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
- age_oldest, age_youngest: Dates from WOLD indicating estiamte of data of entry into English
- 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.
- bor15: Conversion of the WOLD borrowing variable into a numeric (0 = not borrowed, 1 = borrowed)

Load libraries

```
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=""){
  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")+
    coord_cartesian(ylim = c(0,1))
  return(plen)
}
# Code for assessing significance of GAM slopes
source("GAM_derivaties.R")
```

Load data

```
dataloan <- read.csv("../data/loanword9.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)
```

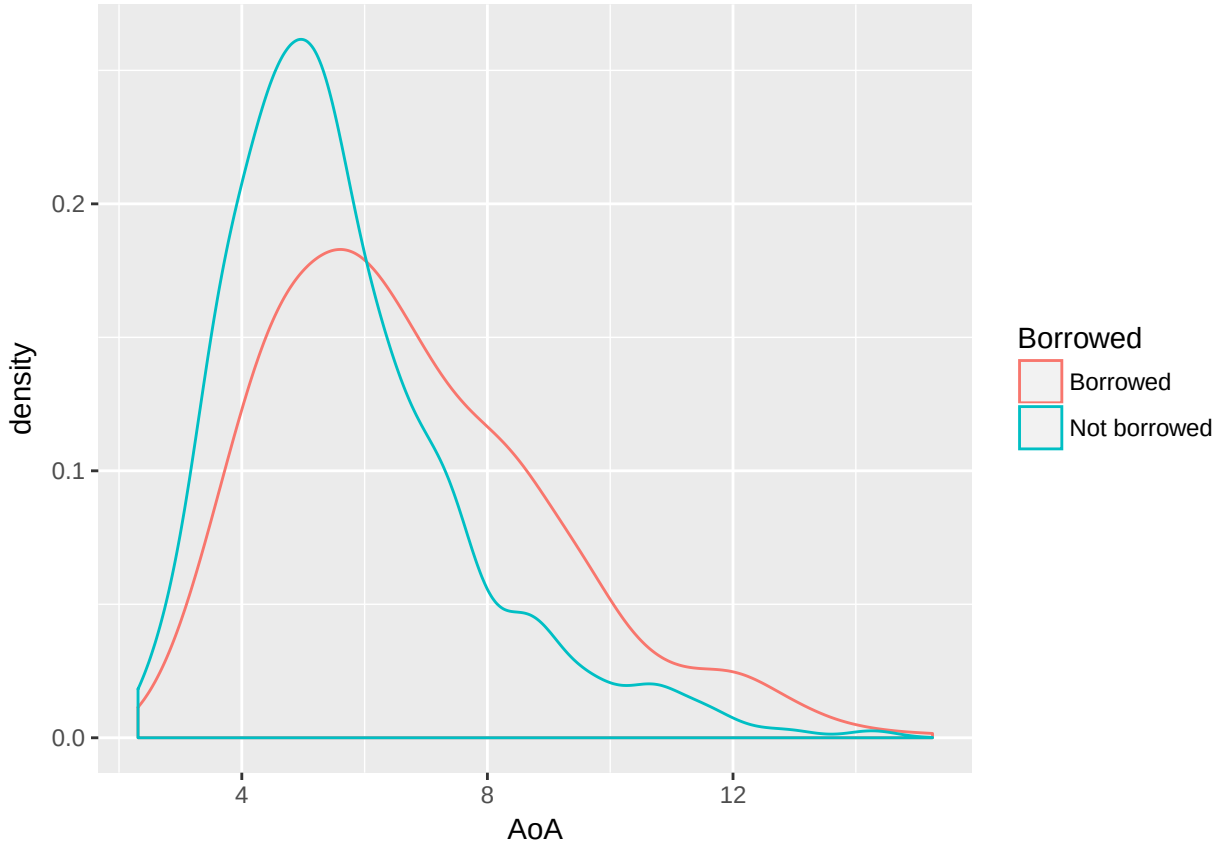
Identify Swadesh words:

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

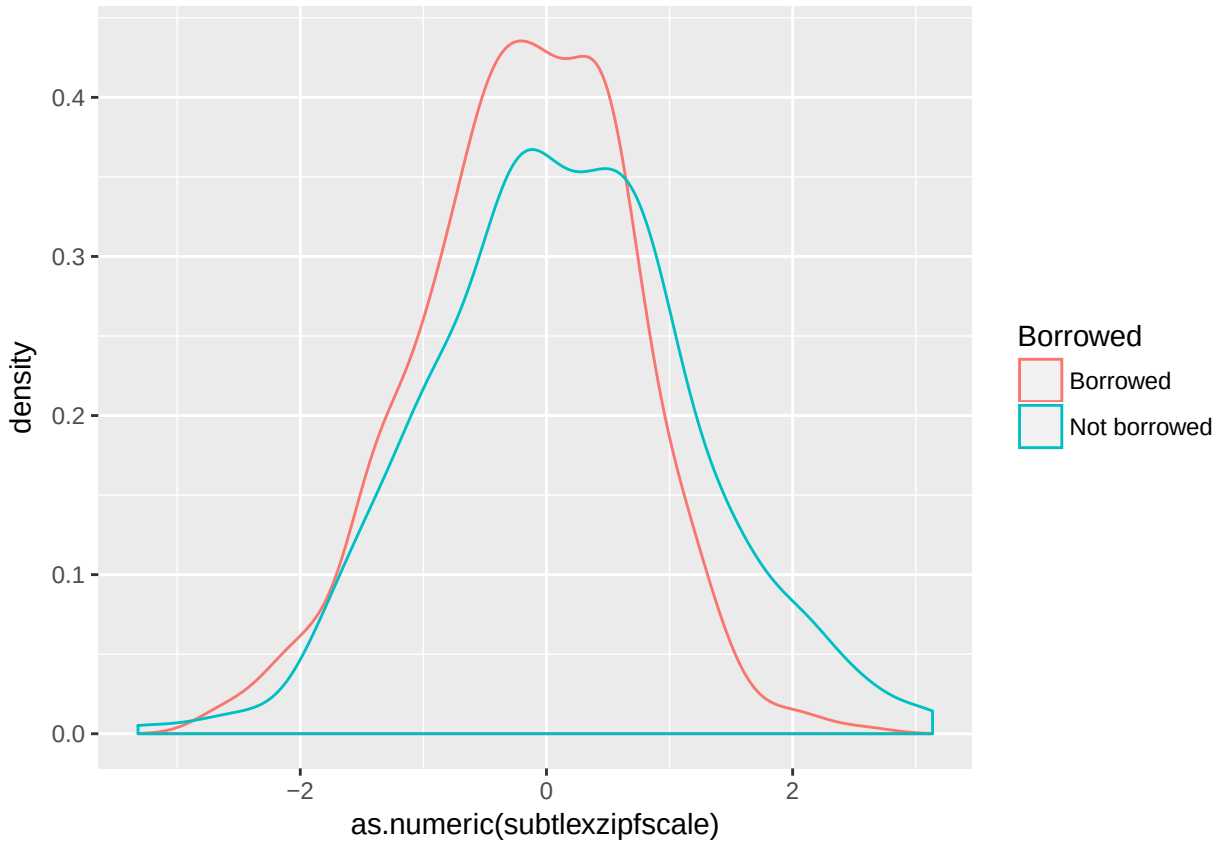
Plots

Raw data

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

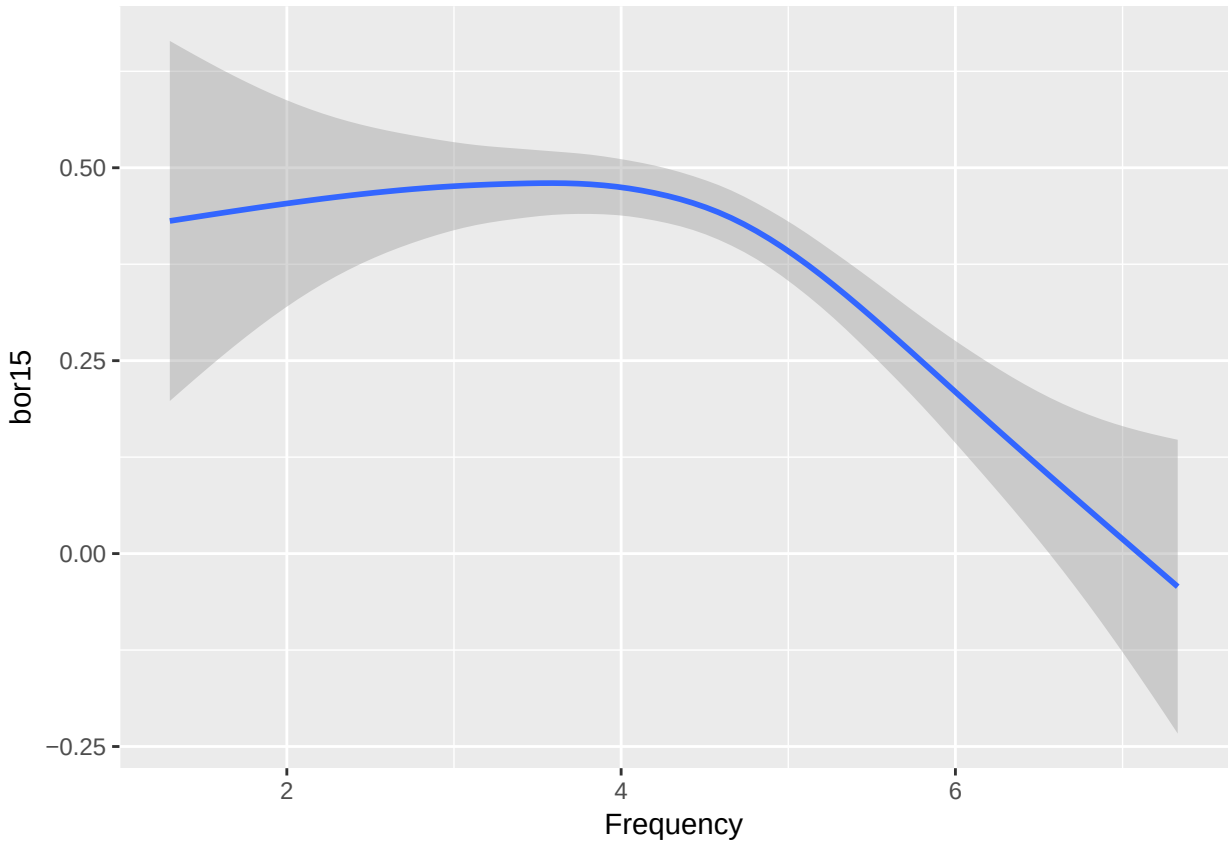



```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipfscale), colour=Borrowed)) +
  geom_density()
```



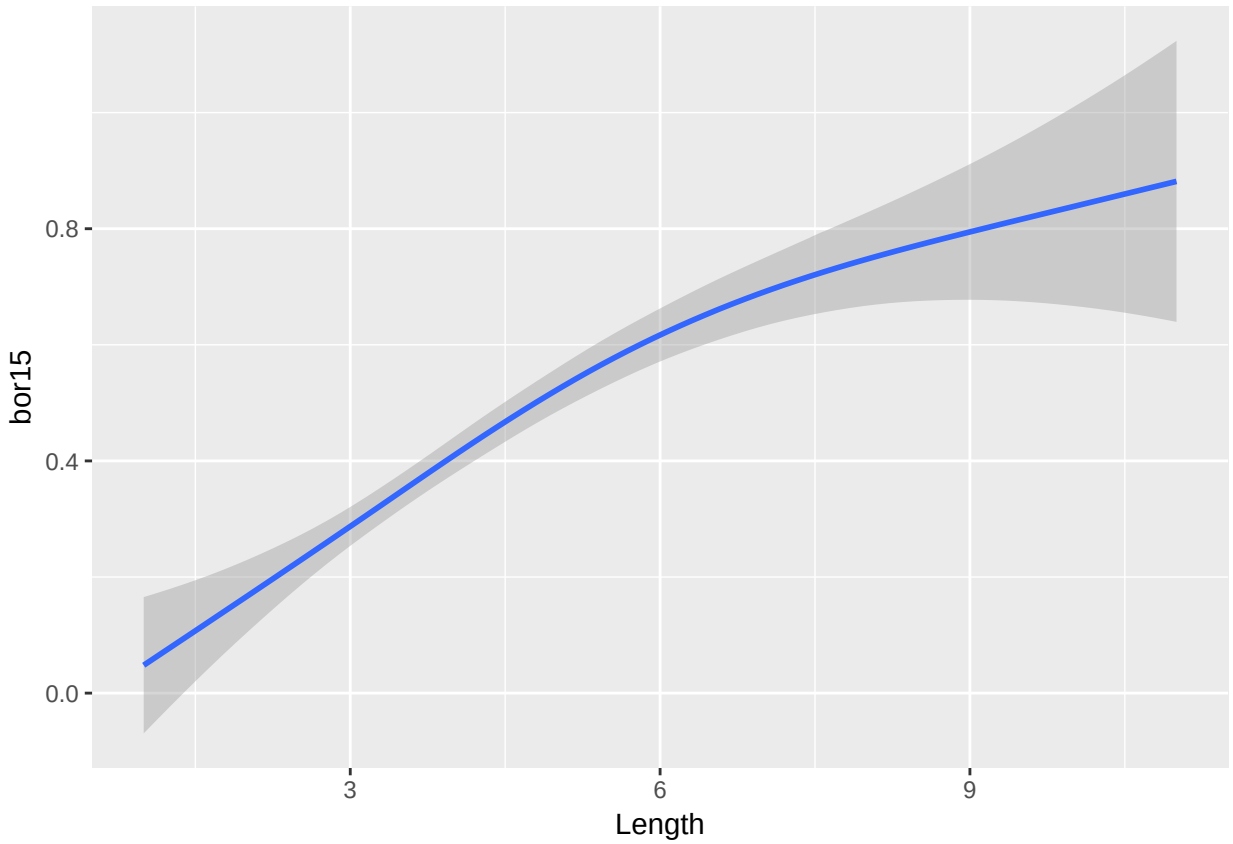

```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipf), y=bor15)) +
  stat_smooth()+
  xlab("Frequency")
```

```
## `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")'
```

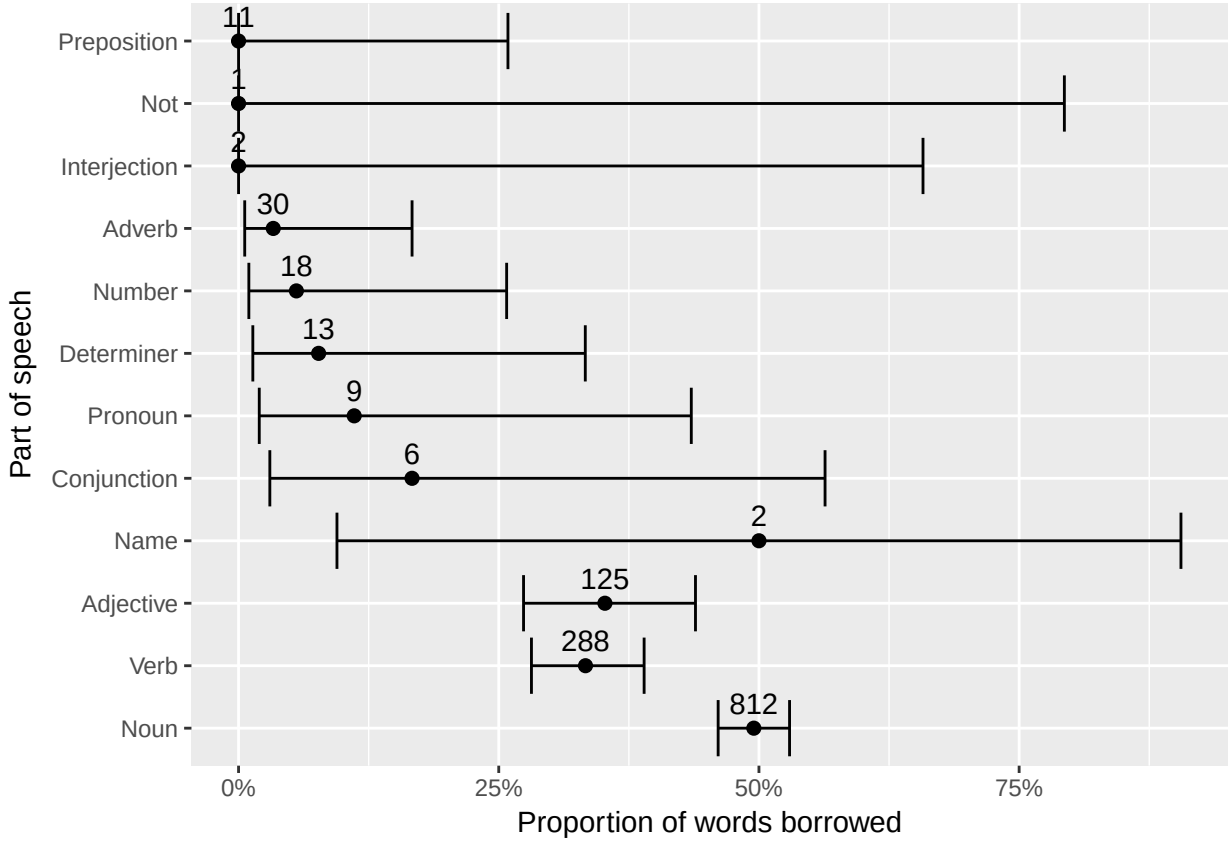



```
dataloan2$subtlexzipf.cat = cut(
  dataloan2$subtlexzipf,
  breaks = quantile(dataloan2$subtlexzipf,
                    prob=seq(0,1,length.out=4)),
  include.lowest = T)
```

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)
```

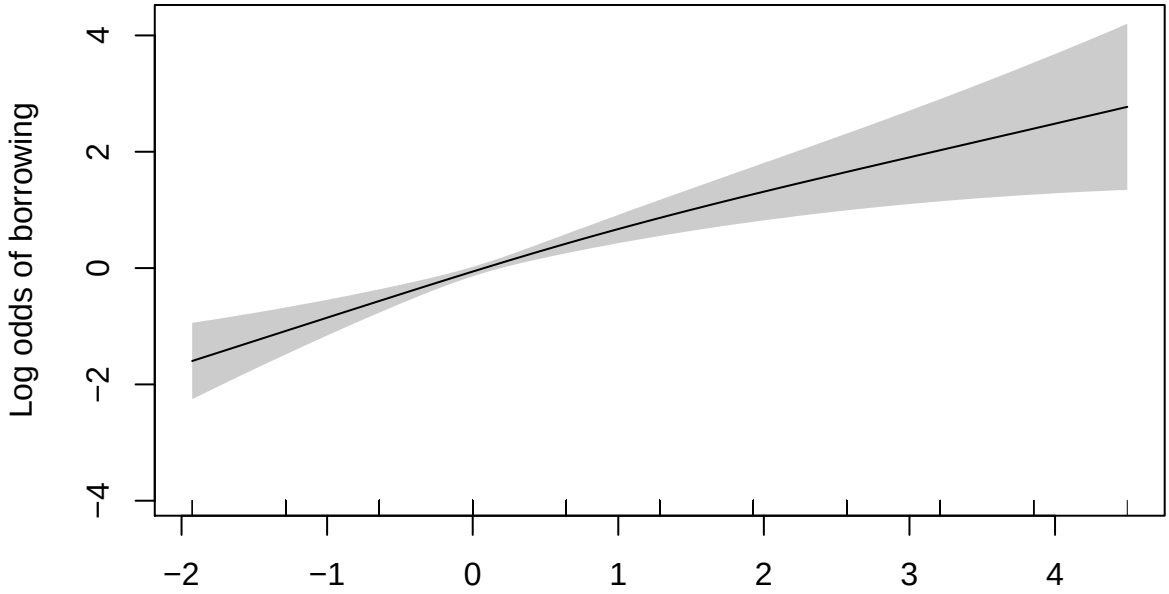
```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
## 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
```

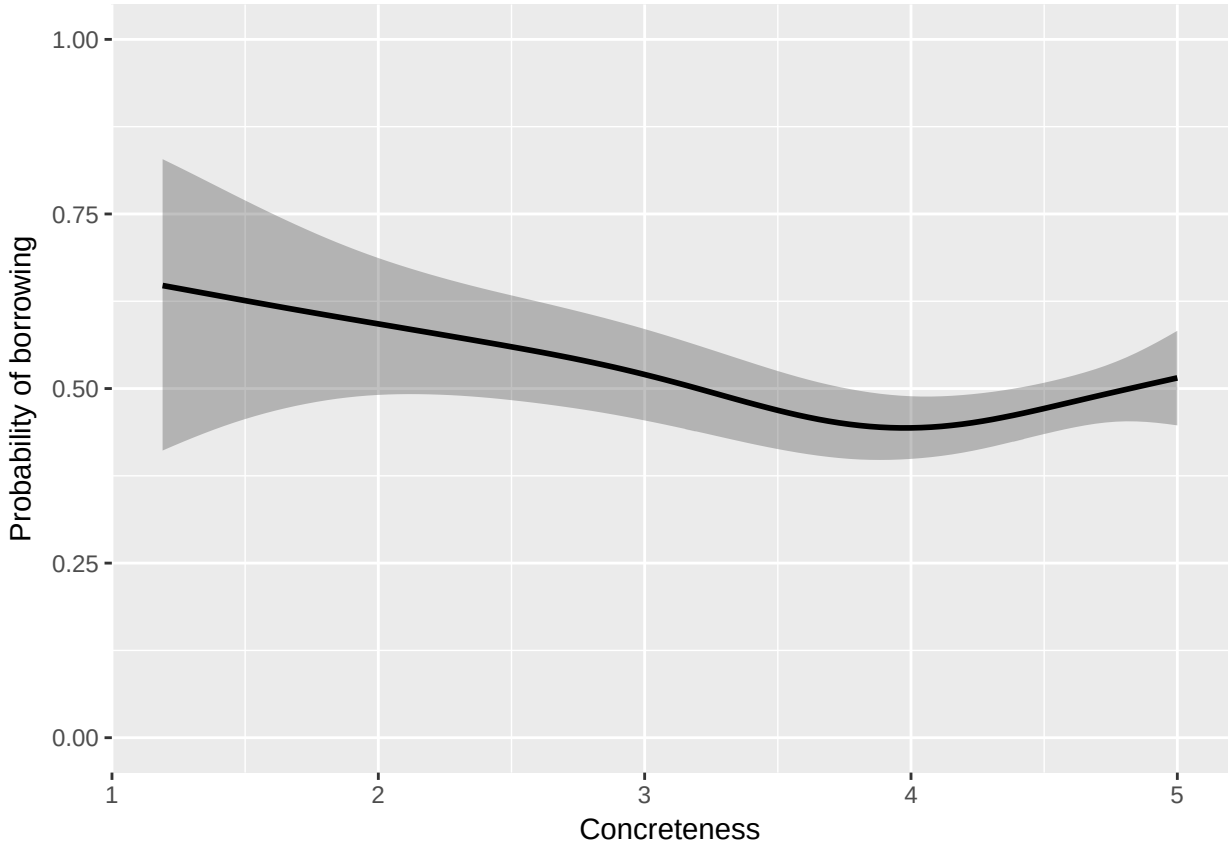
No significant improvement.

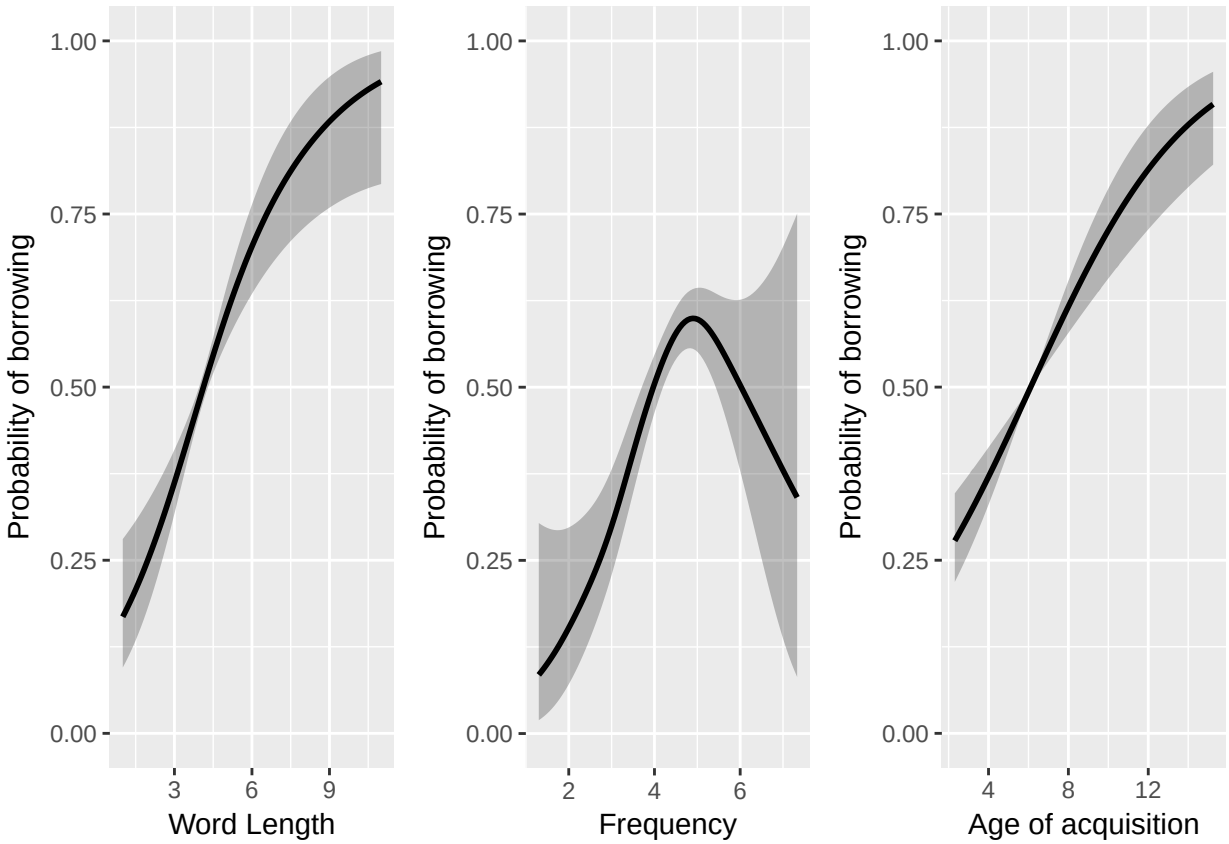
So no interactions are necessary.

Model plots

Plot the model estimates, changing the dependent scale to probability and the independent variables to their original scales. The code uses the `itsadug` package and then rescales the variables back to the original units. This code is hidden, but you can view it in the Rmd file.

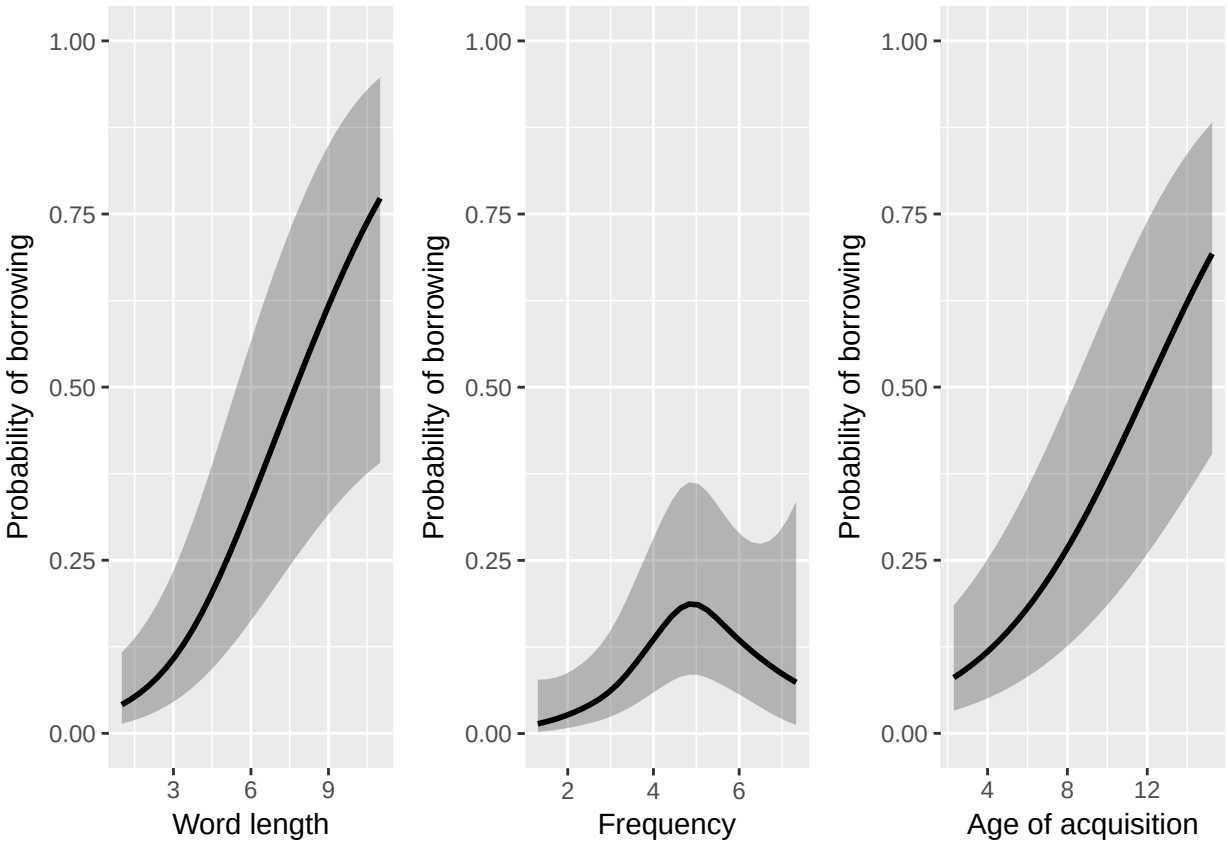







```
## pdf
##   2
```

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.

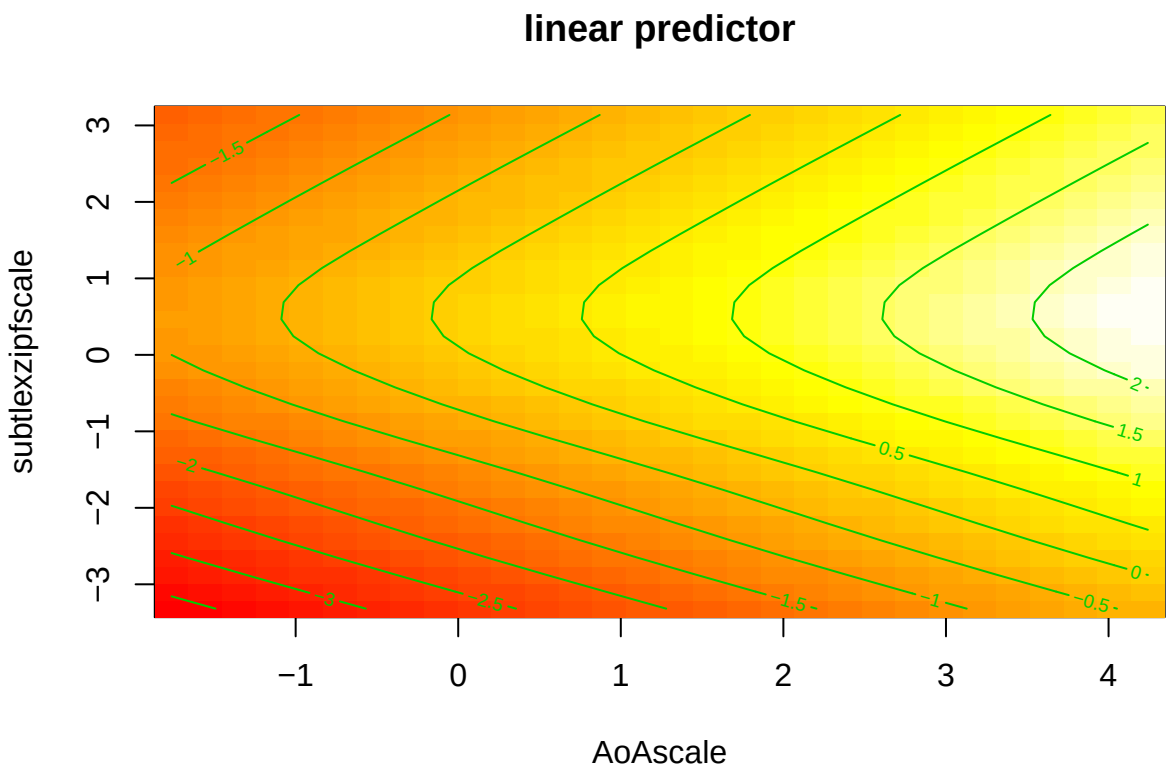



```
## pdf
##   2
```

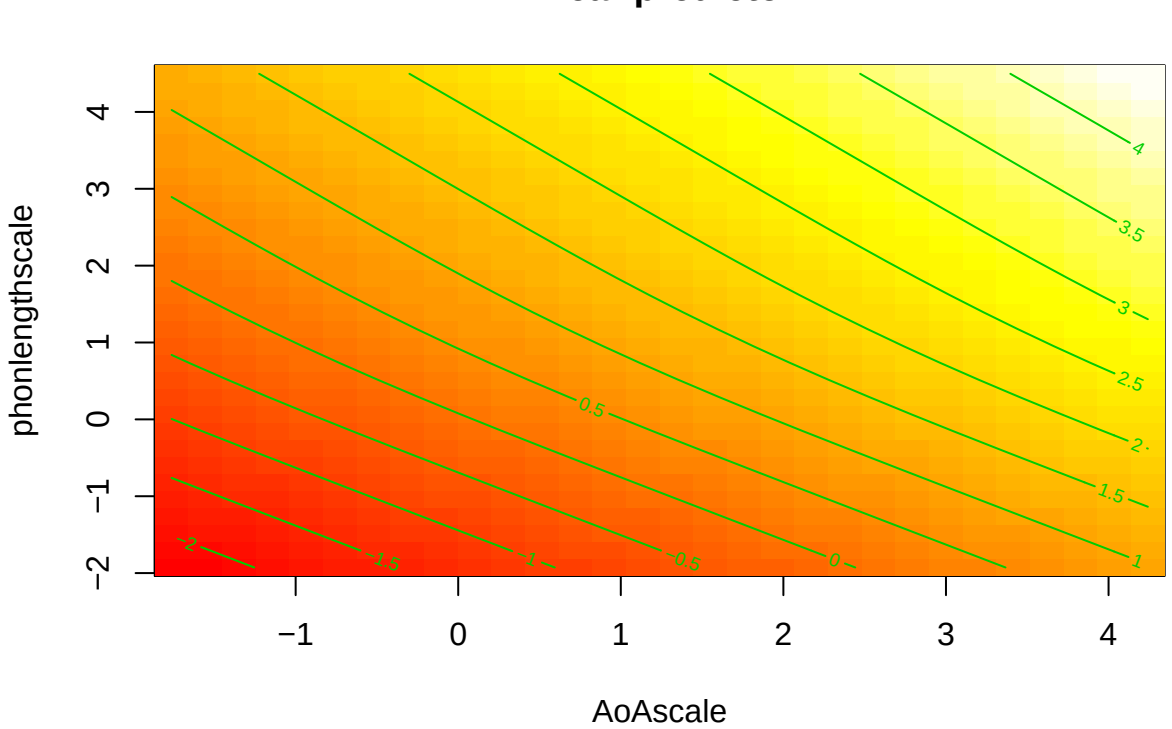
Contour plots

Visualise relationships between variables.

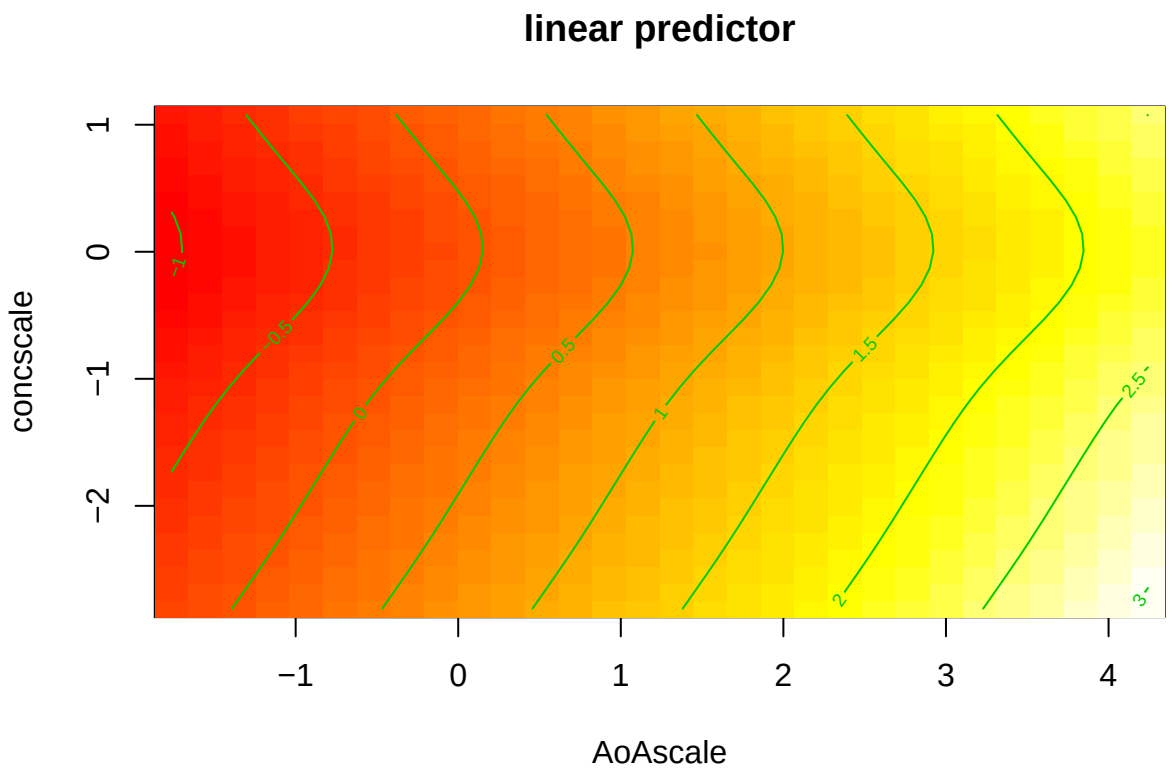
```
vis.gam(m0,view = c("AoAscale","subtlexzipfscale"),plot.type = "contour")
```



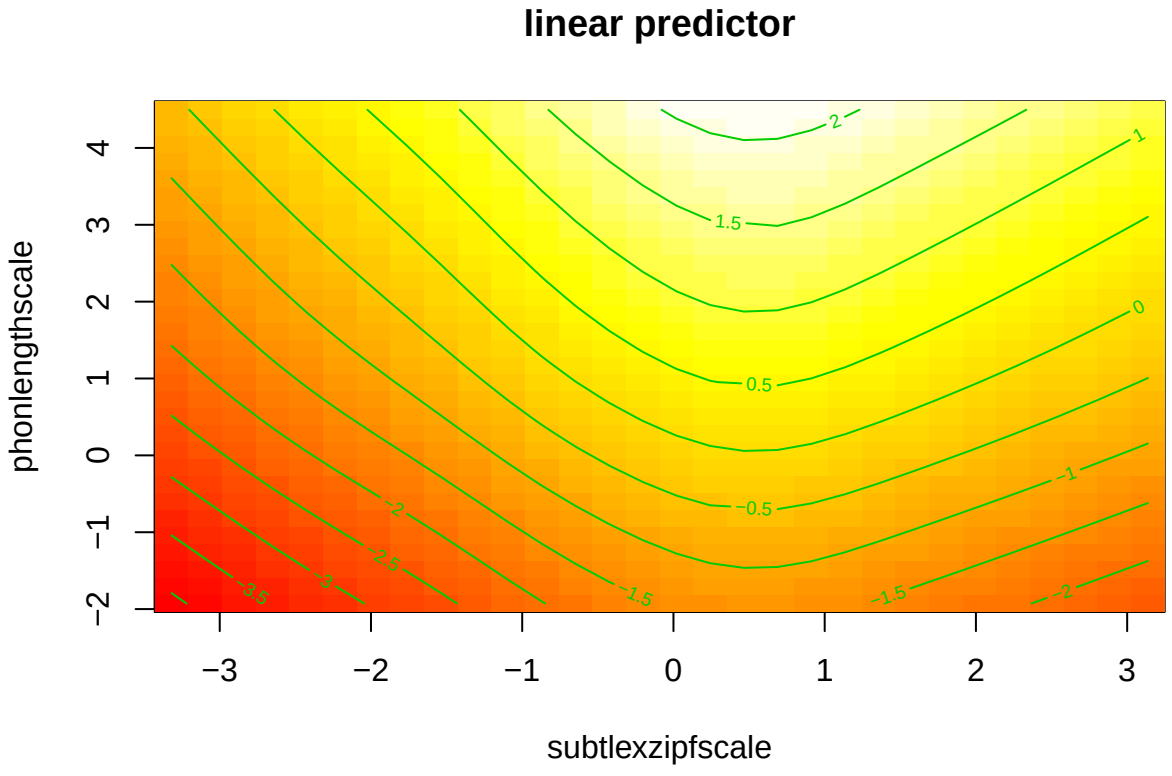

```
vis.gam(m0,view = c("AoAscale","phonlengthscale"),plot.type = "contour")
```



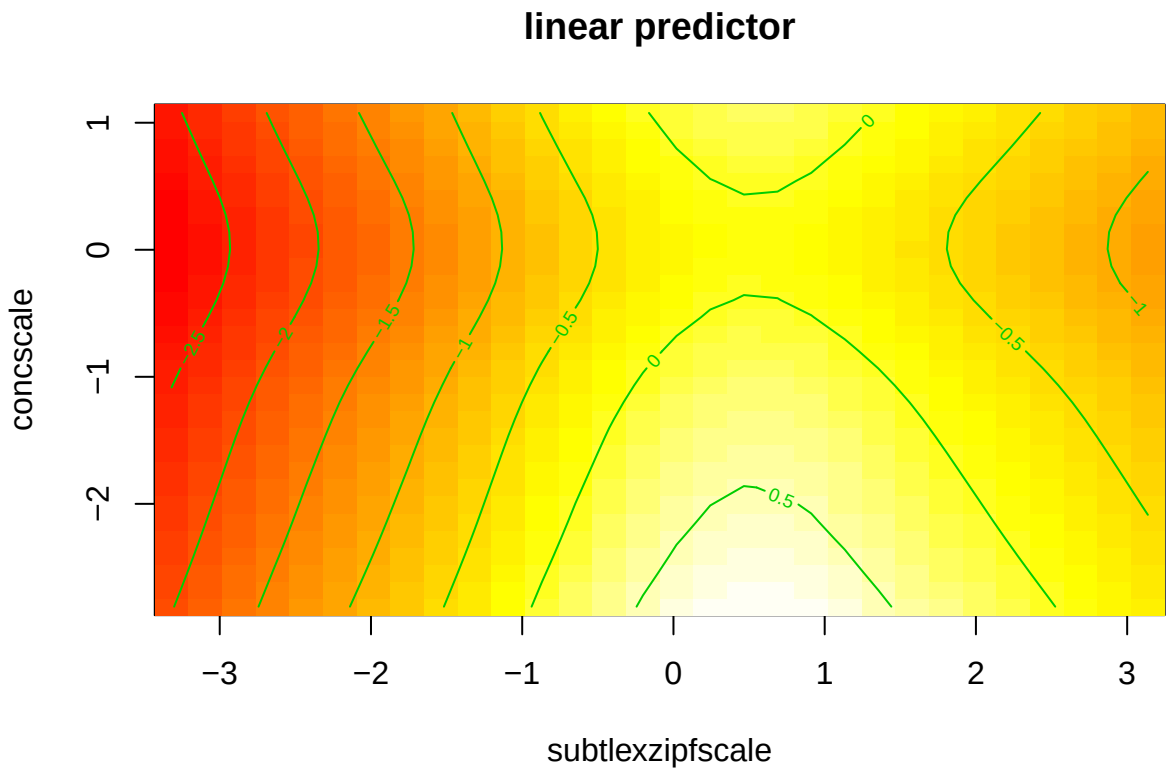

```
vis.gam(m0,view = c("AoAscale","concscale"),plot.type = "contour")
```



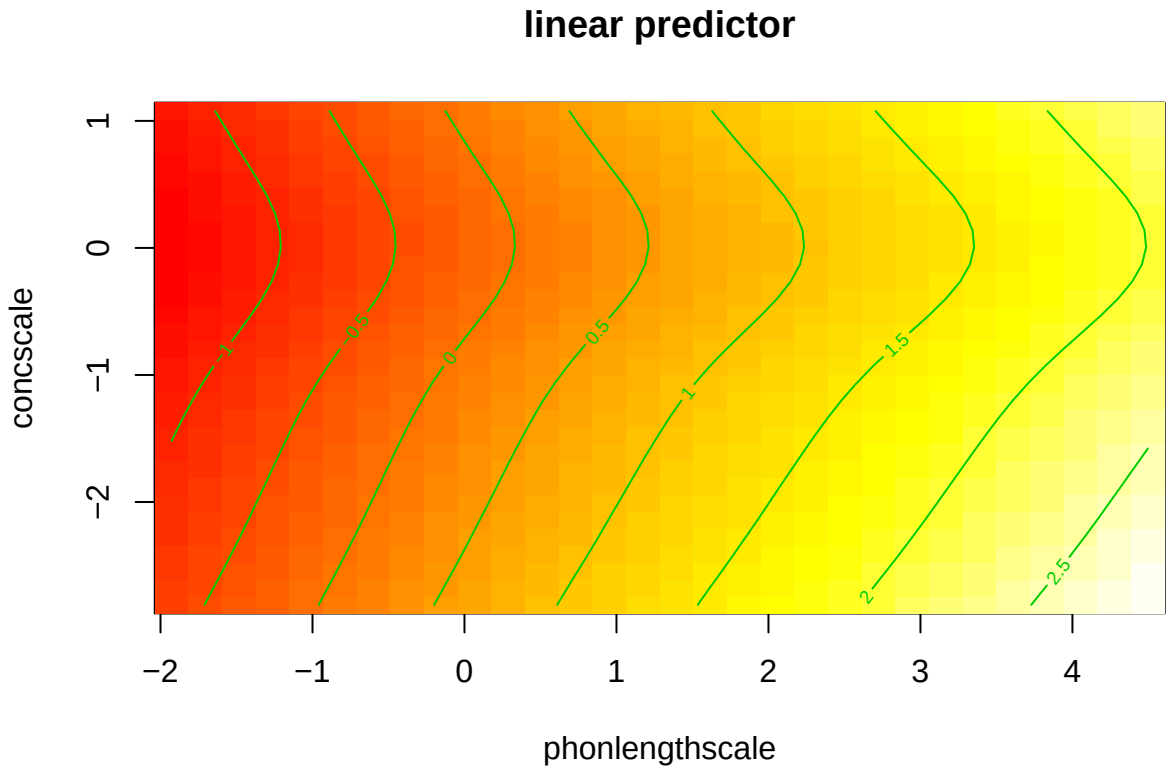

```
vis.gam(m0,view = c("subtlexzipfscale","phonlengthscale"),plot.type = "contour")
```




```
vis.gam(m0,view = c("subtlexzipfscale","concscale"),plot.type = "contour")
```




```
vis.gam(m0,view = c("phonlengthscale","concscale"),plot.type = "contour")
```



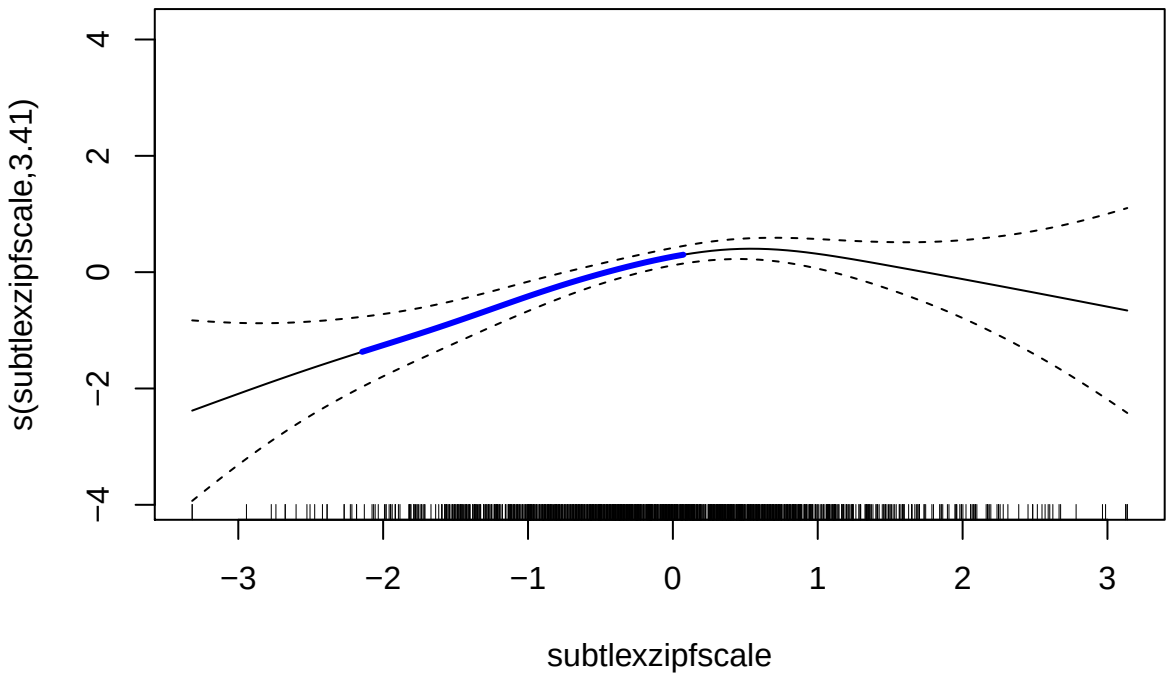
Significant trends

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.

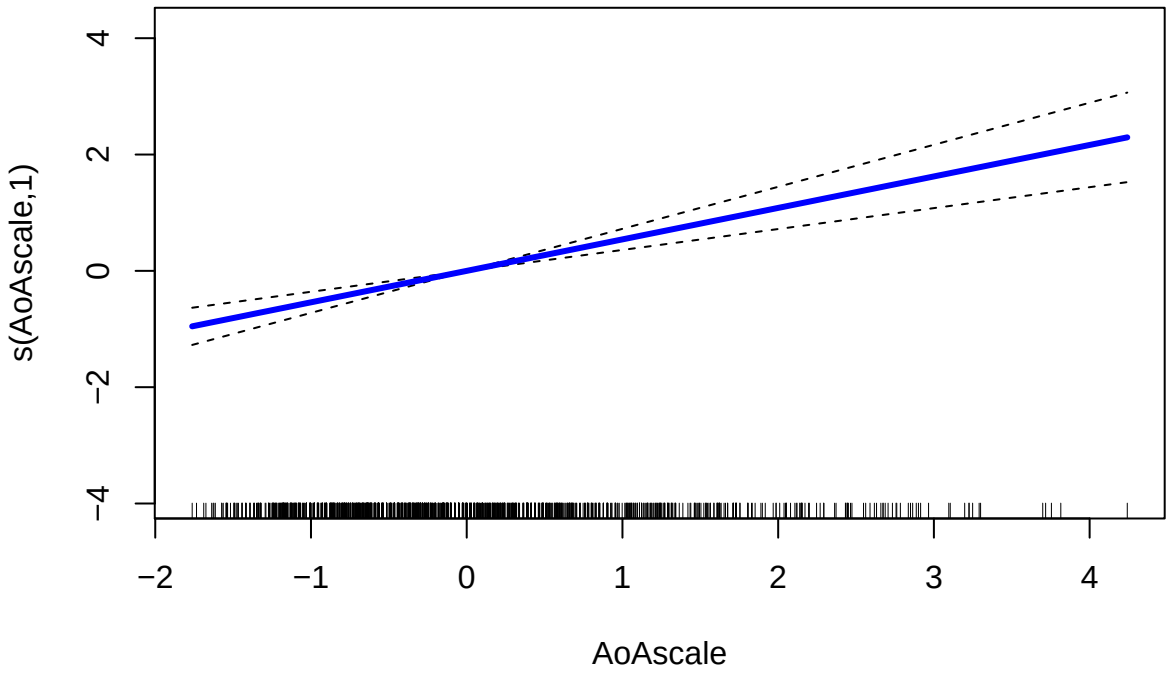
The results suggest:

- 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.

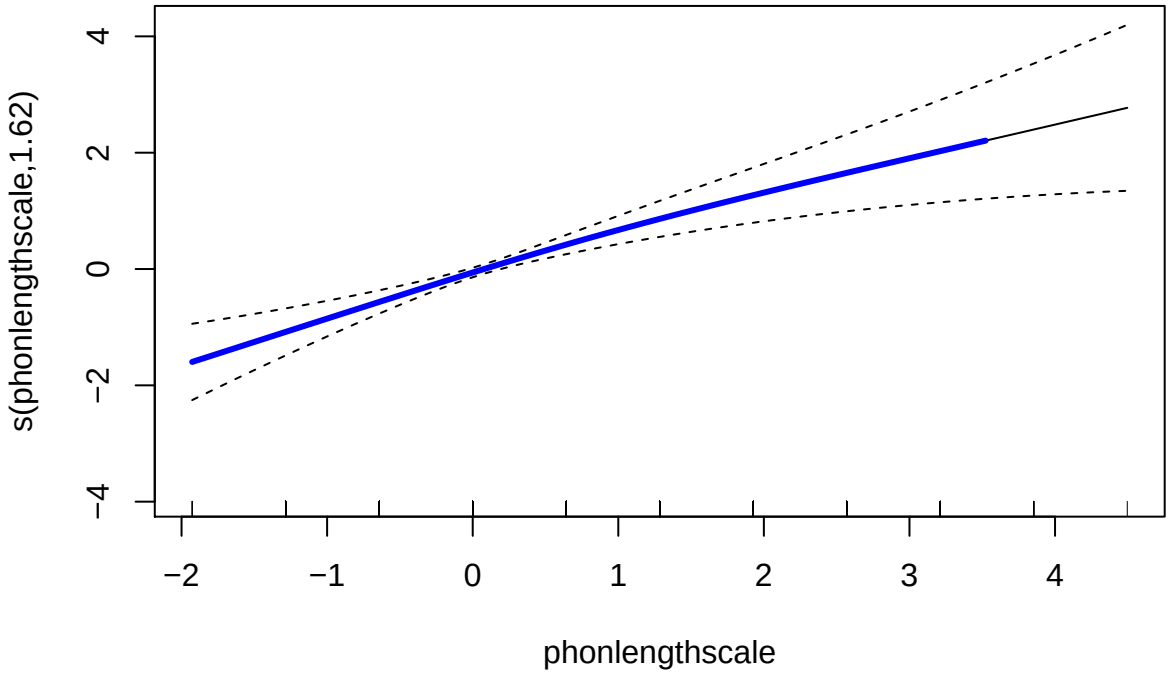
```
plotGAMSignificantSlopes(m0,"subtlexzipfscale","Frequency")
```



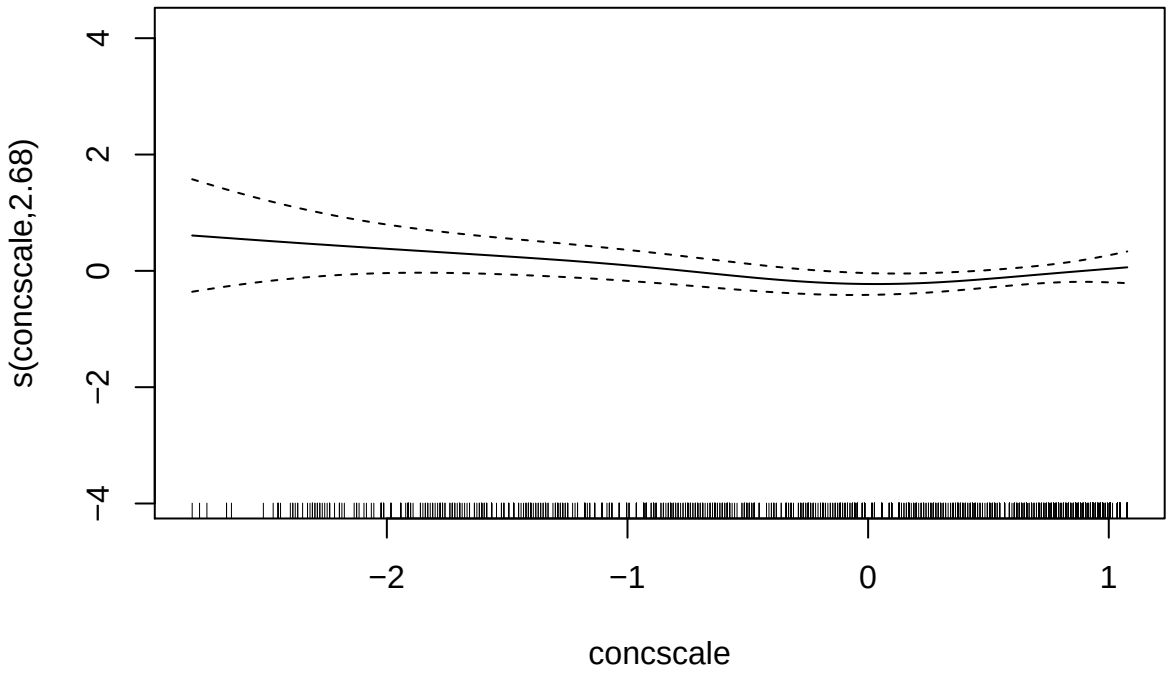

```
plotGAMSignificantSlopes(m0,"AoAscale","AoA")
```




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




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


s(concscale,2.68)
4
2
0
−2
−4
−2
−1
0
1
concscale

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.

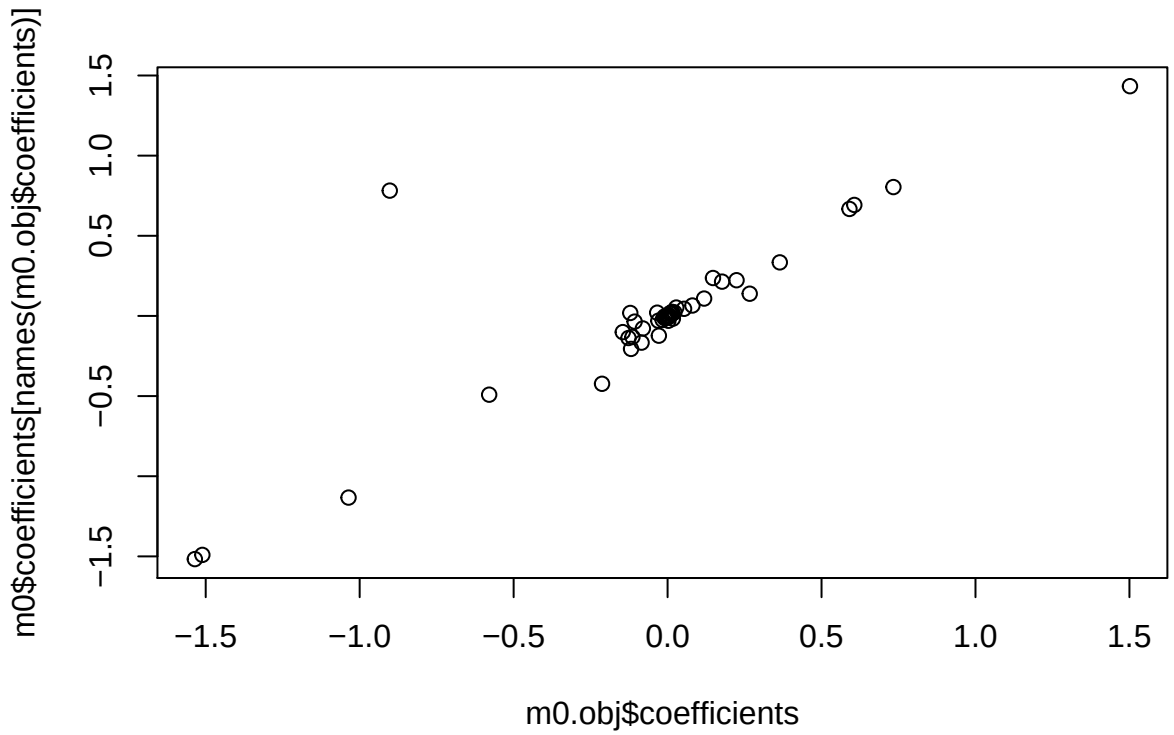
```
m0.obj = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoA_objscaled, k=3) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoA_objscaled,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),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

```
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`.

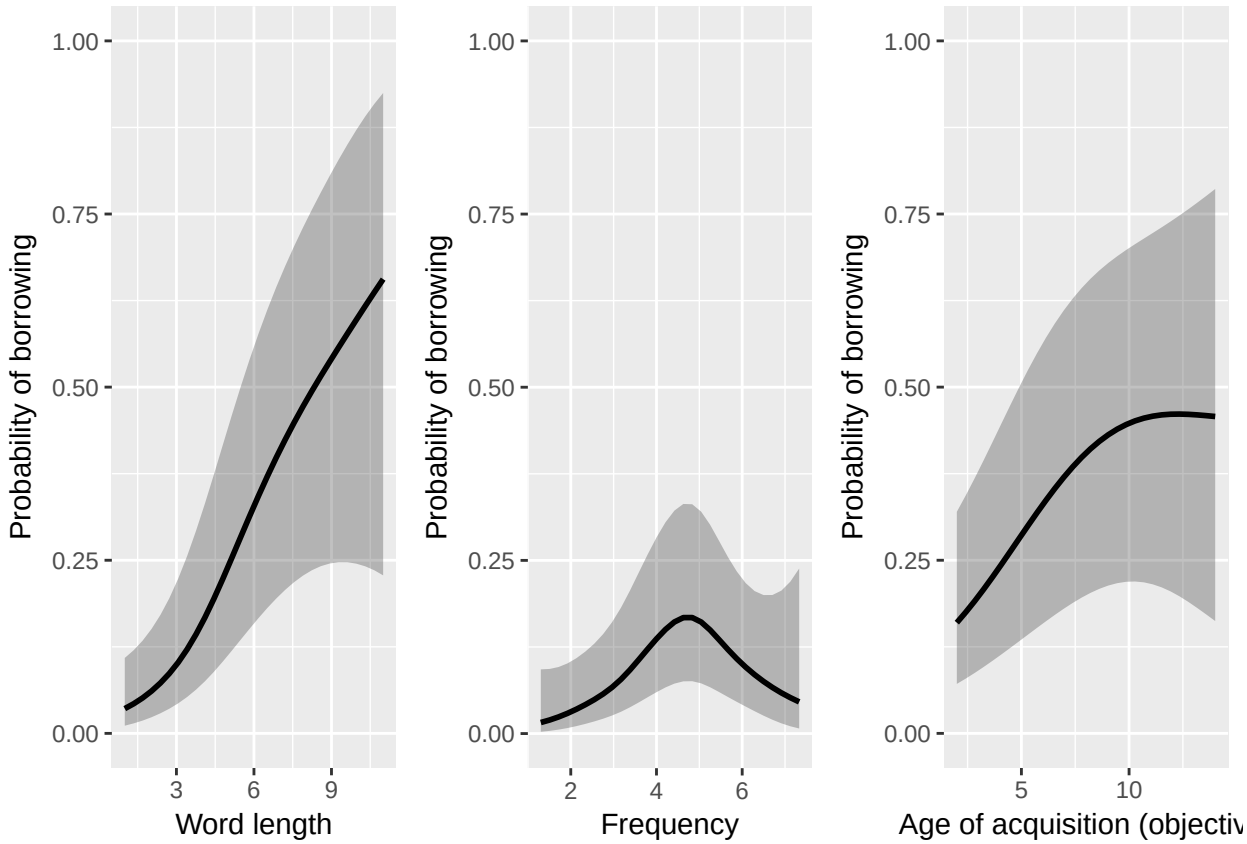
And chi squared terms are similar:

```
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.



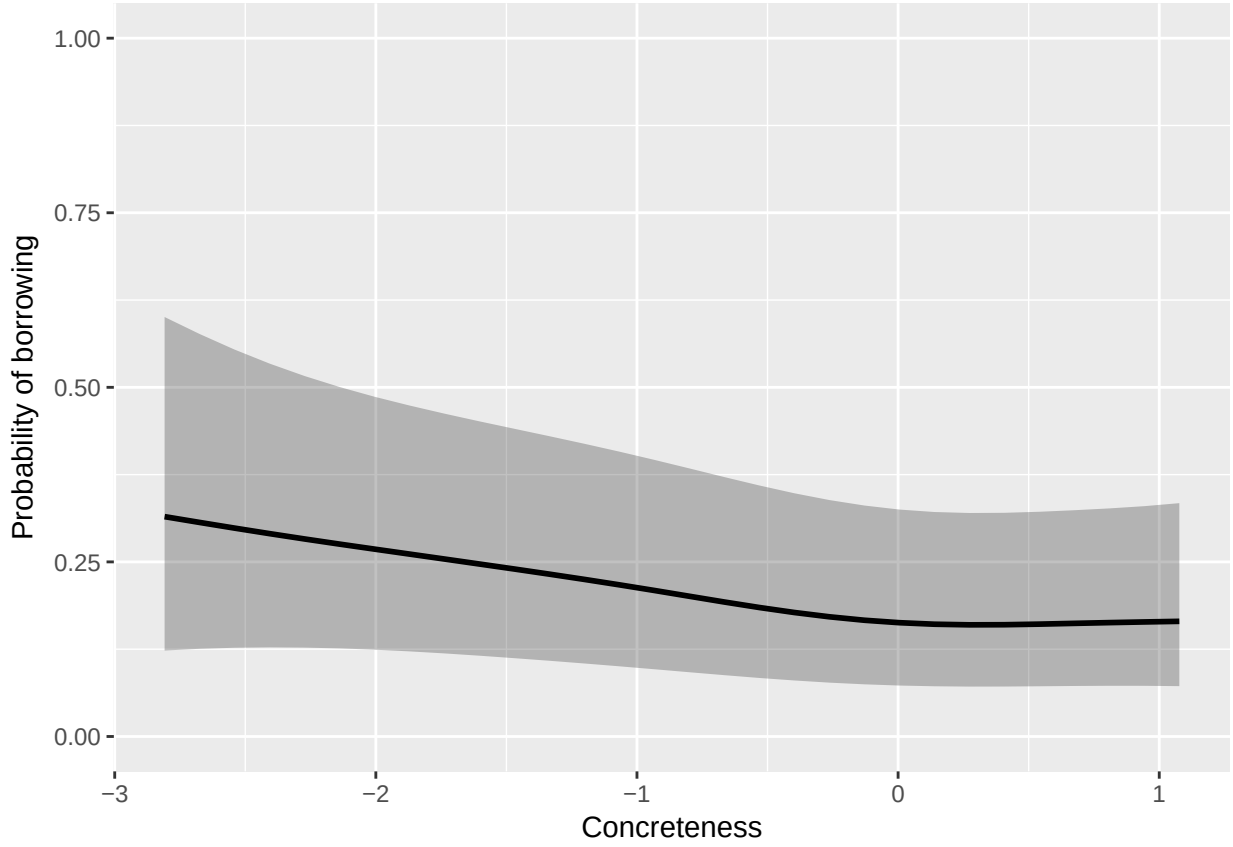

```
## pdf
##   2
```

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:

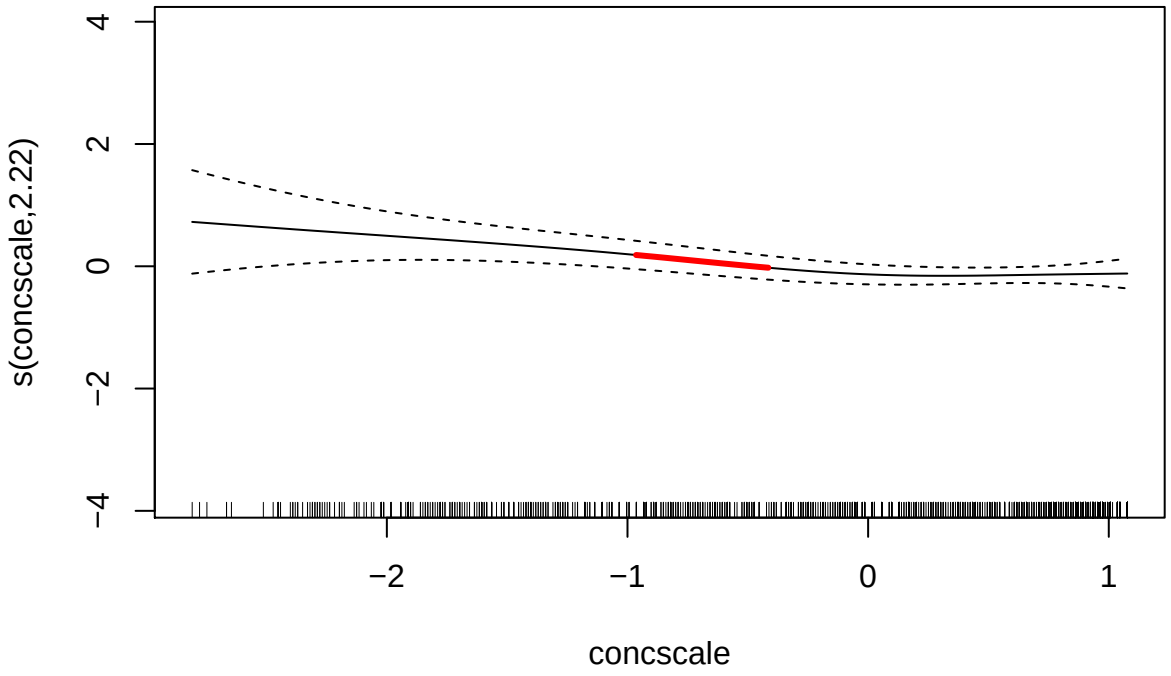
```
px = plot_smooth(m0.obj,view="concscale", rm.ranef = T, print.summary=F)
```

```
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")
```

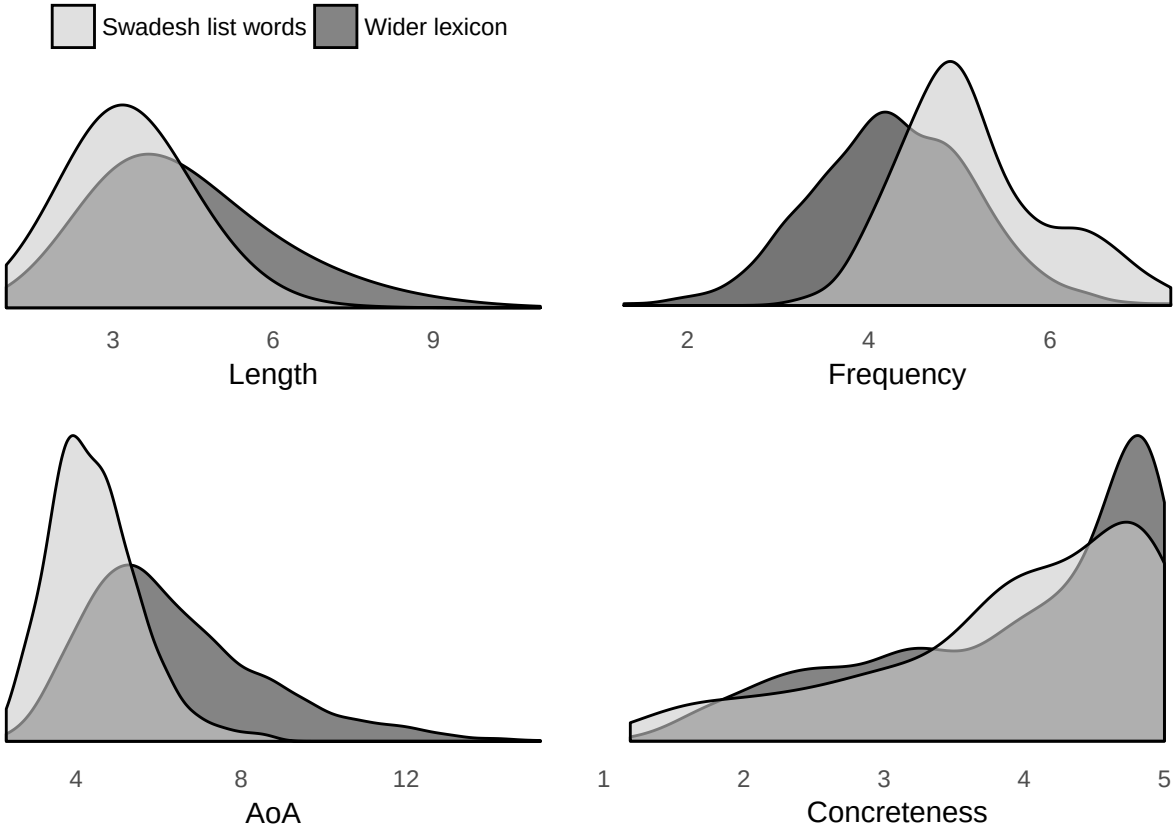


Distribution of words

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):

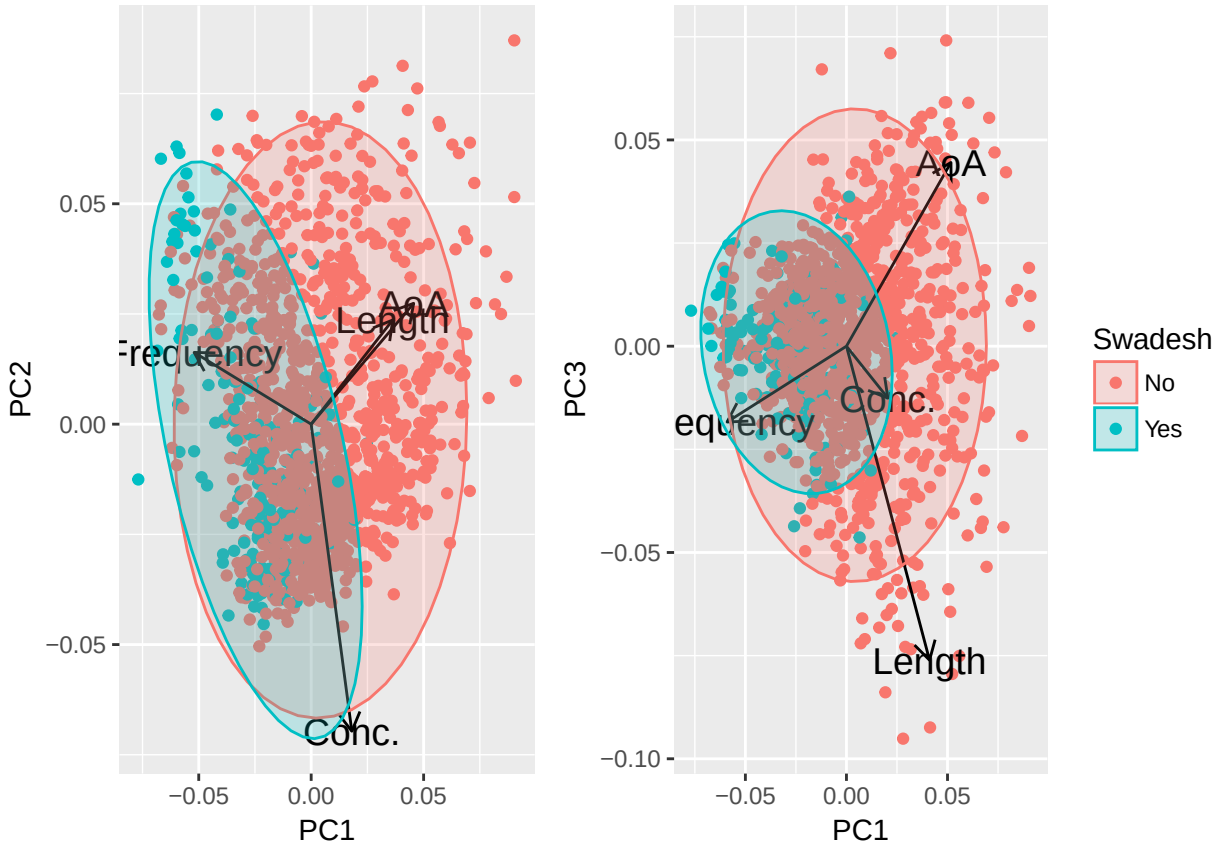



```
## pdf
##   2
```

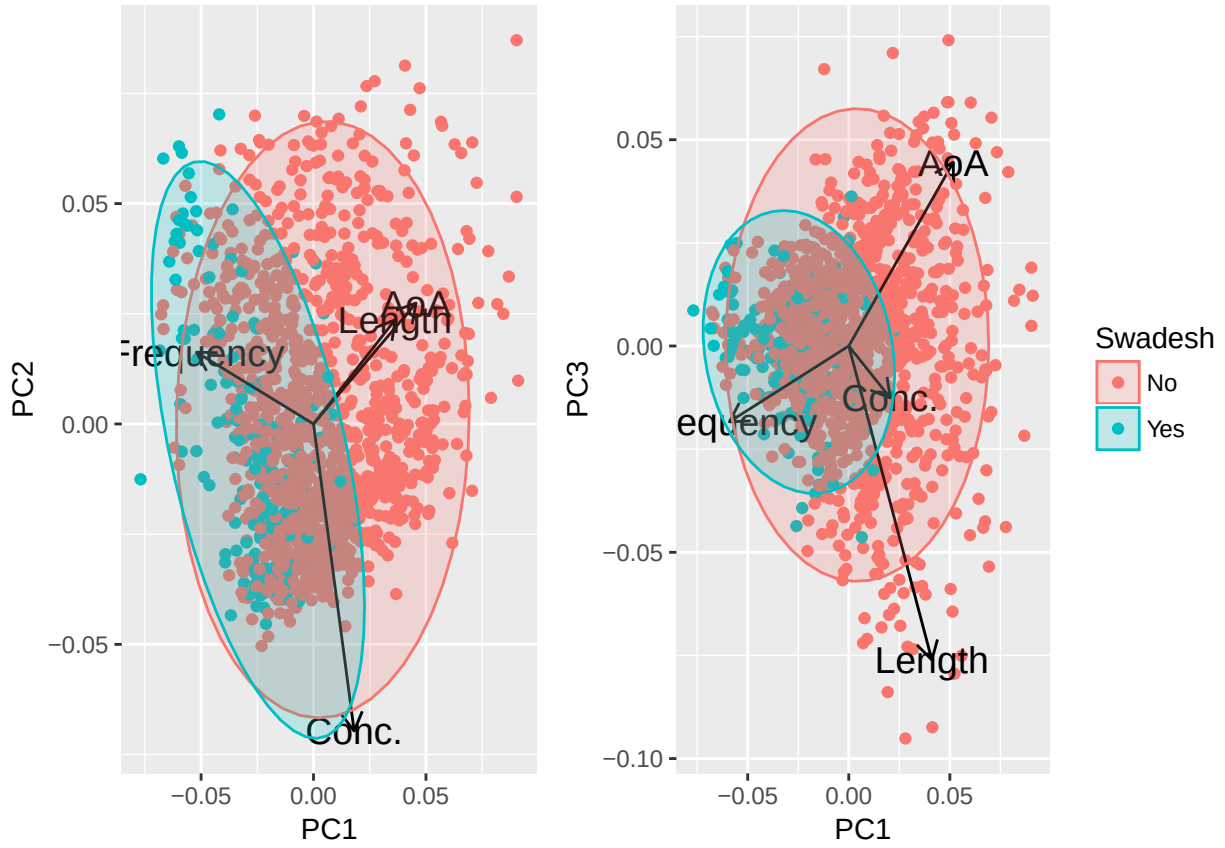
Try a PCA plot:

```
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))
```




```
plot(gxpc123)
```




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

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

Sensitivity analyses

Individual models for each variable

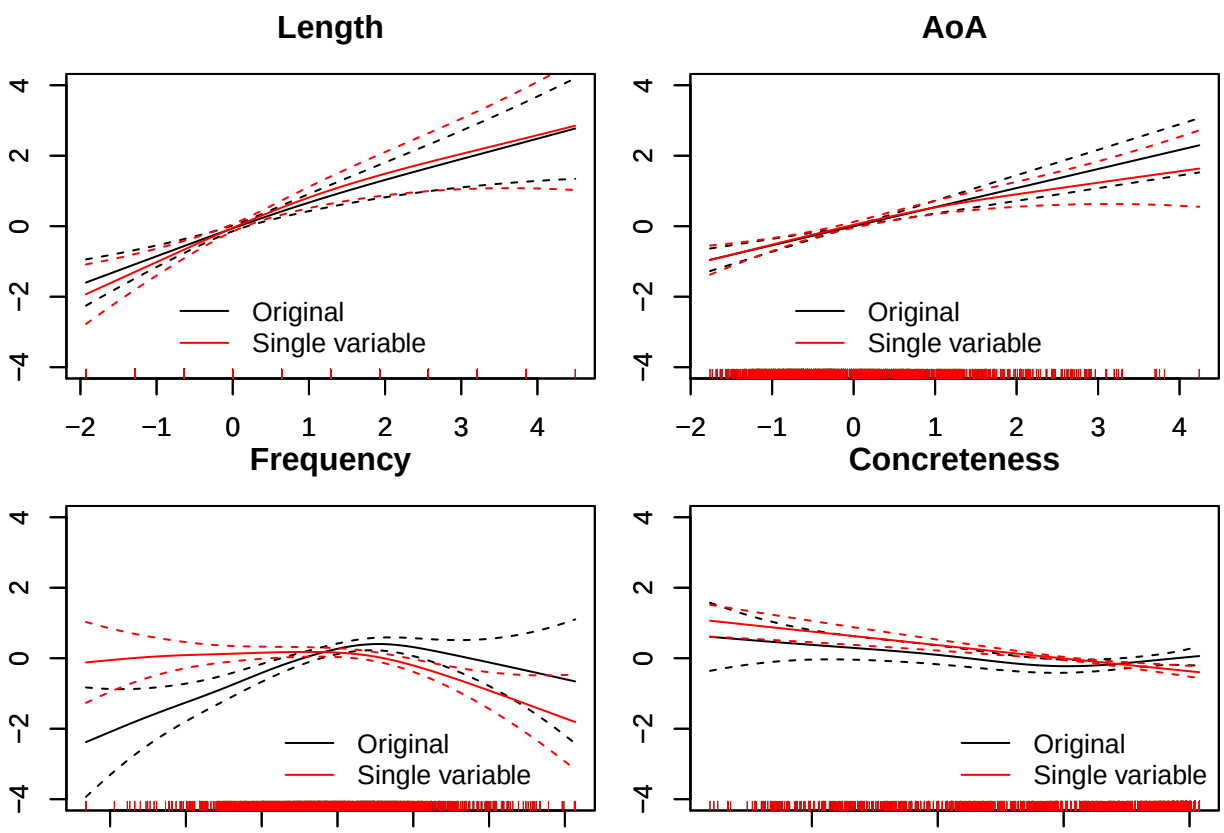
Check that the response function is similar when including a variable alone in a model. Run seperate models with single predictors:

```
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:

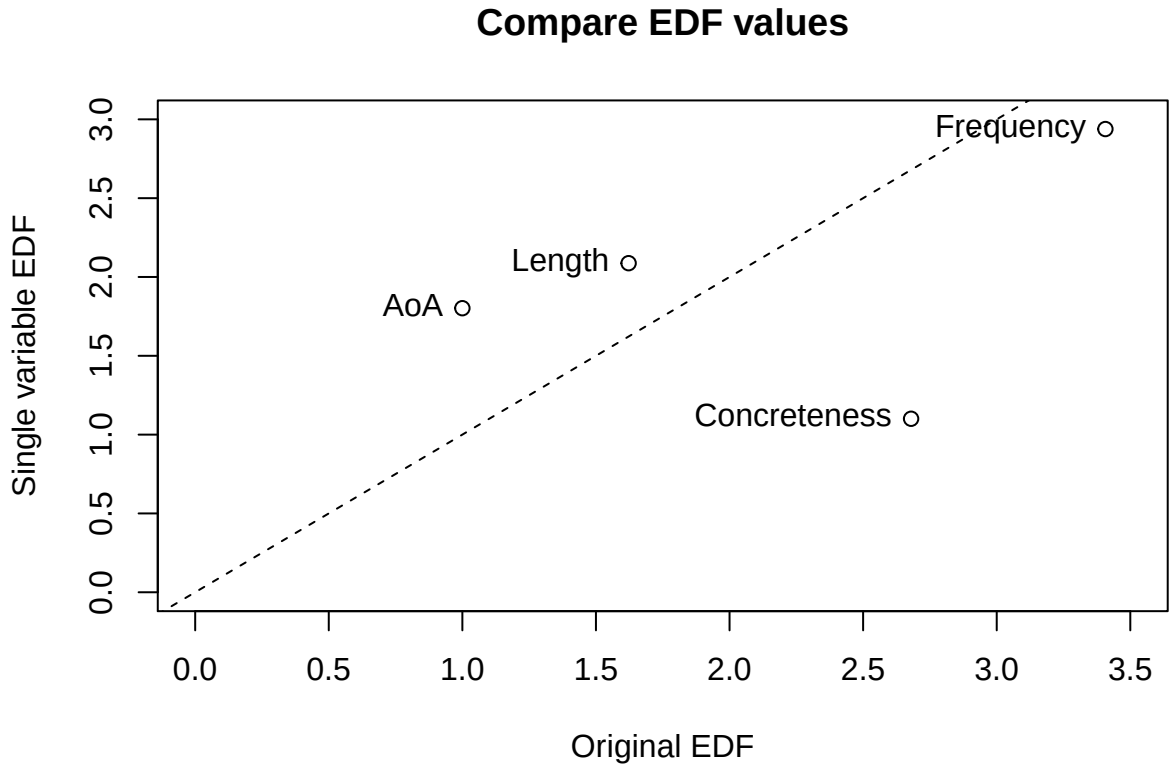
```
par(mfrow=c(2,2), mar=c(1,2,3,1))
mx = list(m0.length, m0.AoA, m0.frequency,m0.conc)
mx.labels = c("Length","AoA","Frequency","Concreteness")
for(i in 1:4){
plot(m0,select=i,main=mx.labels[i],ylim=c(-4,4))
par(new=T)
plot(mx[[i]],select=1,ylim=c(-4,4), col=2,ylab="")
legend(-1,-1.5,legend = c("Original","Single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




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

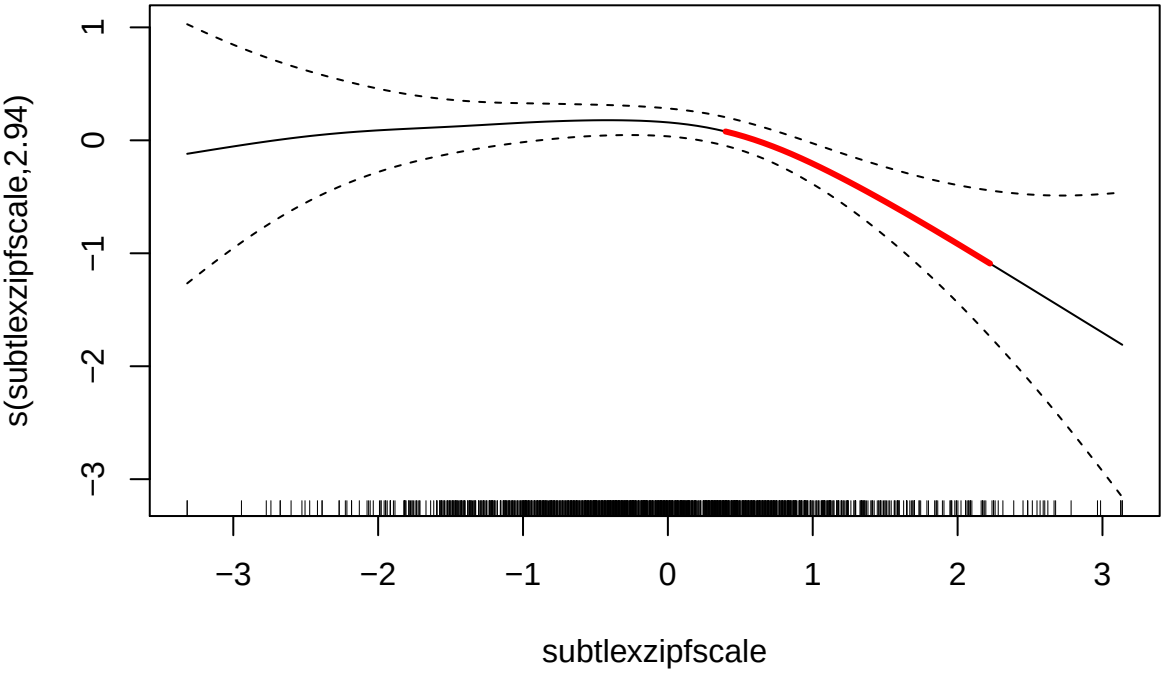
Compare the EDF values.

```
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)
```



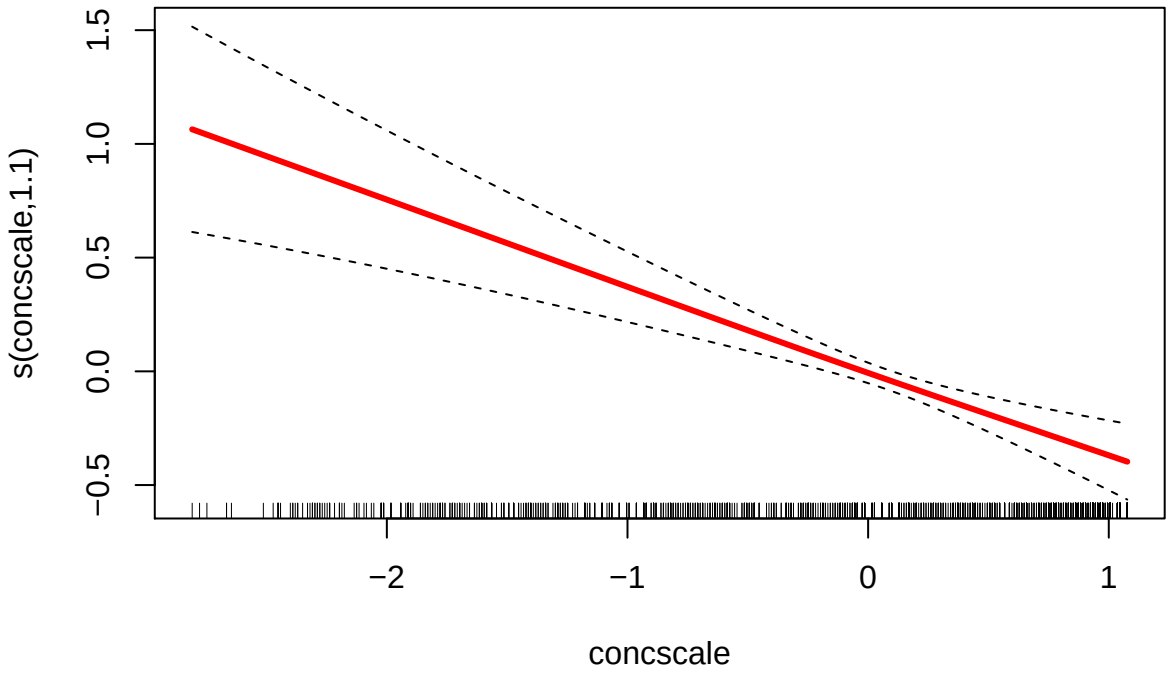
The results for length and AoA are almost identical. The results for frequency are similar (non-linear relationship with a peak in the middle), although the significant slope is now for the higher values:

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



By itself, concreteness is a signficant predictor.

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

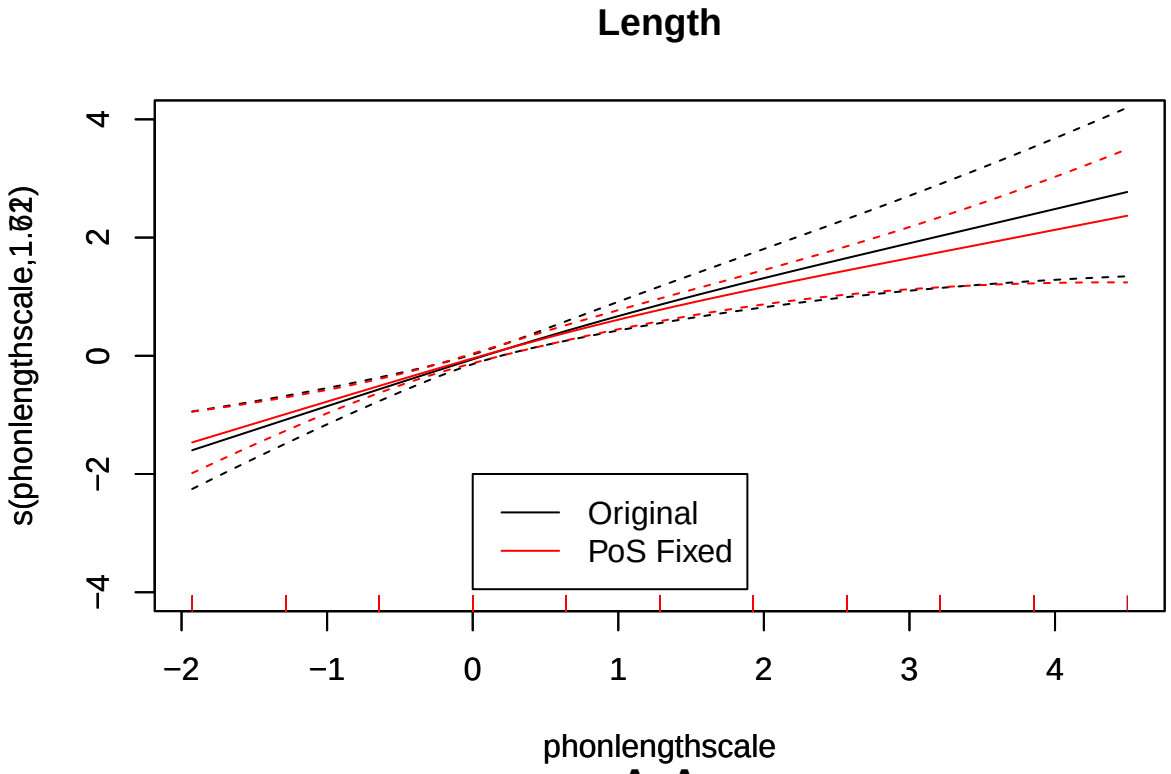


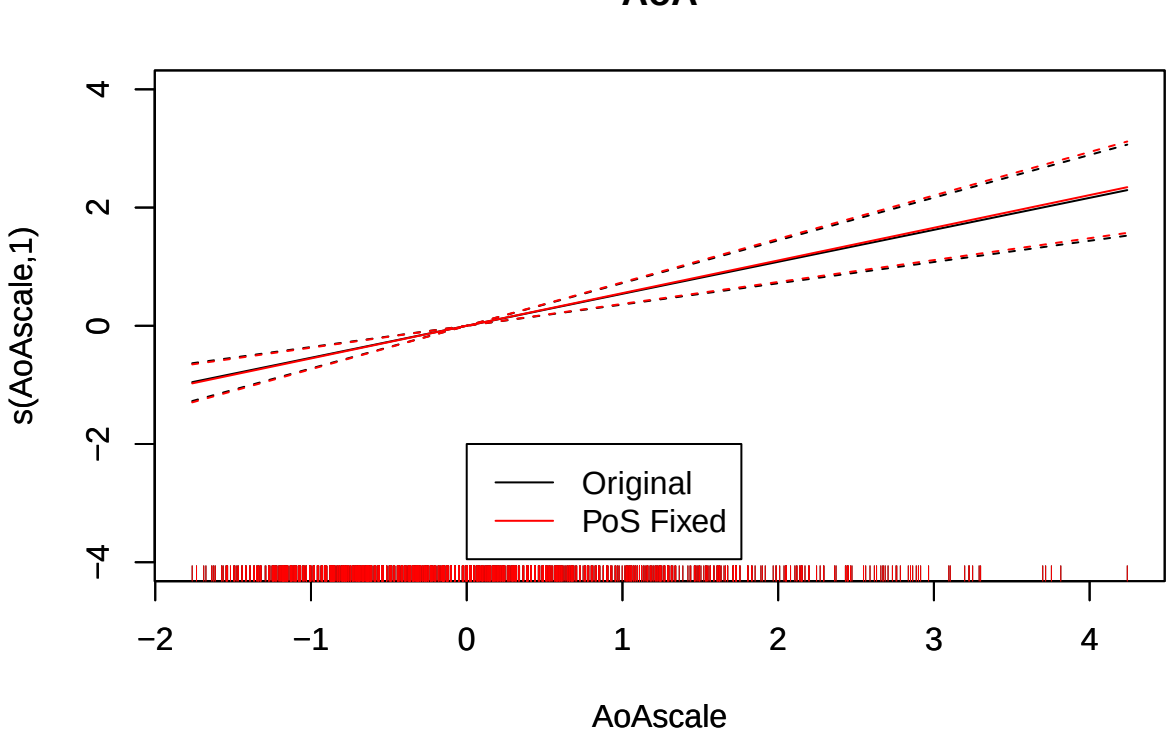
GAM with PoS as fixed effects

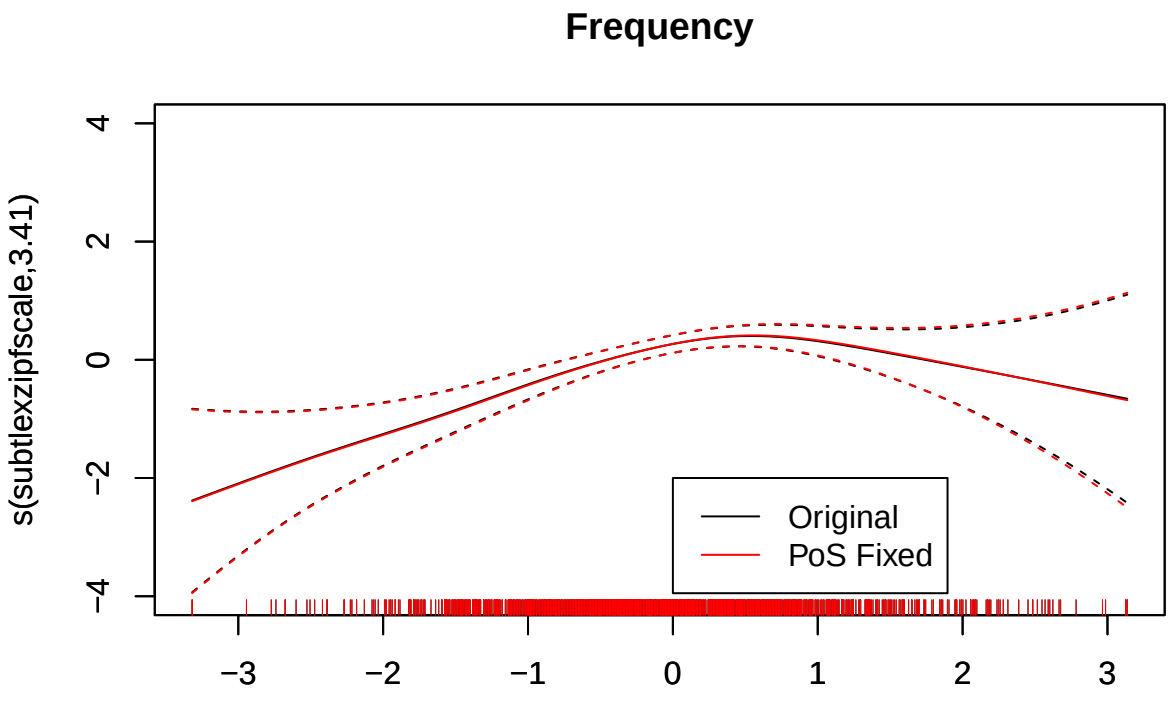
Part of speech was modelled above as a random effect. Here we show that the estimates differ very little if we treat part of speech as a fixed effect:

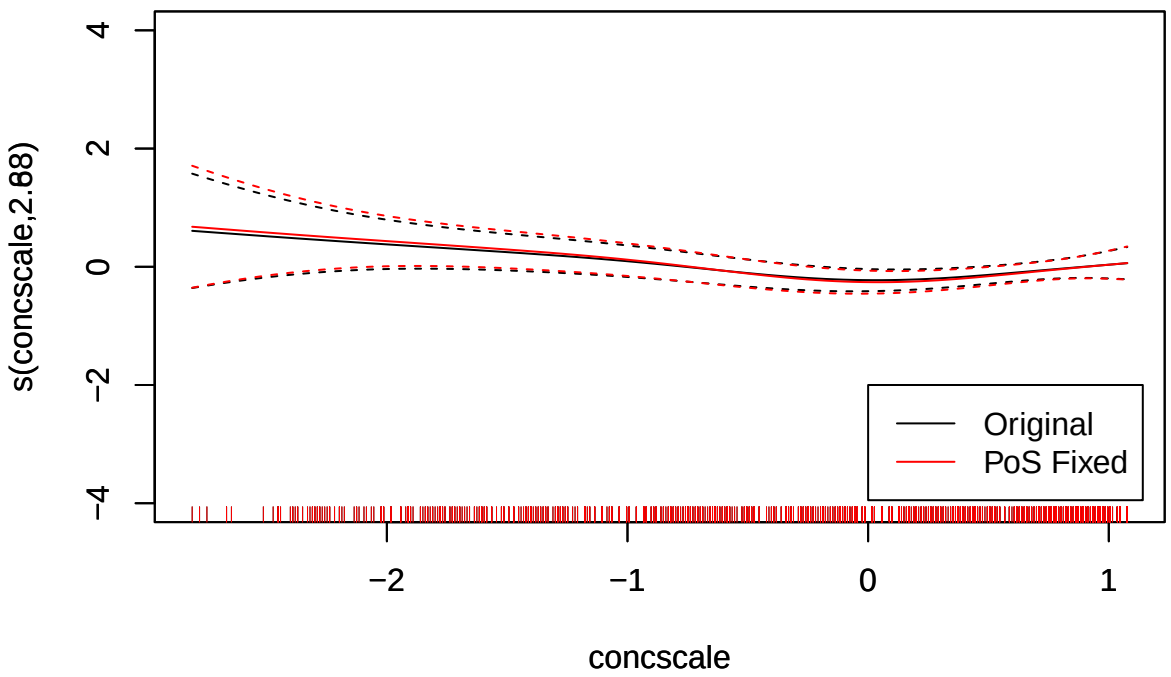
```
m0.posFE = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(concscale) +
      cat,
    data = dataloan2,
    family='binomial')
```

```
vars = c("phonlengthscale","AoAscale","subtlexzipfscale","concscale")
varLabels = c("Length","AoA","Frequency","Concreteness")

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)
}
```


Length
s(phonlengthscale,1.62)
4
2
0
−2
−4
−2
−1
0
1
2
3
4
Original
PoS Fixed
phonlengthscale


AoA
s(AoAscale,1)
4
2
0
−2
−4
−2
−1
0
1
2
3
4
Original
PoS Fixed
AoAscale





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

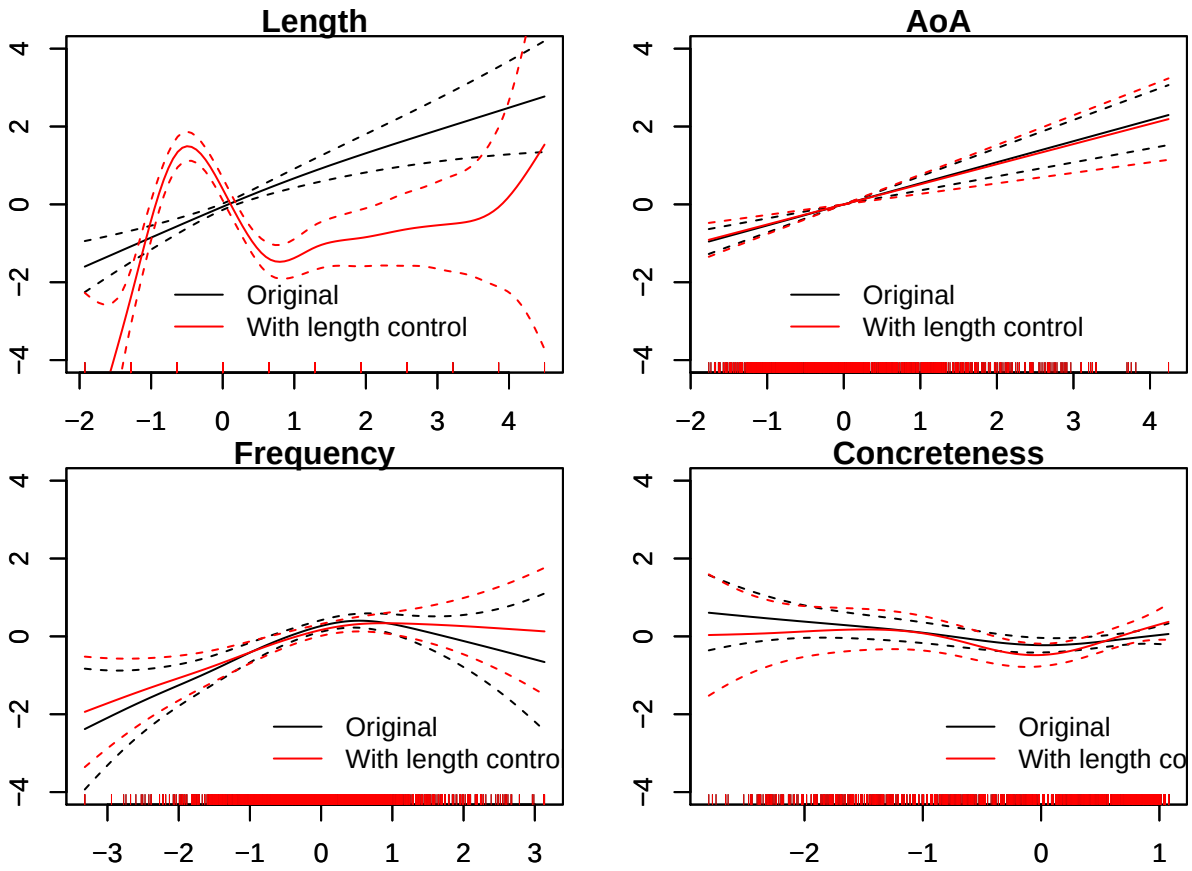
Use the source word length frequency measure to control for borrowing from langauges with longer words.

```
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)  -1.3973     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  97.614  < 2e-16 ***
## s(AoAscale)             1.000e+00  1.000  17.564 2.78e-05 ***
## s(subtlexzipfscale)     2.614e+00  3.356  16.784 0.001305 **
## s(concscale)            3.540e+00  4.392  11.256 0.035537 *
## s(cat)                  5.221e+00 11.000  24.783 0.000617 ***
## s(cat,phonlengthscale)  9.873e-01 11.000   2.630 0.115570
## s(cat,AoAscale)         3.300e-01 11.000   0.419 0.287566
## s(cat,subtlexzipfscale) 4.823e-05 11.000   0.000 0.576074
## s(cat,concscale)        9.998e-01 11.000   1.926 0.364816
## s(SWF)                  1.979e+00  1.999 250.957  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.512   Deviance explained = 41.5%
## fREML =   3945  Scale est. = 1         n = 1317
```

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(-4,4))
par(new=T)
plot(mLenFreqControl,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')
}
```




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

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

```
mMeanLenControl = update(m0, ~.+s(SLMWL,k=3))
```

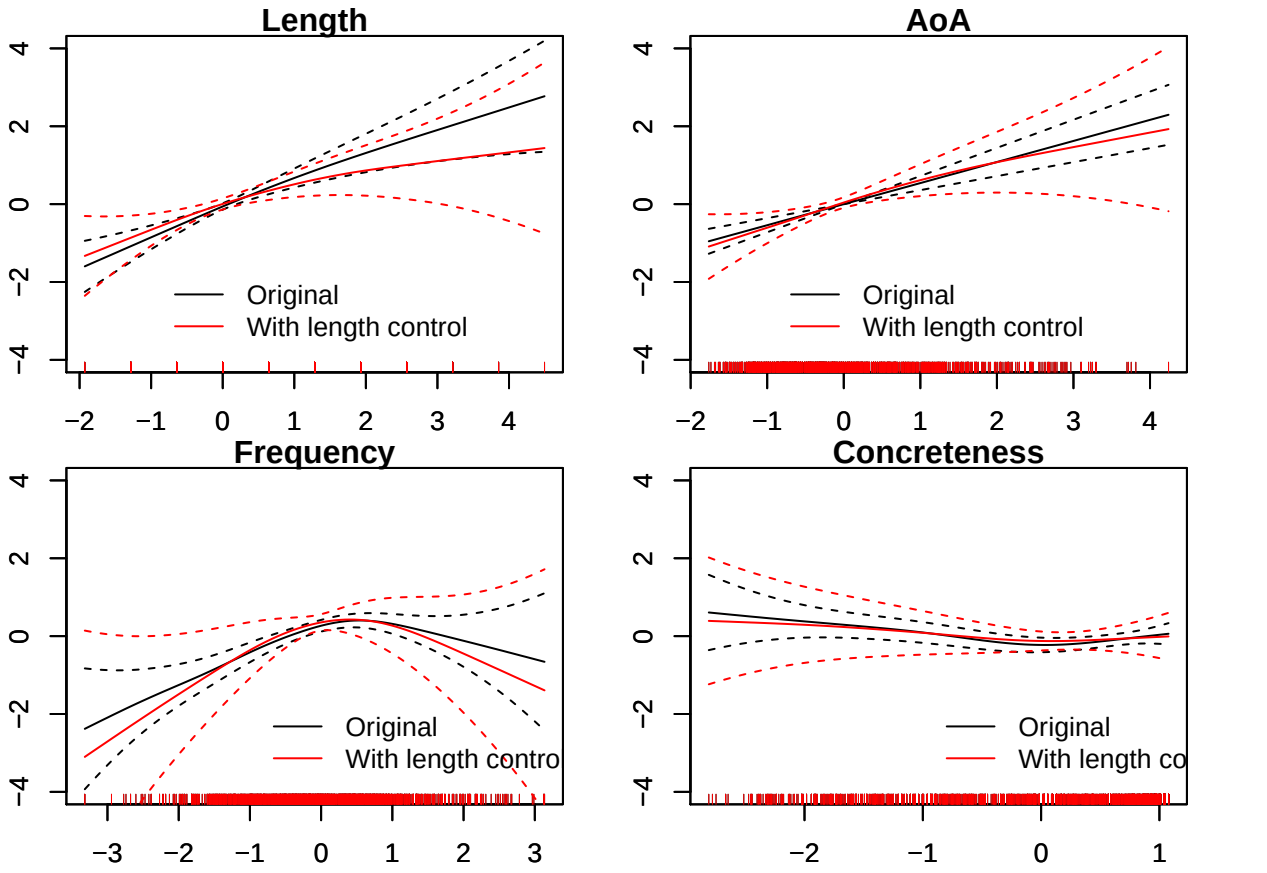
```
## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit

## Warning: Possible divergence detected in fast.REML.fit
```

```
summary(mMeanLenControl)
```

```
##
## 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(SLMWL, k = 3)
##
## 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(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
## s(SLMWL)                1.9352  1.995 92.037 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.702   Deviance explained = 46.6%
## fREML = 9.3393e+10  Scale est. = 1         n = 1317
```

```
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')
}
```




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