant and linear, as for Modern English phonlength. But we are still including words that may have been borrowed BEFORE Old English, so select only those borrowed after Old English:

Words borrowed after Old English

Classify as borrowed only those words borrowed after 900CE (so have changed since old english)

```
dataloan3$age_oldest_num <- as.numeric(dataloan3$age_oldest_num)
dataloan3$bor15oe <- ifelse(dataloan3$bor15==1 &
                              dataloan3$age_oldest_num <=1100 ,
                            1, ifelse(dataloan3$bor15==0,0,NA))
dataloan3$bor15oe.cat <- factor(dataloan3$bor15oe)
```

Run GAM:

```
m2oe = bam(bor15oe.cat ~
             s(old.english.lengthscale) +
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re')+
             s(cat,old.english.lengthscale,bs='re')+
             s(cat,AoAscale,bs='re')+
             s(cat,subtlexzipfscale,bs='re')+
             s(cat,concscale,bs='re'),
           data = dataloan3,
           family='binomial')
summary(m2oe)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15oe.cat ~ s(old.english.lengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, old.english.lengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7231     0.4485  -3.842 0.000122 ***
## ---
## 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(old.english.lengthscale)     1.000e+00  1.000  0.851 0.356390
## s(AoAscale)                    1.000e+00  1.000 33.304 7.88e-09 ***
## s(subtlexzipfscale)            3.497e+00  4.432 25.452 7.92e-05 ***
## s(concscale)                   2.569e+00  3.210 18.229 0.000617 ***
## s(cat)                         5.271e+00 10.000 32.640 0.000739 ***
## s(cat,old.english.lengthscale) 2.316e+00 10.000 41.004 0.001789 **
## s(cat,AoAscale)                2.603e-06 10.000  0.000 0.617202
## s(cat,subtlexzipfscale)        1.266e-05 10.000  0.000 0.456492
## s(cat,concscale)               3.102e-06 10.000  0.000 0.776301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.141   Deviance explained = 12.7%
## fREML =   1583  Scale est. = 1         n = 1117
```

Again, interaction between cat and length, so remove interactions to investigate main effects:

```
m3oe = bam(bor15oe.cat ~
             s(old.english.lengthscale) +
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re'),
           data = dataloan3,
           family='binomial')
summary(m3oe)
```

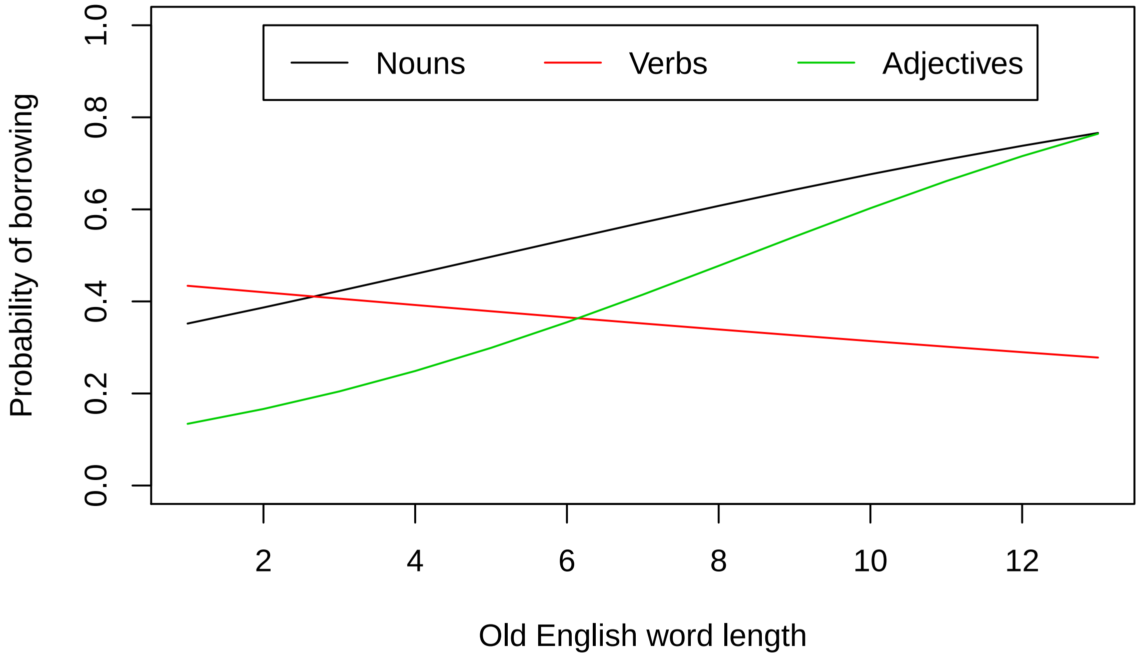
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15oe.cat ~ s(old.english.lengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7174     0.4372  -3.928 8.55e-05 ***
## ---
## 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(old.english.lengthscale) 1.000  1.000  11.27 0.000787 ***
## s(AoAscale)                1.000  1.000  33.90 5.82e-09 ***
## s(subtlexzipfscale)        3.502  4.440  25.09 9.56e-05 ***
## s(concscale)               2.227  2.789  14.28 0.001467 **
## s(cat)                     5.300 10.000  35.42 4.59e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.133   Deviance explained = 11.8%
## fREML = 1583.5  Scale est. = 1         n = 1117
```

As before, a significant linear effect of length.

This is evidence that length of the Old English form influenced the likelihood of that word subsequently being borrowed, and is not just a consequence of borrowed words coming from languages with longer word length than English.

Effects for each part of speech

What is the relationship between old english word length and probability of borrowing for each part of speech? Below are plots which are calculated by using the model `m2oe` to predict borrowing probaility for a range of combinations of all variables, but keeping `cat` (part of speech) fixed to e.g. Nouns. Then the average borrowing probaility for each value of old english word length is plotted. The code is hidden, but can be seen in the Rmd file. These are for diagnosis only, formal tests are below.



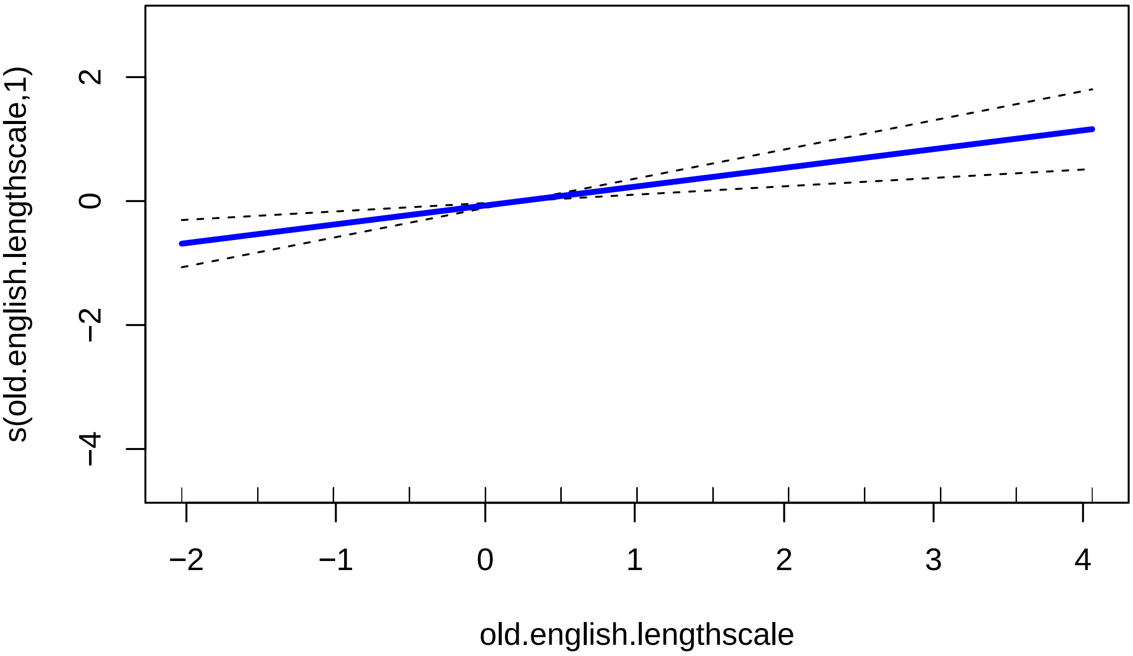
Analyse nouns only: Old English length is significant and monotonically increasing for nouns.

```
m2oen = bam(bor15oe.cat ~
              s(old.english.lengthscale) +
              s(AoAscale) +
              s(subtlexzipfscale) +
              s(concscale),
            data = dataloan3,subset = cat == "Noun",
            family='binomial')
summary(m2oen)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15oe.cat ~ s(old.english.lengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale)
##
```

```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.35603    0.08483  -4.197 2.71e-05 ***
## ---
## 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(old.english.lengthscale) 1.000  1.000 13.015 0.000309 ***
## s(AoAscale)                1.535  1.910 21.068 7.13e-05 ***
## s(subtlexzipfscale)        2.943  3.770 16.431 0.002198 **
## s(concscale)               1.792  2.249  9.731 0.010753 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.108   Deviance explained = 9.02%
## fREML = 924.44  Scale est. = 1         n = 646
```

```
plotGAMSignificantSlopes(m2oen,"old.english.lengthscale","OE Length")
```



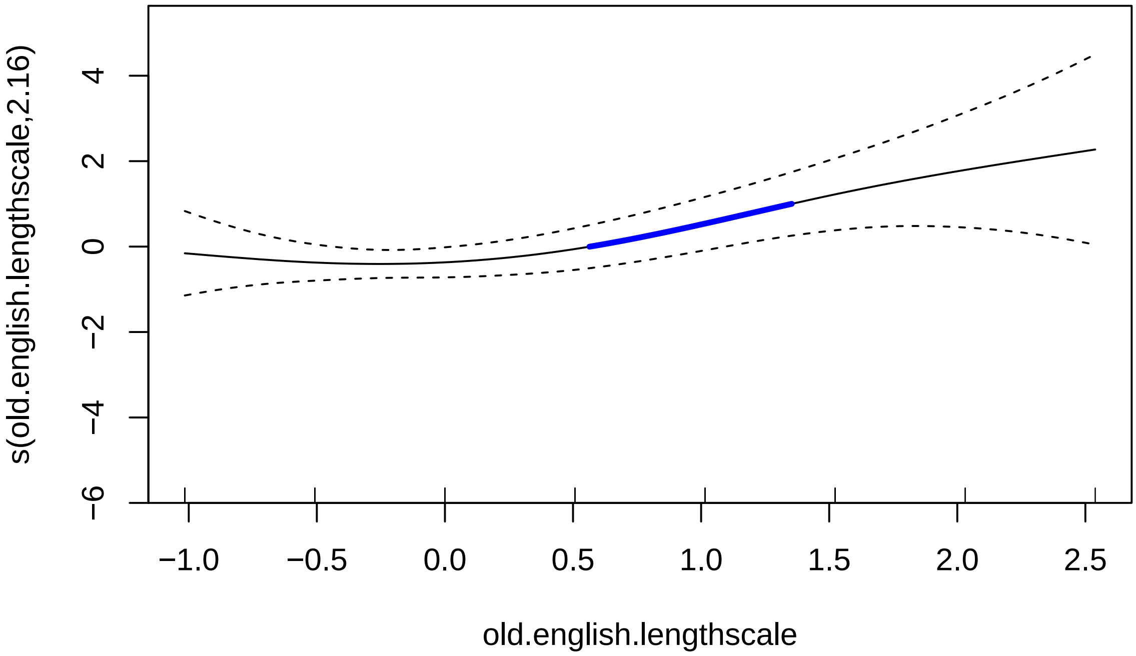
Analyse adjectives: Old English length is significant and increasing for adjectives.

```
m2oea = bam(bor15oe.cat ~
              s(old.english.lengthscale, k = 7) +
              s(AoAscale) +
              s(subtlexzipfscale) +
              s(concscale),
            data = dataloan3,subset = cat == "Adjective",
            family='binomial')
summary(m2oea)
```

```
##
## Family: binomial
## Link function: logit
```

```
##
## Formula:
## bor15oe.cat ~ s(old.english.lengthscale, k = 7) + s(AoAscale) +
##     s(subtlexzipfscale) + s(concscale)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8647     0.2249  -3.845 0.000121 ***
## ---
## 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(old.english.lengthscale) 2.164  2.725  8.311 0.02531 *
## s(AoAscale)                1.000  1.000  8.670 0.00323 **
## s(subtlexzipfscale)        1.293  1.529  4.059 0.05816 .
## s(concscale)               1.111  1.213  0.181 0.80144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.168   Deviance explained = 16.9%
## fREML = 168.26  Scale est. = 1         n = 117
```

```
plotGAMSignificantSlopes(m2oea,"old.english.lengthscale","OE Length")
```



Analyse verbs: Length is not significantly related to the probability of borrowing for verbs.

```
m2oev = bam(bor15oe.cat ~
              s(old.english.lengthscale, k = 7) +
              s(AoAscale) +
              s(subtlexzipfscale) +
              s(concscale),
            data = dataloan3,subset = cat == "Verb",
            family='binomial')
```

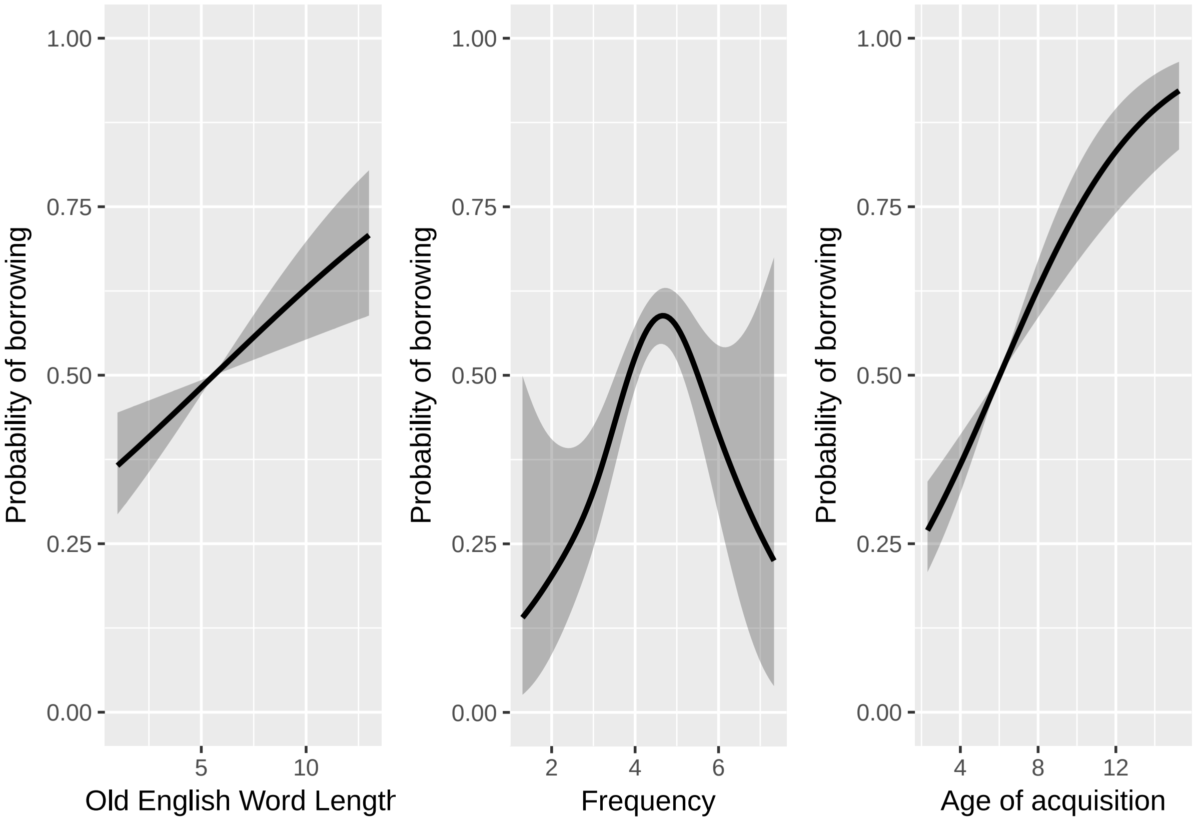
```r
summary(m2oev)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15oe.cat ~ s(old.english.lengthscale, k = 7) + s(AoAscale) +
##     s(subtlexzipfscale) + s(concscale)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.7711     0.1387  -5.557 2.74e-08 ***
## ---
## 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(old.english.lengthscale) 2.480  3.132  4.088 0.23811
## s(AoAscale)                1.000  1.000  4.502 0.03385 *
## s(subtlexzipfscale)        3.102  3.938  6.174 0.20907
## s(concscale)               1.000  1.000  8.458 0.00363 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.125   Deviance explained = 11.6%
## fREML = 394.53  Scale est. = 1         n = 274
```

Model plots

Plot the model estimates with Old English length, changing the dependent scale to probability and the independent variables to their original scales. This code is hidden, but you can view it in the Rmd file.

Plotting model estimates:


Probability of borrowing
1.00
0.75
0.50
0.25
0.00
5
10
Old English Word Lengtł
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
2
4
6
Frequency
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
4
8
12
Age of acquisition

Swadesh words

Next analyses are for likelihood of borrowing just for the Swadesh words:

Identify Swadesh words:

```
swd = read.csv("../data/SwadeshConcepts.txt",
               header = F, stringsAsFactors = F)$V1
dataloan3$Swadesh = dataloan3$word %in% swd
```

Run GAM:

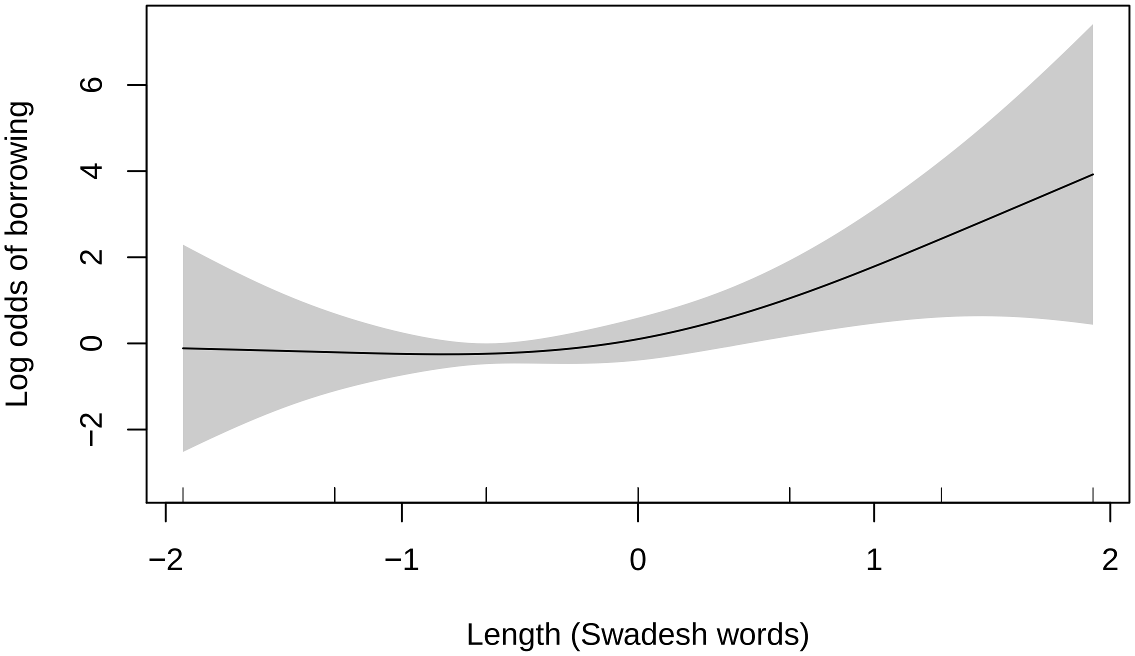
The phonetic length variable only has 10 unique values, leading to fewer unique covariate combinations than the default degrees of freedom. Following Winter & Wieling (2016), we set the number of knots to half of the unique values (k=5).

```
m0sw = bam(bor15.cat ~
             s(phonlengthscale,k=5) +
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re'),
           data = dataloan3,
           subset = Swadesh == TRUE,
           family='binomial')
summary(m0sw)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale, k = 5) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9582     0.2489  -7.868  3.6e-15 ***
## ---
## 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(phonlengthscale)  1.998e+00  2.509  7.749  0.0415 *
## s(AoAscale)         1.312e+00  1.564  0.195  0.7443
## s(subtlexzipfscale) 1.000e+00  1.000  1.248  0.2640
## s(concscale)        1.000e+00  1.000  0.092  0.7616
## s(cat)              2.339e-06  8.000  0.000  0.7699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0891   Deviance explained = 10.4%
## fREML =  251.5  Scale est. = 1         n = 175
```

The only significant effect is length.

```
px = plot.gam(m0sw,select=1, xlab="Length (Swadesh words)", ylab="Log odds of borrowing",shade = T)
```



Coexistence, insertion or replacement

This works out influences on type of borrowing: coexistence, insertion, or replacement:

```
dataloan2$effect2 = dataloan2$effect
dataloan2$effect2[!dataloan2$effect2 %in%
      c("Insertion","Replacement","Coexistence")] = NA
dataloan2$effect2 = factor(dataloan2$effect2)
dataloan2$effect2 = as.numeric(dataloan2$effect2)-1
```

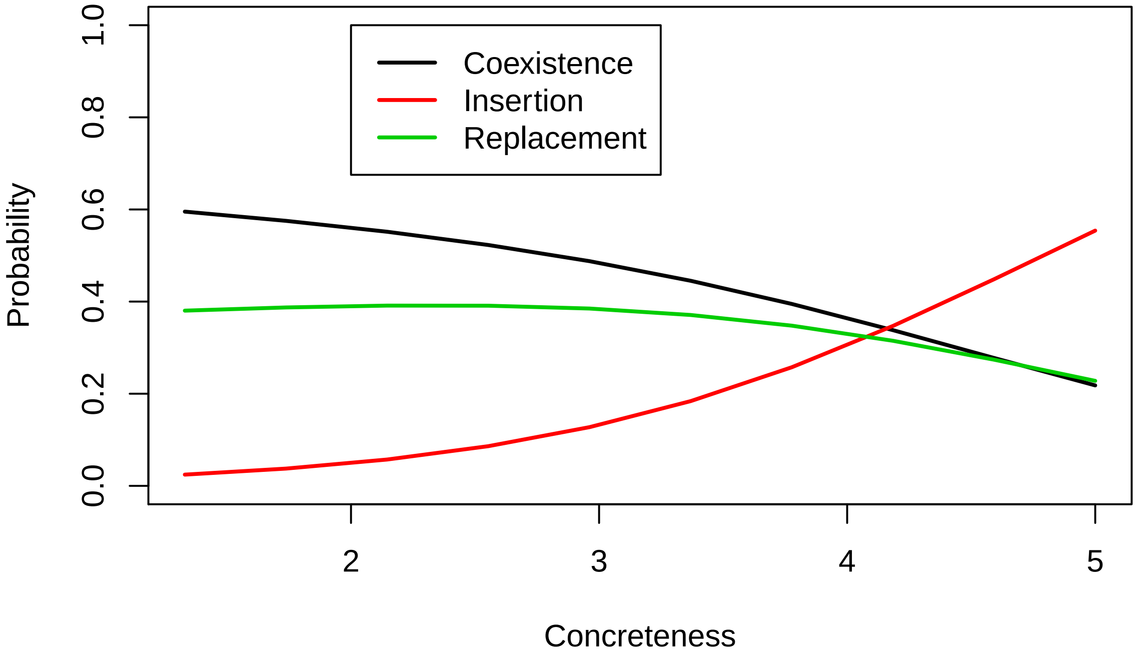
Multinomial GAM (predicting probability of each category given length, AoA, frequency, concreteness and PoS).

```
mCIR = gam(
        list(effect2 ~
            # formula for 2nd category
             s(phonlengthscale)+
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re'),
            # formula for 3rd category
             ~ s(phonlengthscale)+
             s(AoAscale) +
             s(subtlexzipfscale) +
             s(concscale) +
             s(cat,bs='re')),
           data = dataloan2[!is.na(dataloan2$effect2),],
           family=multinom(K=2))
summary(mCIR)
```

```
##
## Family: multinom
## Link function:
##
## Formula:
## effect2 ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re")
## ~s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) + s(concscale) +
##     s(cat, bs = "re")
##
## Parametric coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.166581   0.298300  -0.558    0.577
## (Intercept).1 -0.007741   0.164043  -0.047    0.962
##
## Approximate significance of smooth terms:
##                         edf Ref.df Chi.sq  p-value
## s(phonlengthscale)    1.0001  1.000  0.175   0.6761
## s(AoAscale)           1.0003  1.001  0.573   0.4490
## s(subtlexzipfscale)   1.9658  2.526  1.946   0.4665
## s(concscale)          1.0138  1.027 39.430 4.18e-10 ***
## s(cat)                1.1957  6.000  3.325   0.0714 .
## s.1(phonlengthscale)  1.5543  1.939  2.154   0.3832
## s.1(AoAscale)         1.0001  1.000  0.030   0.8614
## s.1(subtlexzipfscale) 1.0001  1.000  0.047   0.8284
## s.1(concscale)        1.0000  1.000  0.934   0.3338
## s.1(cat)              0.5735  6.000  0.957   0.2024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Deviance explained =   11%
## -REML =  332.4  Scale est. = 1         n = 477
```

Only concreteness is a significant predictor.

Visualise the relationship between concreteness and the probability of each category. Code is not shown, but available in the Rmd file.


Coexistence
Insertion
Replacement
Probability
Concreteness