

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 psycholinguistic 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)
  pln = 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)){
    pln = pln + scale_x_continuous(breaks = breaks)
  }
  if(!is.null(xlim)){
    pln = pln + coord_cartesian(ylim = c(0,1), xlim=xlim)
  } else{
    pln = pln + 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.

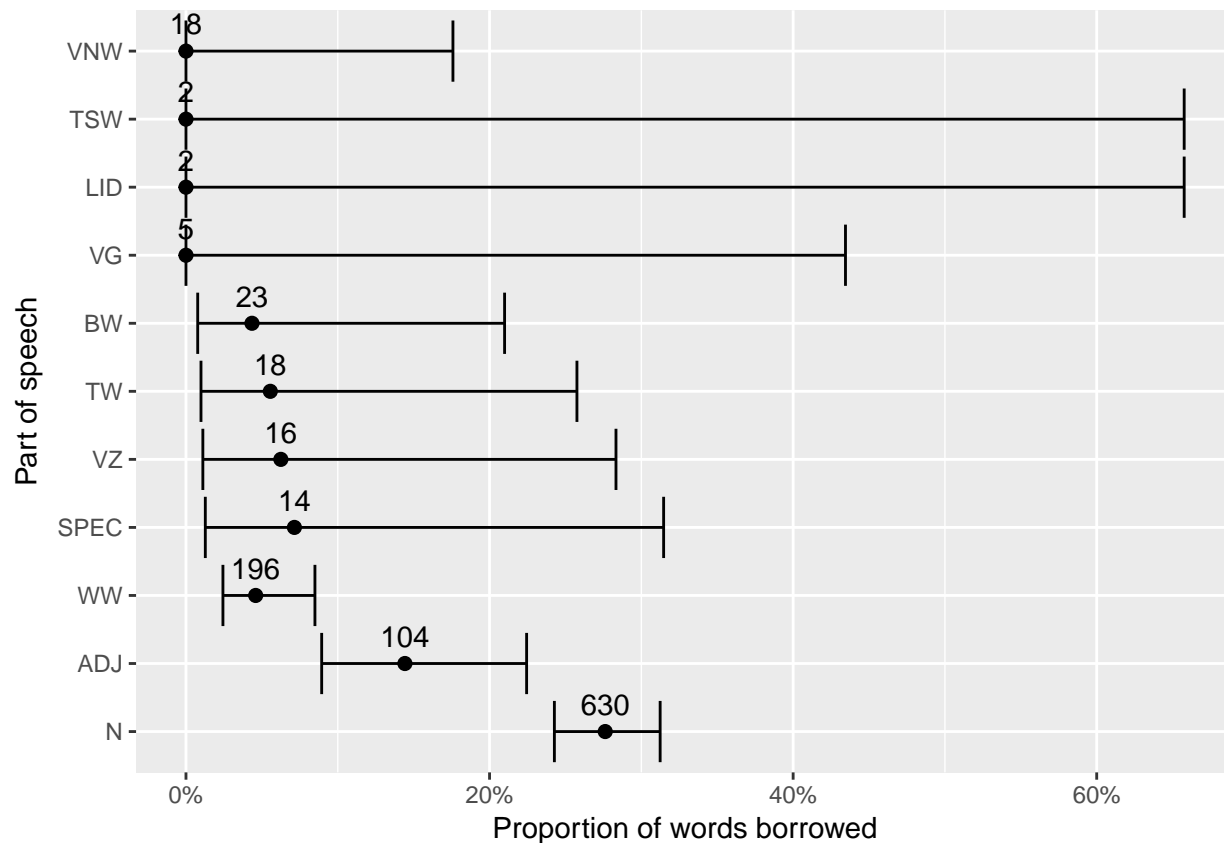
```

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(subtlelexzipfscale) +
s(concyscale) +
s(cat,bs='re')+
s(cat,phonlengthscale,bs='re')+
s(cat,AoAscale,bs='re')+
s(cat,subtlelexzipfscale,bs='re')+
s(cat,concyscale,bs='re'),
data = dutch,
family='binomial')

```

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

```

##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlelexzipfscale) +
##      s(concyscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##      bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlelexzipfscale,
##      bs = "re") + s(cat, concyscale, 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(subtlelexzipfscale)  3.667e+00  4.606 12.048 0.021505 *
## s(concyscale)         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,subtlelexzipfscale) 1.346e-05 10.000  0.000 0.668251
## s(cat,concyscale)     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(subtlelexzipfscale) +
  s(concscale) +
  s(cat,bs='re')+
  s(cat,phonlengthscale,bs='re')+
  s(cat,AoAscale,bs='re')+
  s(cat,subtlelexzipfscale,bs='re')+
  s(cat,concscale,bs='re') +
  te(AoAscale,subtlelexzipfscale),
  data = dutch,
  family='binomial')

lrtest(m0.dutch,m1.dutch)

## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale, k = 4) + s(AoAscale) + s(subtlelexzipfscale) +
##   s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##   bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlelexzipfscale,
##   bs = "re") + s(cat, concscale, bs = "re")
## Model 2: bor15.cat ~ s(phonlengthscale, k = 3) + s(AoAscale) + s(subtlelexzipfscale) +
##   s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##   bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlelexzipfscale,
##   bs = "re") + s(cat, concscale, bs = "re") + te(AoAscale,
##   subtlelexzipfscale)
##   #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(subtlelexzipfscale) +
  s(concscale) +
  s(cat,bs='re')+
  s(cat,phonlengthscale,bs='re')+
  s(cat,AoAscale,bs='re')+
  s(cat,subtlelexzipfscale,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
```

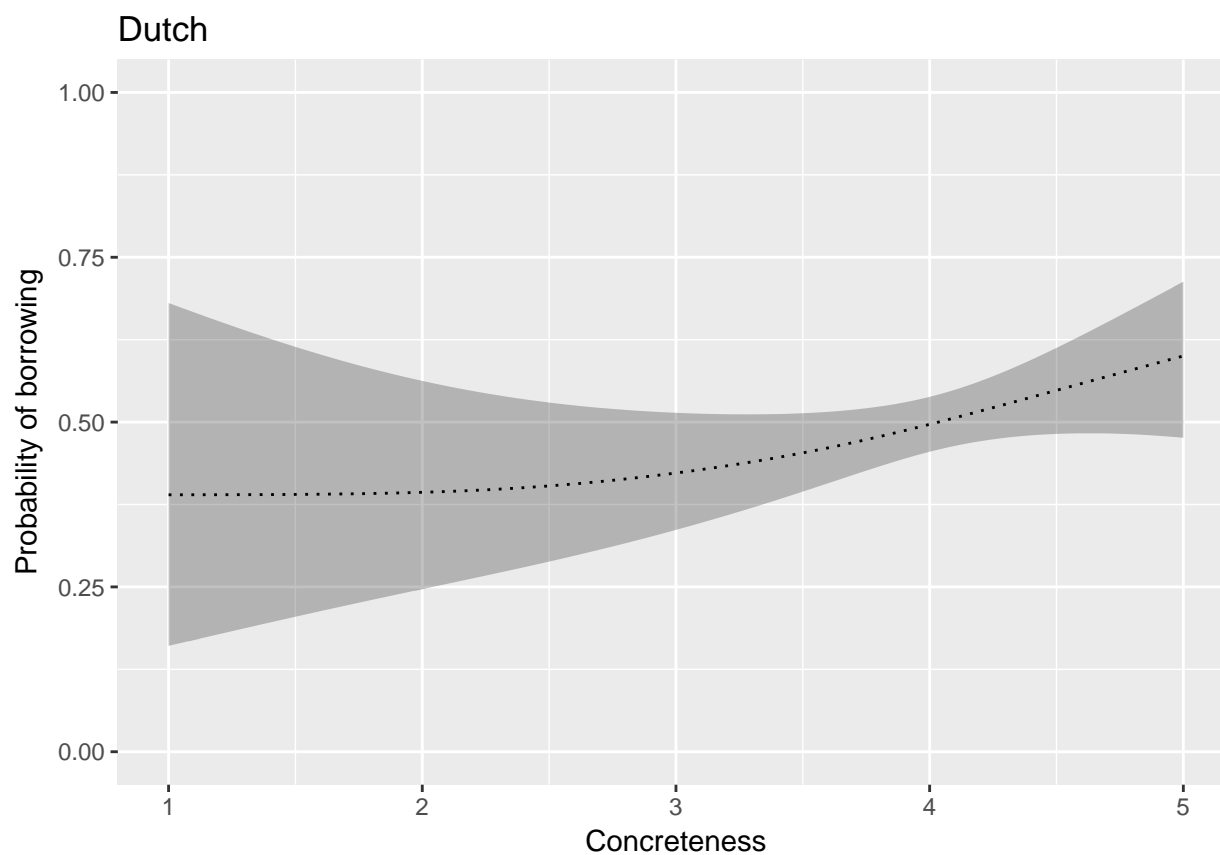
No significant improvement.

So no interactions are necessary.

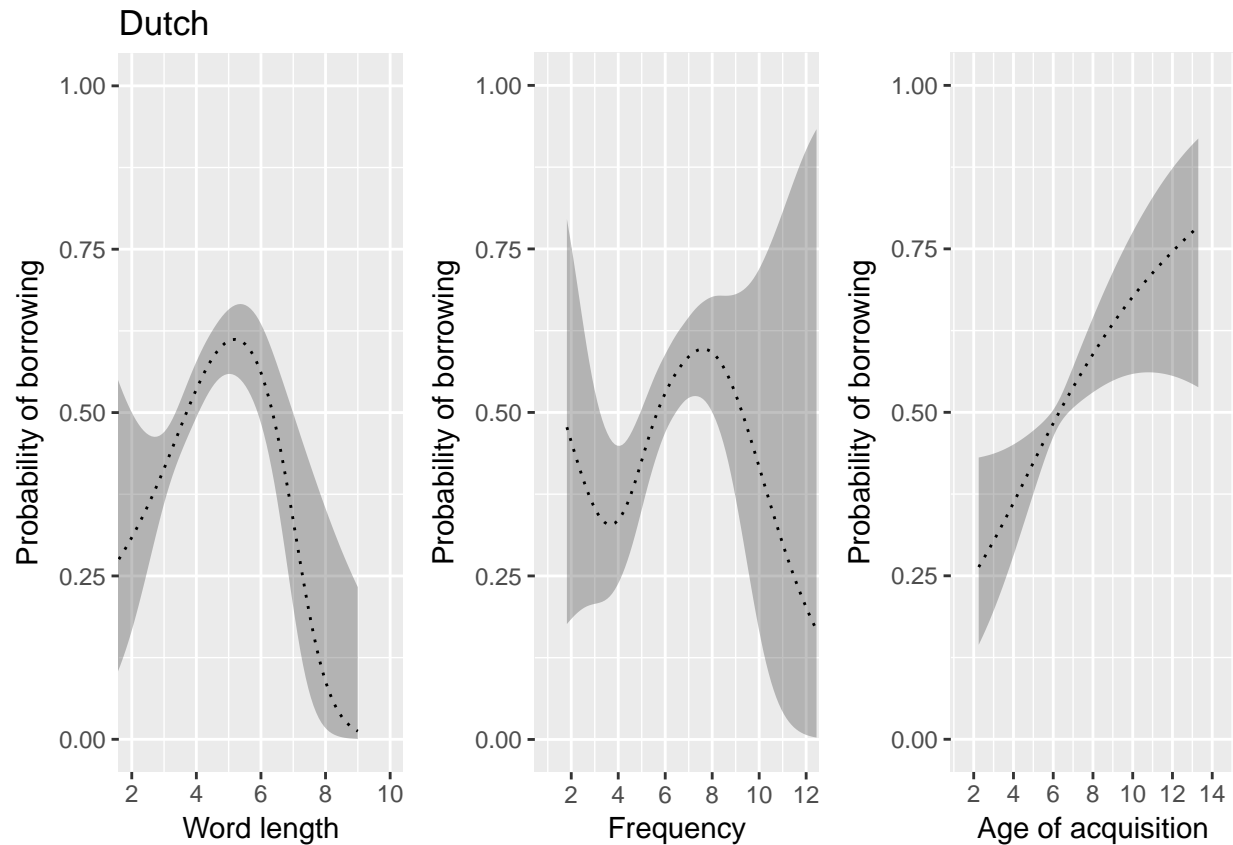
Model estimates

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$AoAscale, "Frequency",  
                   xlim = c(1,12), breaks=c(2,4,6,8,10,12)) + ggtitle("")  
pconc = rescaleGam(px, 4, dutch$concyscale, "Concreteness") + ggtitle("Dutch")  
  
pconc
```



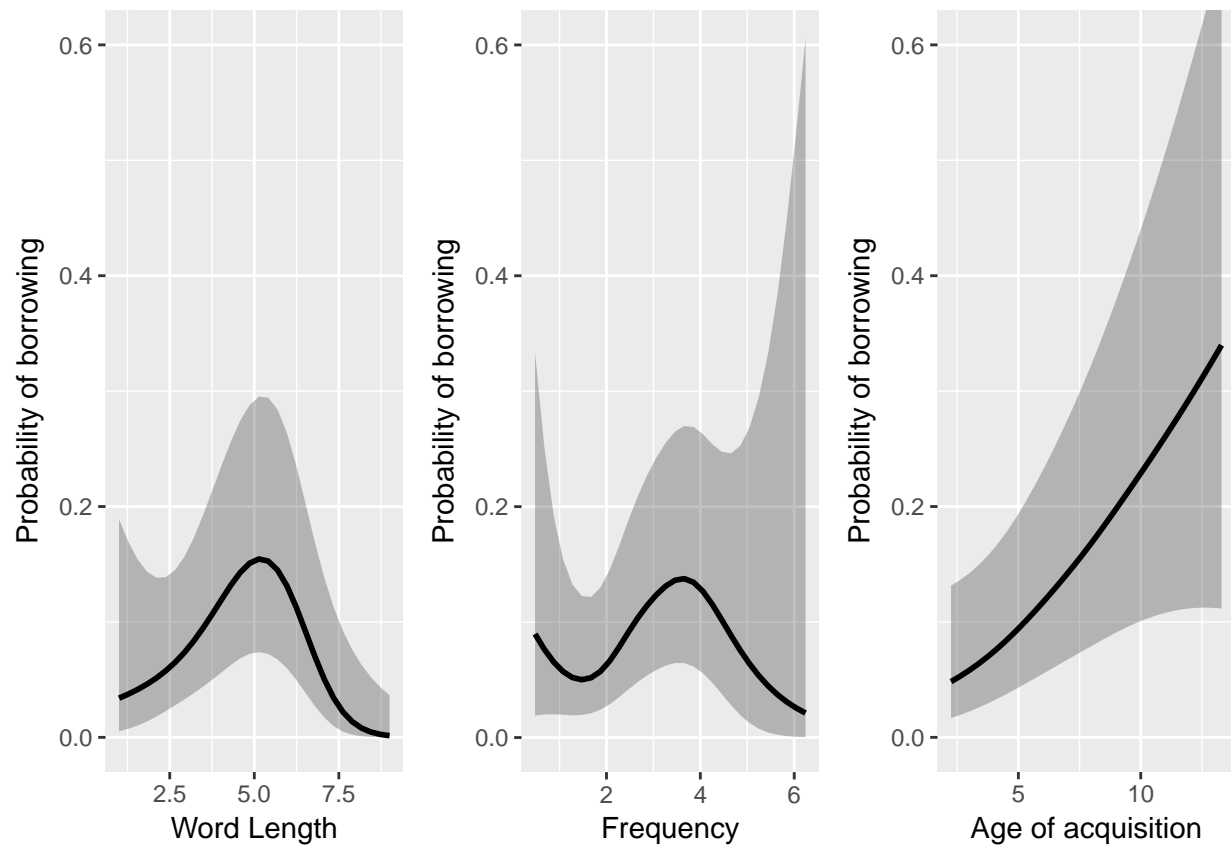
```
grid.arrange(plen,pfreq,paoa, nrow=1)
```



```
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

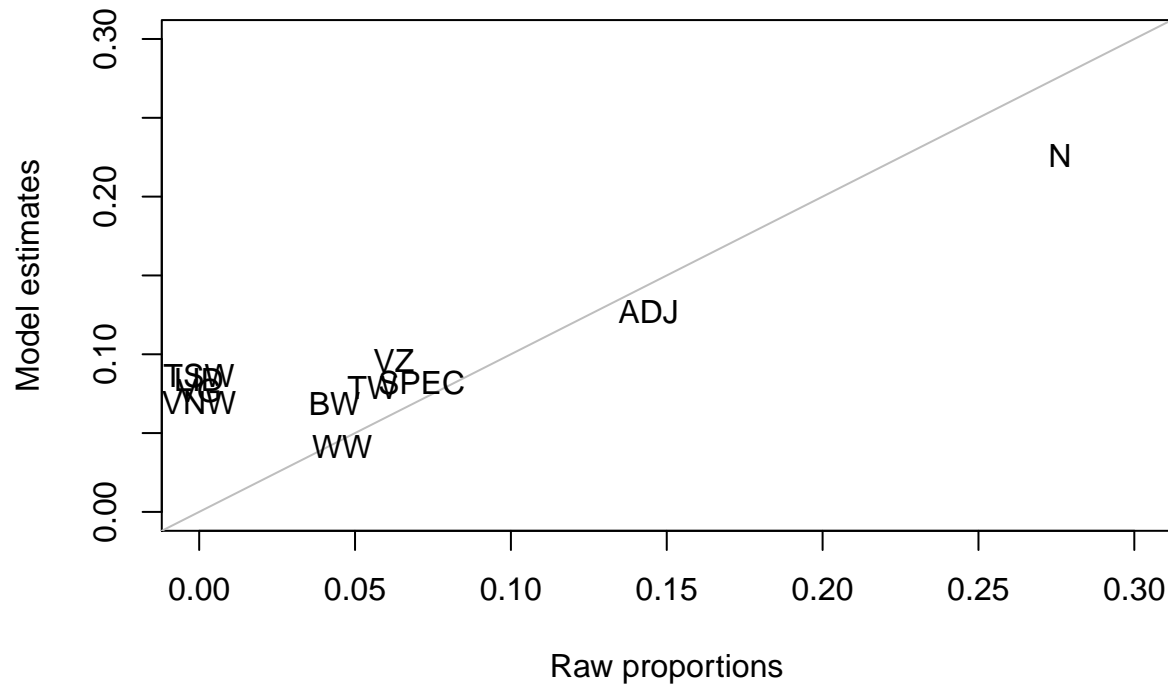
```
mc = m0.dutch$coefficients
mc[grepl("s\\(cat\\)", names(mc))]

##      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')
```

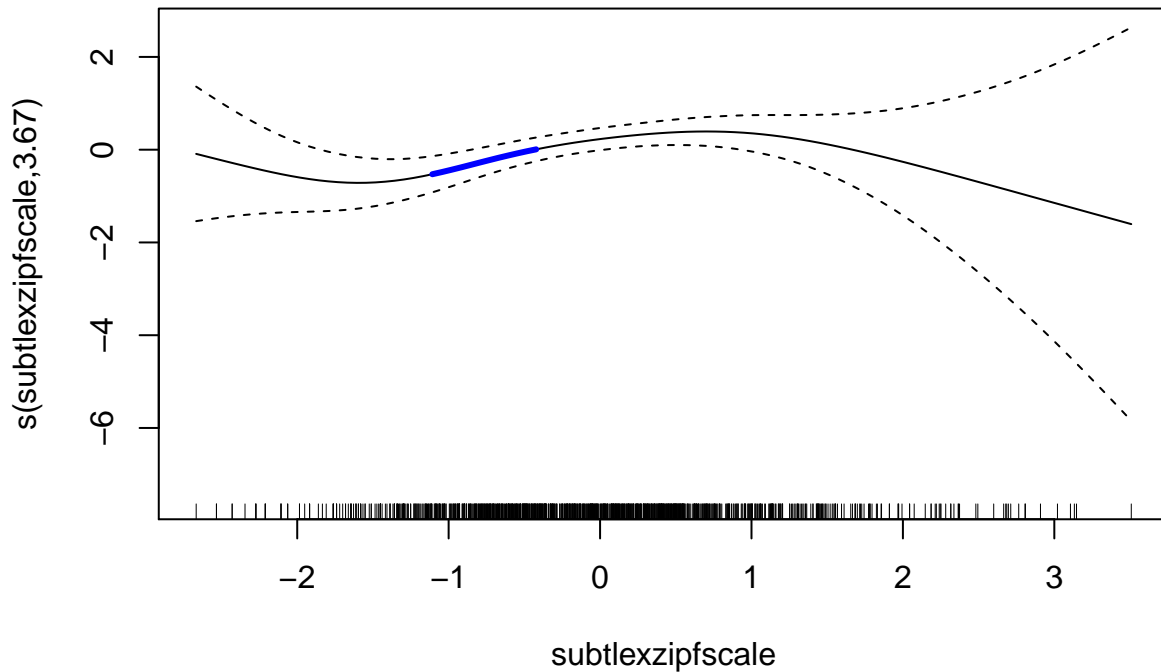
```
text(raw, model.est, names(raw))
```



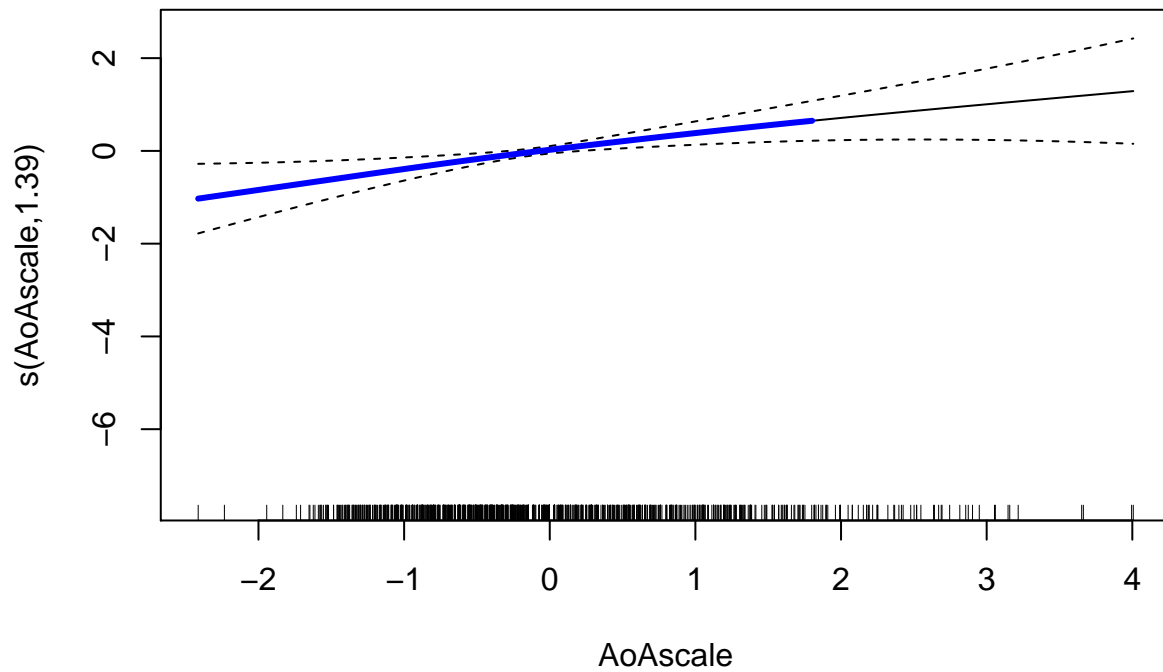
Derivatives test

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

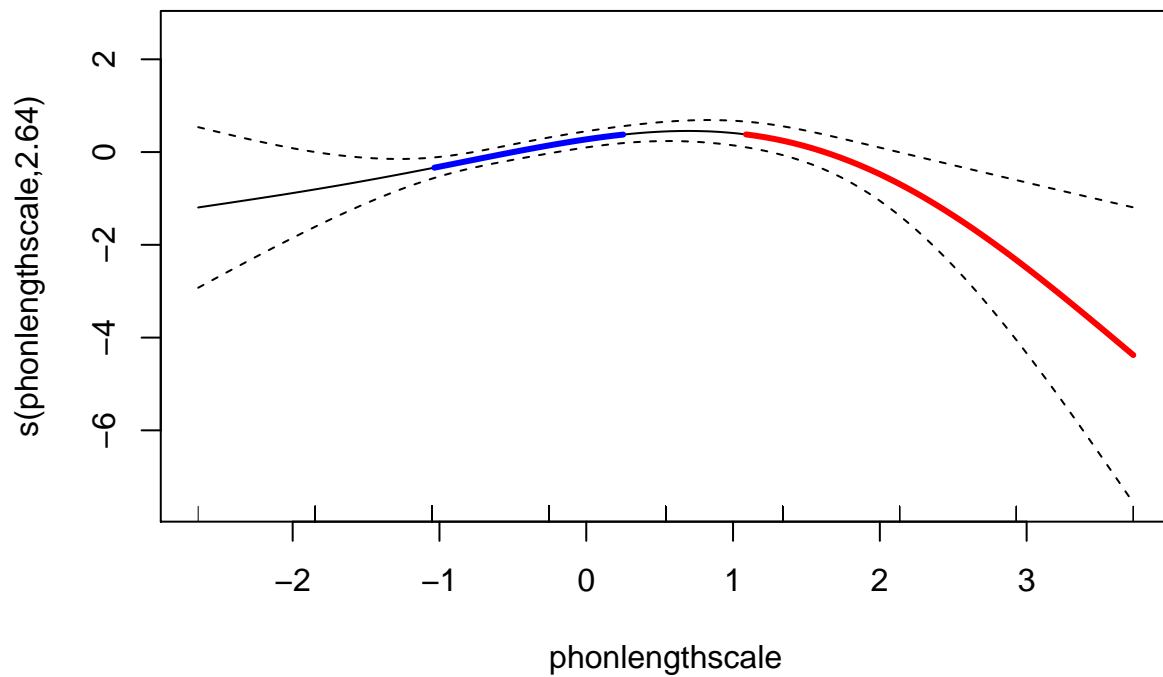
```
pSigFreq = plotGAMSignificantSlopes(m0.dutch, "subtlexzipfscale", "Frequency")
```



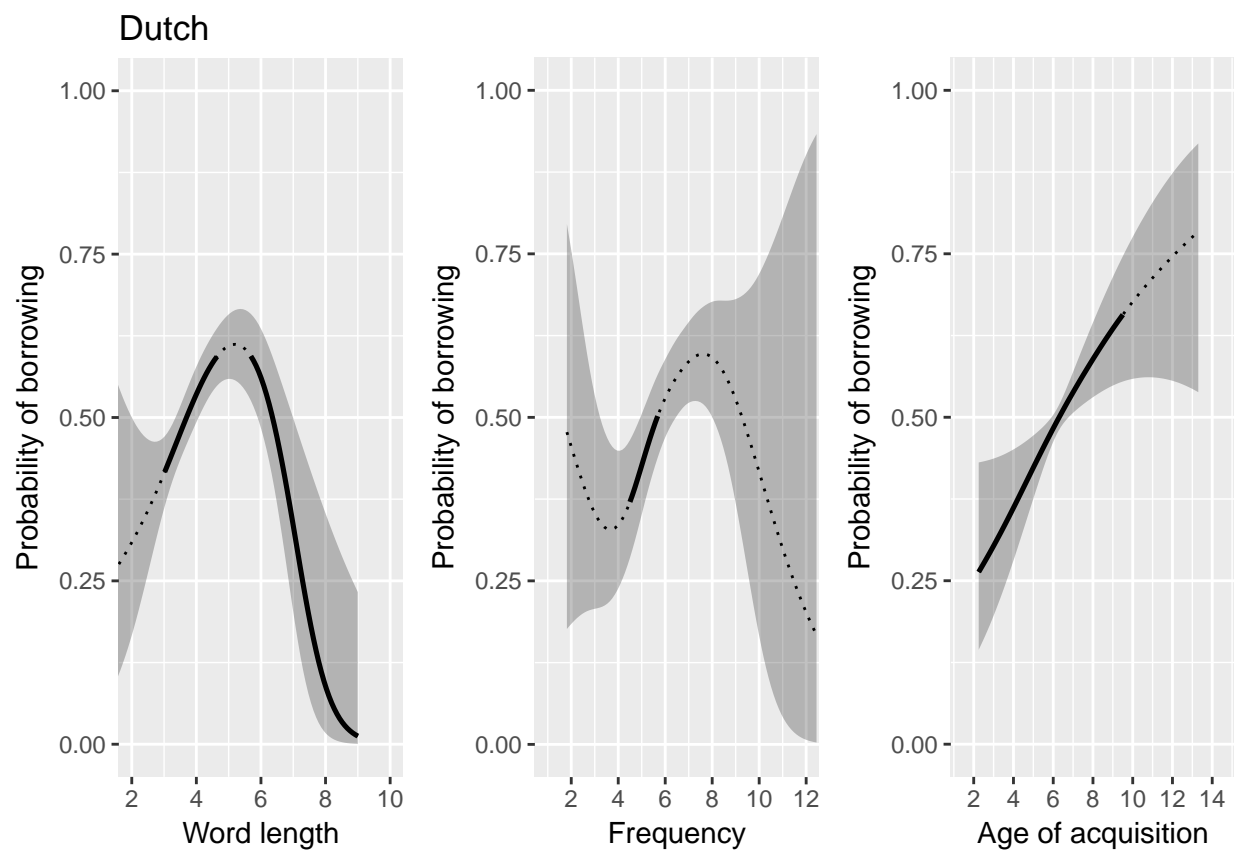
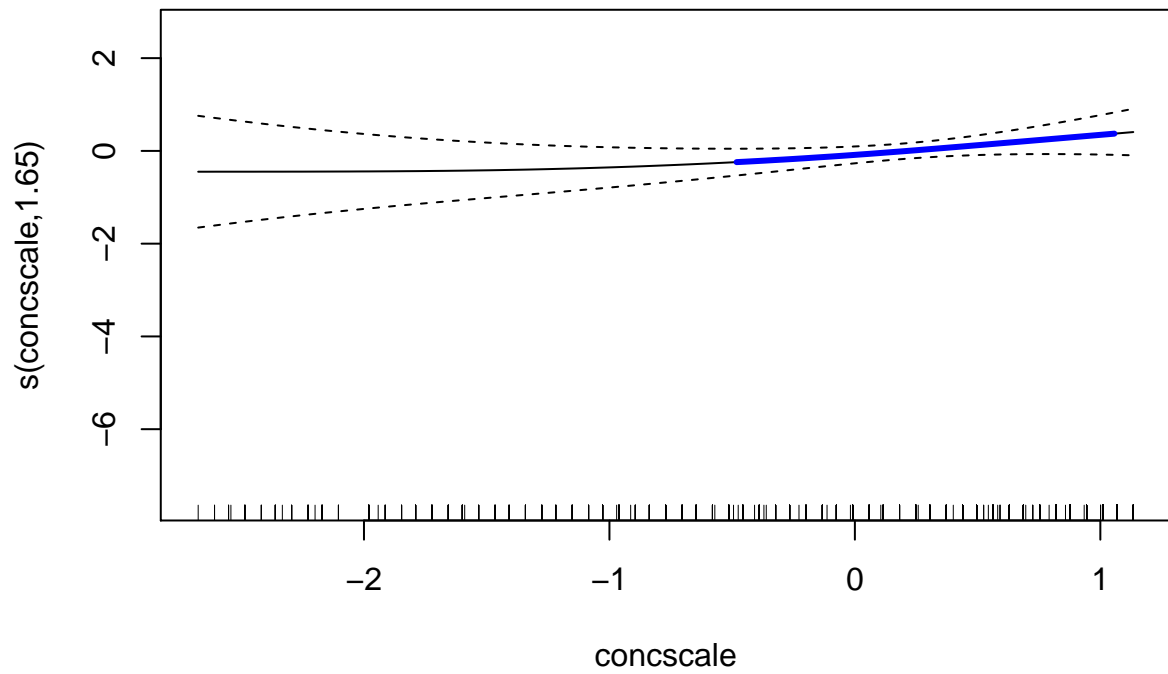
```
pSigAoA = plotGAMSignificantSlopes(m0.dutch, "AoAscale", "AoA")
```



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



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



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