Supporting materials for Iconicity and diachronic language change

Contents

Introduction

This file shows the R code for the analyses in *Iconicity and diachronic langauge change*. We show whether a word is borrowed or not can be predicted by its iconicity.

Below is a list of variable names in the data with a short explanation:

- word: Orthographic form
- borrowing: variable from WOLD indicating level of evidence for borrowing (1 = definately borrowed, 5 = no evidence of borrowing).
- bor15, bor15.cat: 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).
- AoAObj: Objective, test-based age of acquisition 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.

- pagel_rate: Rate of lexical replacement from Pagel, Atkinson & Meade (2007)
- iconicity: Iconicity rating from Winter et al. (2017). The data actually comes from the supporting materials for Perry et al. (2018), which has slightly more data.
- systematicity: Systematicity ratings from Monaghan et al. (2014)
- borrowing_type: Whether a word was not borrowed versus different types of borrowing: replacement, coexistence or insertion. Values are NA if there is no information available.
- borrowing_type_num: The effect categories, turned into numbers for the GAM model.

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.

Brysbaert, M., & Biemiller, A. (2017). Test-based age-of-acquisition norms for 44 thousand English word meanings. Behavior research methods, 49(4), 1520-1523.

Brysbaert, M., Warriner, A. B., & Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. Behavior research methods, 46(3), 904-911.

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.

Monaghan, P., Lupyan, G., & Christiansen, M. (2014). The systematicity of the sign: Modeling activation of semantic attributes from nonwords. In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 36, No. 36).

Winter, B., Perlman, M., Perry, L. K., & Lupyan, G. (2017). Which words are most iconic?. Interaction Studies, 18(3), 443-464.

Perry, L. K., Perlman, M., Winter, B., Massaro, D. W., & Lupyan, G. (2018). Iconicity in the speech of children and adults. Developmental Science, 21(3), e12572.

Load libraries

```r
library(mgcv)
library(lattice)
library(ggplot2)
library(party)
library(gridExtra)
library(Hmisc)
library(survival)
library(Formula)
library(tidyverse)
```

Custom functions for rescaling gam results and making table of correlations. This also loads the custom script *GAM_derivatives.R* which runs the analysis of slope significance.

```r
# functions for plotting:
logit2per = function(X){
  return(exp(X)/(1+exp(X)))
}

corstars <-function(x, method=c("pearson", "spearman"), removeTriangle=c("upper", "lower"),
                    result=c("none", "html", "latex")){
  #Compute correlation matrix
  require(Hmisc)
  x <- as.matrix(x)
  correlation_matrix<-rcorr(x, type=method[1])
  R <- correlation_matrix$r # Matrix of correlation coeficients
  p <- correlation_matrix$P # Matrix of p-value

  ## Define notions for significance levels; spacing is important.
  mystars <- ifelse(p < .001, "***", ifelse(p < .01, "**", ifelse(p < .05, "* ", "  ")))

  ## trunctuate the correlation matrix to two decimal
  R <- format(round(cbind(rep(-1.11, ncol(x)), R), 3))[,-1]

  ## build a new matrix that includes the correlations with their apropriate stars
  Rnew <- matrix(paste(R, mystars, sep=""), ncol=ncol(x))
  diag(Rnew) <- paste(diag(R), " ", sep="")
  rownames(Rnew) <- colnames(x)
  colnames(Rnew) <- paste(colnames(x), "", sep="")

  ## remove upper triangle of correlation matrix
  #if(removeTriangle[1]=="upper"){
    Rnew <- as.matrix(Rnew)
    Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
    Rnew <- as.data.frame(Rnew)
  #}

  ## remove lower triangle of correlation matrix
  if(removeTriangle[1]=="lower"){
    Rnew = t(Rnew)
    Rnew = Rnew[,-1]
  } else{
    Rnew <- Rnew[-1,]
  }

  if (result[1]=="none") return(Rnew)
  else{
    if(result[1]=="html") print(xtable(Rnew), type="html")
    else print(xtable(Rnew), type="latex")
  }
}

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,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

Load in borrowing data, with pagel lexical replacement rate and the Winter et al. iconicity ratings. The data also includes the systematicity ratings from Monaghan et al. (2014). For posterity, we note the command line commands to create the data file:

```
cat iconicity_ratings.csv loanwords9.csv | gawk 'BEGIN{FS=",";OFS=","}
  {if(NR<=3002)i[$1]=$2;
  else {if(i[$1]>0)print $0,i[$1];else print $0,"#N/A"}}' > loanwords_withiconicity.csv
cat monaghan2014_systematicity.csv loanwords_withiconicity.csv | gawk 'BEGIN{FS=",";OFS=","}
  {if(NR<=2911)i[$1]=$NF;
    else {if(i[$1]>0)print $0,i[$1];
    else print $0,"#N/A"}}' > loanwords_withiconicity_systematicity.csv
```

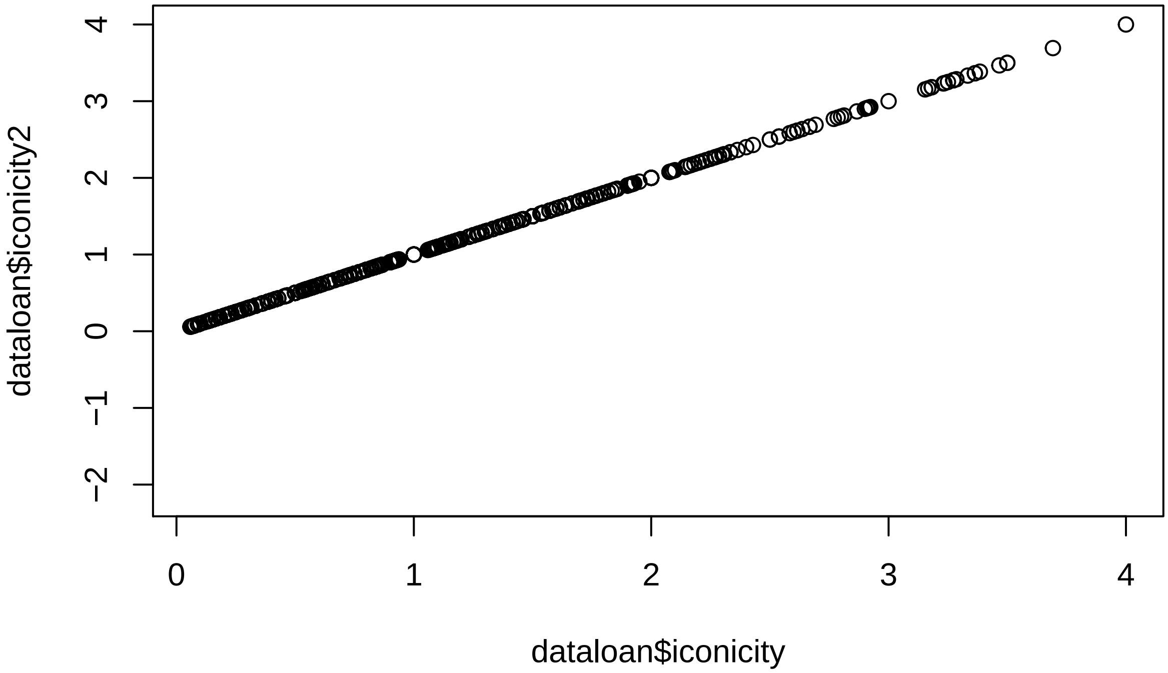
Load data and create key variables.

```
datafile = "../data/loanwords_withiconicity_systematicity.csv"
dataloan <- read.csv(datafile,stringsAsFactors = F)
dataloan$pagel_rate = as.numeric(dataloan$pagel_rate)
dataloan$iconicity = as.numeric(dataloan$iconicity)
```

```
dataloan$subtlexzipf = as.numeric(dataloan$subtlexzipf)
dataloan$AoA = as.numeric(dataloan$AoA)
dataloan$phonlength = as.numeric(dataloan$phonlength)
dataloan$systematicity = as.numeric(dataloan$systematicity)
dataloan$conc = as.numeric(dataloan$conc)
dataloan$cat = factor(dataloan$cat)
```

We want to use the iconcity data from Winter et al. (2017). However, the SI for Perry et al. (2018) have slightly more data. Here we show that the values are the same, and use the Perry et al. source which has more data:

```
i = read.csv("../data/iconicity_PerryEtAl2018.csv",stringsAsFactors = F)
dataloan$iconicity2 = i[match(dataloan$word,i$Word),]$Iconicity
plot(dataloan$iconicity,dataloan$iconicity2)
```



```
apply(dataloan[,c("iconicity",'iconicity2')],2,function(X){sum(!is.na(X))})
```

```
##  iconicity iconicity2
##        656        834
```

```
dataloan$iconicity = dataloan$iconicity2
```

Data preparation

Create binary borrowing variable:

```
dataloan$bor15 <- ifelse(dataloan$borrowing==1,1, ifelse(dataloan$borrowing==5,0,NA))
dataloan$bor15.cat <- factor(dataloan$bor15)
```

Percentage of missing data for each variable:

```
prop = apply(dataloan[,c("phonlength","AoA",
               "subtlexzipf", "cat", "iconicity", "systematicity",
               'conc','bor15')],2,
      function(X){sum(is.na(X))})/nrow(dataloan)
t(t(round(prop*100,2)))
```

```
##                [,1]
## phonlength     0.00
## AoA            1.81
## subtlexzipf    0.49
## cat            0.00
## iconicity     41.96
## systematicity 73.76
## conc           2.57
## bor15          5.29
```

There is too little data if we include only data with values for systematicity. So instead we omit systematicity:

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

This leaves 784 observations.

Make correlation table:

```
corstars(dataloan2[,c("subtlexzipf",
                          "AoA","phonlength",
                           'conc',
                          "iconicity")],
        removeTriangle = "lower")
```

```
##             AoA         phonlength  conc        iconicity
## subtlexzipf "-0.482***" "-0.335***" "-0.497***" "-0.050   "
## AoA         ""          " 0.243***" "-0.051   " "-0.039   "
## phonlength  ""          ""          " 0.097** " "-0.021   "
## conc        ""          ""          ""          "-0.038   "
## iconicity   ""          ""          ""          ""
```

Correlation table just for nouns:

```
corstars(dataloan2[dataloan2$cat=="Noun",
                   c("subtlexzipf",
                      "AoA","phonlength",
                      'conc',
                      "iconicity")],
         removeTriangle = "lower")
```

```
##             AoA         phonlength  conc        iconicity
## subtlexzipf "-0.501***" "-0.231***" "-0.208***" "-0.120** "
## AoA         ""          " 0.196***" "-0.230***" " 0.033   "
## phonlength  ""          ""          "-0.051   " " 0.009   "
## conc        ""          ""          ""          "-0.037   "
## iconicity   ""          ""          ""          ""
```

Scale variables

Center length by median:

```
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
```

Scale rest by mean:

```
dataloan2$AoAscale <- scale(dataloan2$AoA)
dataloan2$subtlexzipfscale <- scale(dataloan2$subtlexzipf)
dataloan2$concscale <- scale(dataloan2$conc)
dataloan2$iconscale <- scale(dataloan2$iconicity)
```

Replicating previous findings

This model replicates Perry et al. (2015) on the relationship between AoA and iconcity. The original model had the following output (not run here):

```
xmdl.AOA <- lm(KupermanAOA ~ Iconicity + Conc +
    POS + NMorph + ANC_LogFreq,
    data = icon)
summary(xmdl.AOA)

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     11.84456    0.25984  45.584  < 2e-16 ***
Iconicity       -0.26302    0.03590  -7.327 3.40e-13 ***
Conc            -1.04296    0.04278 -24.381  < 2e-16 ***
POSAdverb       -0.42920    0.29797  -1.440  0.14991
POSGrammatical  -0.55620    0.22957  -2.423  0.01549 *
POSInterjection -1.52095    0.55081  -2.761  0.00581 **
POSNoun          0.65116    0.11778   5.528 3.65e-08 ***
POSVerb         -0.13609    0.13520  -1.007  0.31426
NMorph           0.01238    0.10399   0.119  0.90528
ANC_LogFreq     -1.09390    0.04441 -24.631  < 2e-16 ***
```

We run the same model on our data:

```
perry = lm(AoAscale ~ iconscale + concscale +
    cat + phonlengthscale + subtlexzipfscale,
    data = dataloan2)
summary(perry)
```

```
##
## Call:
## lm(formula = AoAscale ~ iconscale + concscale + cat + phonlengthscale +
##     subtlexzipfscale, data = dataloan2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25382 -0.55155 -0.07502  0.44628  2.88520
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      -0.23142    0.08921  -2.594  0.00967 ** 
## iconscale        -0.06345    0.02979  -2.130  0.03349 *  
## concscale        -0.44278    0.03974 -11.141  < 2e-16 ***
## catAdverb         0.19925    0.21468   0.928  0.35362    
## catConjunction    0.20787    0.37673   0.552  0.58126    
## catDeterminer     0.61533    0.25466   2.416  0.01591 *  
## catInterjection  -0.69336    0.57618  -1.203  0.22920    
## catNot            0.25273    0.80502   0.314  0.75365    
## catNoun           0.30494    0.10485   2.908  0.00374 ** 
## catNumber         0.36468    0.29728   1.227  0.22030    
## catPreposition    0.47563    0.29771   1.598  0.11054    
## catPronoun        1.09721    0.30857   3.556  0.00040 ***
## catVerb           0.11473    0.10429   1.100  0.27162    
## phonlengthscale   0.05099    0.03086   1.652  0.09893 .  
## subtlexzipfscale -0.70222    0.03802 -18.469  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.7959 on 769 degrees of freedom
## Multiple R-squared:  0.3778, Adjusted R-squared:  0.3665 
## F-statistic: 33.35 on 14 and 769 DF,  p-value: < 2.2e-16
```

We note that:

- Both have a significant negative effect of iconcity.
- Both have a significant negative effect of concreteness.
- Neither have a significant effect of length.
- Both have a significant negative effect of frequency.
- There are differences between parts of speech, but the categories are different.

The results seem to replicate in broad terms.

Winter et al. (2017) predict iconicity by various measures, including concreteness, imagability and sensory experience. They find that concreteness is not directly correlated with iconicity (r = -0.008, p = 0.66). We calculated the raw correlation between iconicity and concreteness for each part of speech:

```
                r       p
Interjection    0.775   0.003
Verb            0.408   0.000
Grammatical   0.332 0.003
Adjective       0.160   0.000
Adverb        0.184 0.255
Noun           -0.024   0.332
Name           -0.001   0.999
```

```
res = data.frame()
for(p in c("Verb","Determiner","Pronoun","Preposition","Adverb","Adjective","Noun")){
   x = cor.test(dataloan2[dataloan2$cat==p,]$iconscale,
                dataloan2[dataloan2$cat==p,]$concscale)
   res = rbind(res, c(p,round(x$estimate,3),round(x$p.value,3)))
}
names(res)=c("POS","r","p")
res
```

```
##    POS     r    p
## 1 Verb 0.327    0
## 2 <NA>  <NA> <NA>
## 3 <NA>  <NA> <NA>
## 4 <NA>  <NA> <NA>
## 5 <NA>  <NA> <NA>
## 6 <NA>  <NA> <NA>
## 7 <NA>  <NA> <NA>
```

We note that there is a positive correlation for verbs, and no significant relationship for nouns in both datasets. The datasets disagree for adjectives, but our dataset has a smaller proportion of adjectives in comparison to Winter et al. (12% vs. 18%, see below).

For Winter et al's data, we show that concreteness does predict iconicity when controlling for frequency and part of speech:

```
summary(xmdl1 <- lm(Iconicity ~ Conc + LogFreq + POS, icon))
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.30786    0.10313  12.681  < 2e-16 ***
Conc             0.10906    0.02303   4.736 2.29e-06 ***
LogFreq         -0.20180    0.02260  -8.931  < 2e-16 ***
POSAdverb        0.07080    0.18061   0.392 0.695091
POSGrammatical  -0.10014    0.14345  -0.698 0.485169
POSInterjection  1.80180    0.31154   5.784 8.12e-09 ***
POSName         -0.89159    0.29941  -2.978 0.002928 **
POSNoun         -0.54445    0.05992  -9.087  < 2e-16 ***
POSVerb          0.23416    0.06809   3.439 0.000593 ***
```

But not when controling for frequency alone:

```
summary(xmdl1 <- lm(Iconicity ~ Conc + LogFreq, icon))
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.58949    0.10509  15.124   <2e-16 ***
Conc        -0.03883    0.02094  -1.854   0.0639 .
LogFreq     -0.20060    0.02096  -9.570   <2e-16 ***
```

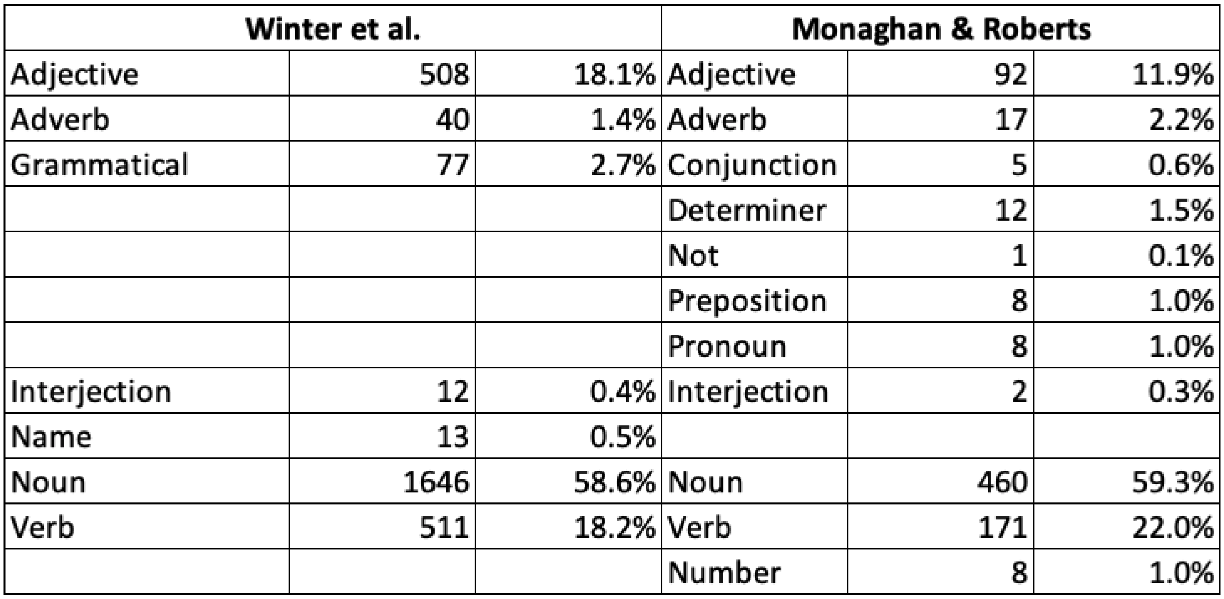
Similar model in our data:

```
winter = lm(iconscale ~concscale +subtlexzipfscale + cat ,
    data = dataloan2)
summary(winter)
```

```
##
## Call:
## lm(formula = iconscale ~ concscale + subtlexzipfscale + cat,
##     data = dataloan2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6815 -0.6251 -0.0556  0.5614  3.1688
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.46726    0.10630   4.396 1.26e-05 ***
## concscale          0.06866    0.04757   1.443  0.14934
## subtlexzipfscale -0.13523    0.04385  -3.084  0.00212 **
## catAdverb         -0.28777    0.25931  -1.110  0.26745
## catConjunction    -0.85424    0.45435  -1.880  0.06047 .
## catDeterminer     -0.39334    0.30731  -1.280  0.20095
## catInterjection   1.88435    0.69288   2.720  0.00668 **
## catNot            0.70486    0.97285   0.725  0.46896
## catNoun          -0.72062    0.12381  -5.821 8.60e-09 ***
## catNumber        -0.02331    0.35938  -0.065  0.94831
## catPreposition    0.04779    0.35973   0.133  0.89434
## catPronoun        0.34545    0.37192   0.929  0.35328
## catVerb          -0.16607    0.12567  -1.321  0.18674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9622 on 771 degrees of freedom
## Multiple R-squared:  0.08834,    Adjusted R-squared:  0.07415
## F-statistic: 6.226 on 12 and 771 DF,  p-value: 1.495e-10
```

Cocreteness is a significant predictor for Winter et al., but not in our data.

It seems that part of speech is a key factor in the results. Below we comapre the distribution of parts of speech in our data. We note that we're using more fine-grained categories, and some categories are less represented.



Winter et al.			**Monaghan & Roberts**		
Adjective	508	18.1%	Adjective	92	11.9%
Adverb	40	1.4%	Adverb	17	2.2%
Grammatical	77	2.7%	Conjunction	5	0.6%
			Determiner	12	1.5%
			Not	1	0.1%
			Preposition	8	1.0%
			Pronoun	8	1.0%
Interjection	12	0.4%	Interjection	2	0.3%
Name	13	0.5%			
Noun	1646	58.6%	Noun	460	59.3%
Verb	511	18.2%	Verb	171	22.0%
			Number	8	1.0%

Modelling

First, a simple linear model trying to predict the rate of lexical replacement from Pagel et al. (2007). The power is too low to detect effects:

```
a <- lm(pagel_rate ~
          subtlexzipf + AoA + phonlength +
          iconicity + systematicity + conc + cat,
        data = dataloan)
summary(a)
```

```
##
## Call:
## lm(formula = pagel_rate ~ subtlexzipf + AoA + phonlength + iconicity +
##     systematicity + conc + cat, data = dataloan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1757 -0.8946 -0.0779  0.8307  3.9234
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.7251     5.7515   0.995   0.3244
## subtlexzipf     -0.5312     0.6183  -0.859   0.3944
## AoA              0.5193     0.2868   1.811   0.0763 .
## phonlength      -0.1173     0.3025  -0.388   0.6999
## iconicity        0.1355     0.3334   0.407   0.6861
## systematicity 3586.8696  2529.4980   1.418   0.1625
## conc            -0.7168     0.7048  -1.017   0.3142
## catAdverb        0.1497     1.1310   0.132   0.8952
## catNoun          0.5651     0.9950   0.568   0.5727
## catNumber       -2.4284     1.6767  -1.448   0.1539
## catVerb          1.3393     0.6971   1.921   0.0605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.498 on 49 degrees of freedom
##   (1377 observations deleted due to missingness)
## Multiple R-squared:  0.4251, Adjusted R-squared:  0.3078
## F-statistic: 3.624 on 10 and 49 DF,  p-value: 0.001158
```

Next, a simple logisitic model predicting borrowing from several variables including systematicity. Given the amount of missing data (see above), this only includes 226 complete observations:

```r
b <- glm(bor15.cat ~ subtlexzipf + AoA + phonlength +
           iconicity + systematicity + conc + cat,
         data = dataloan, family=binomial)
summary(b)
```

```
##
## Call:
## glm(formula = bor15.cat ~ subtlexzipf + AoA + phonlength + iconicity +
##     systematicity + conc + cat, family = binomial, data = dataloan)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4705  -0.8487  -0.6242   1.0072   2.4829
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.7609     2.7793  -1.353  0.17599
## subtlexzipf       0.2623     0.3318   0.790  0.42928
## AoA               0.2551     0.1229   2.075  0.03795 *
## phonlength        0.4115     0.2527   1.628  0.10348
## iconicity        -0.6503     0.2044  -3.182  0.00146 **
## systematicity  1345.9103  1565.3148   0.860  0.38988
## conc             -0.2281     0.2685  -0.850  0.39550
## catAdverb       -16.3012   970.4996  -0.017  0.98660
## catDeterminer     0.4835     1.5344   0.315  0.75269
## catNoun           0.4201     0.5181   0.811  0.41745
## catNumber       -15.7714  1375.8424  -0.011  0.99085
## catVerb          -0.1392     0.5184  -0.269  0.78827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 274.74  on 225  degrees of freedom
## Residual deviance: 243.30  on 214  degrees of freedom
##   (1211 observations deleted due to missingness)
## AIC: 267.3
##
## Number of Fisher Scoring iterations: 15
```

Now we model the data with a GAM. We first replicate the GAM from Monaghan & Roberts (2019). This predicts borrowing from length, age of acquisition, frequency, concreteness, with random intercepts for part of speech, and random slopes for variables by part of speech. Results are similar on this sub-sample of data, though frequency is no longer significant:

```r
m0 = bam(bor15.cat ~
      s(subtlexzipfscale) +
      s(AoAscale) +
      s(phonlengthscale) +
      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(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
##     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)  -1.4141     0.4073  -3.472 0.000517 ***
## ---
## 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(subtlexzipfscale)     2.183e+00  2.799  4.614 0.203558
## s(AoAscale)             1.386e+00  1.684 16.194 0.000163 ***
## s(phonlengthscale)      2.611e+00  3.330 50.595 1.74e-10 ***
## s(concscale)            1.624e+00  2.018  1.301 0.519986
## s(cat)                  4.633e+00 10.000 22.131 3.27e-05 ***
## s(cat,phonlengthscale)  2.267e-05 10.000  0.000 0.384068
## s(cat,AoAscale)         4.080e-06 10.000  0.000 0.553318
## s(cat,subtlexzipfscale) 1.702e-01 10.000  0.184 0.312370
## s(cat,concscale)        9.554e-05 10.000  0.000 0.352356
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.183   Deviance explained =   16%
## fREML = 1112.4  Scale est. = 1         n = 784
```

Now the main GAM for the current paper, which takes the previous model and adds iconicity as a predictor:

```r
m1= bam(bor15.cat ~
      s(subtlexzipfscale) +
      s(AoAscale) +
      s(phonlengthscale) +
      s(concscale) +
      s(iconscale) +
      s(cat,bs='re')+
      s(cat,iconscale,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(m1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
##     s(concscale) + s(iconscale) + s(cat, bs = "re") + s(cat,
##     iconscale, 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)  -1.3474     0.3941  -3.419 0.000629 ***
## ---
## 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(subtlexzipfscale)     2.191e+00  2.813  3.645 0.277726
## s(AoAscale)             1.086e+00  1.166 15.340 0.000127 ***
## s(phonlengthscale)      2.409e+00  3.074 40.022  1.3e-08 ***
## s(concscale)            1.000e+00  1.000  0.174 0.676407
## s(iconscale)            1.761e+00  2.245 17.385 0.000284 ***
## s(cat)                  4.393e+00 10.000 17.666 0.000447 ***
## s(cat,iconscale)        3.161e-05 10.000  0.000 0.382085
## s(cat,phonlengthscale)  2.614e-01 10.000  0.318 0.287497
## s(cat,AoAscale)         5.396e-05 10.000  0.000 0.368722
## s(cat,subtlexzipfscale) 2.327e-01 10.000  0.259 0.303759
## s(cat,concscale)        8.752e-06 10.000  0.000 0.581252
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.203   Deviance explained = 17.6%
## fREML = 1118.8  Scale est. = 1         n = 784
```

Pretty results:

```
res = summary(m1)$s.table

res = round(res,3)
rownames(res)[rownames(res)=="s(subtlexzipfscale)"] = "Frequency"
rownames(res)[rownames(res)=="s(AoAscale)"] = "AoA"
rownames(res)[rownames(res)=="s(phonlengthscale)"] = "Length"
rownames(res)[rownames(res)=="s(concscale)"] = "Concreteness"
rownames(res)[rownames(res)=="s(iconscale)"] = "Iconicity"
rownames(res)[rownames(res)=="s(cat)"] = "Grammatical category"
res[,4][res[,4]==0] = "<0.001"
res[,4] = gsub("0\\.",".",res[,4])
res
```

```
##                         edf     Ref.df  Chi.sq   p-value
## Frequency               "2.191" "2.813" "3.645"  ".278"
## AoA                     "1.086" "1.166" "15.34"  "<.001"
## Length                  "2.409" "3.074" "40.022" "<.001"
## Concreteness            "1"     "1"     "0.174"  ".676"
## Iconicity               "1.761" "2.245" "17.385" "<.001"
## Grammatical category    "4.393" "10"    "17.666" "<.001"
## s(cat,iconscale)        "0"     "10"    "0"      ".382"
## s(cat,phonlengthscale)  "0.261" "10"    "0.318"  ".287"
## s(cat,AoAscale)         "0"     "10"    "0"      ".369"
## s(cat,subtlexzipfscale) "0.233" "10"    "0.259"  ".304"
## s(cat,concscale)        "0"     "10"    "0"      ".581"
```

```
write.csv(res,file="../results/GAM_results.csv")
```

Test only iconicity to show that it is an independent predictor:

```
m2= bam(bor15.cat ~
          s(iconscale),
        data = dataloan2,
        family='binomial')
summary(m2)
```

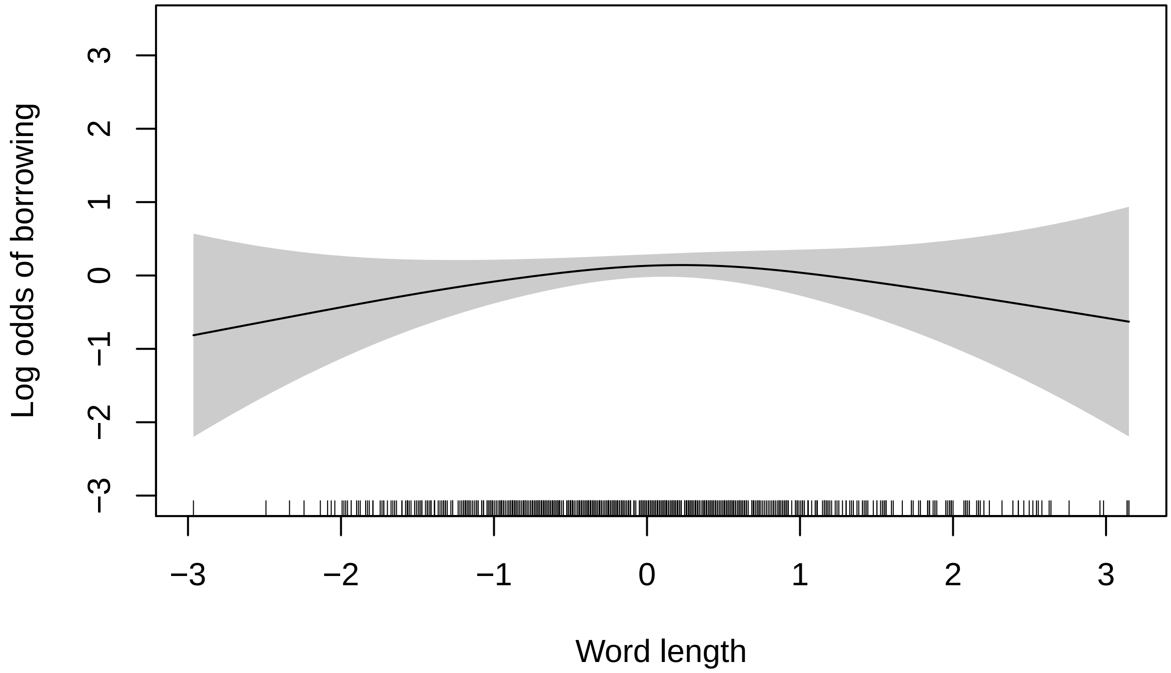
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(iconscale)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.51974    0.07531  -6.902 5.14e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq  p-value
## s(iconscale) 1.194  1.367  22.09 7.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## R-sq.(adj) =  0.0291   Deviance explained = 2.47%
## fREML = 1114.3  Scale est. = 1         n = 784
```

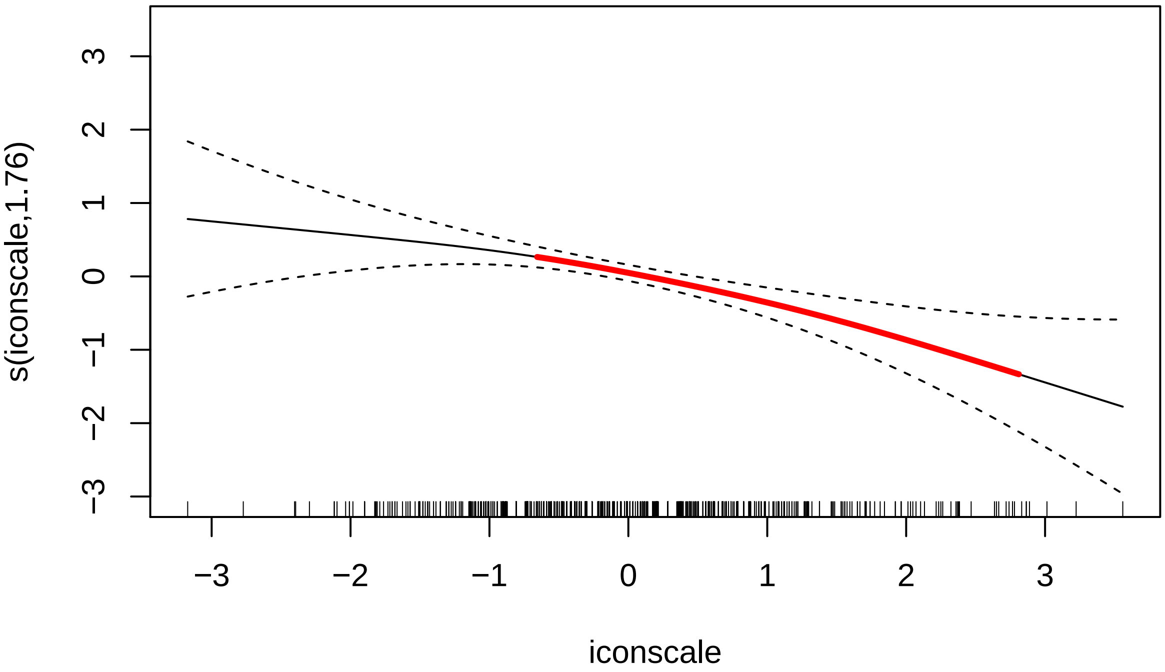
Visualise effects

Some fo the plots below highlight which sections of the GAM splines are significantly increasing or decreasing. This method comes from this source. 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.

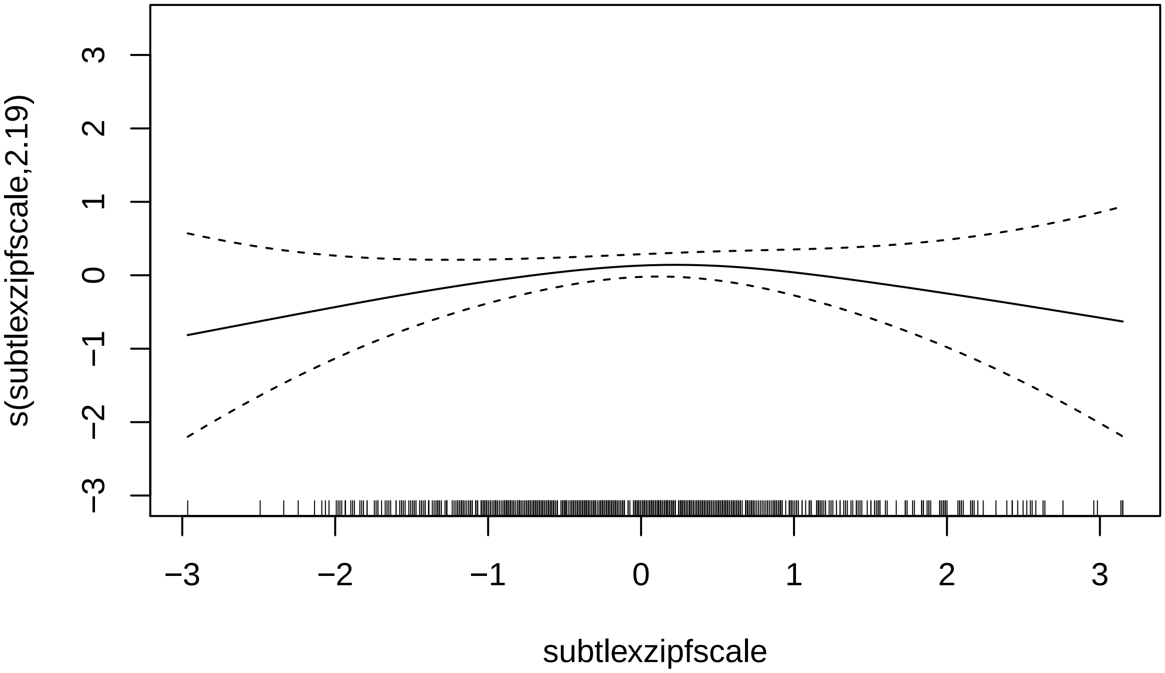
```
px = plot.gam(m1,select=1, xlab="Word length", ylab="Log odds of borrowing",shade = T)
```



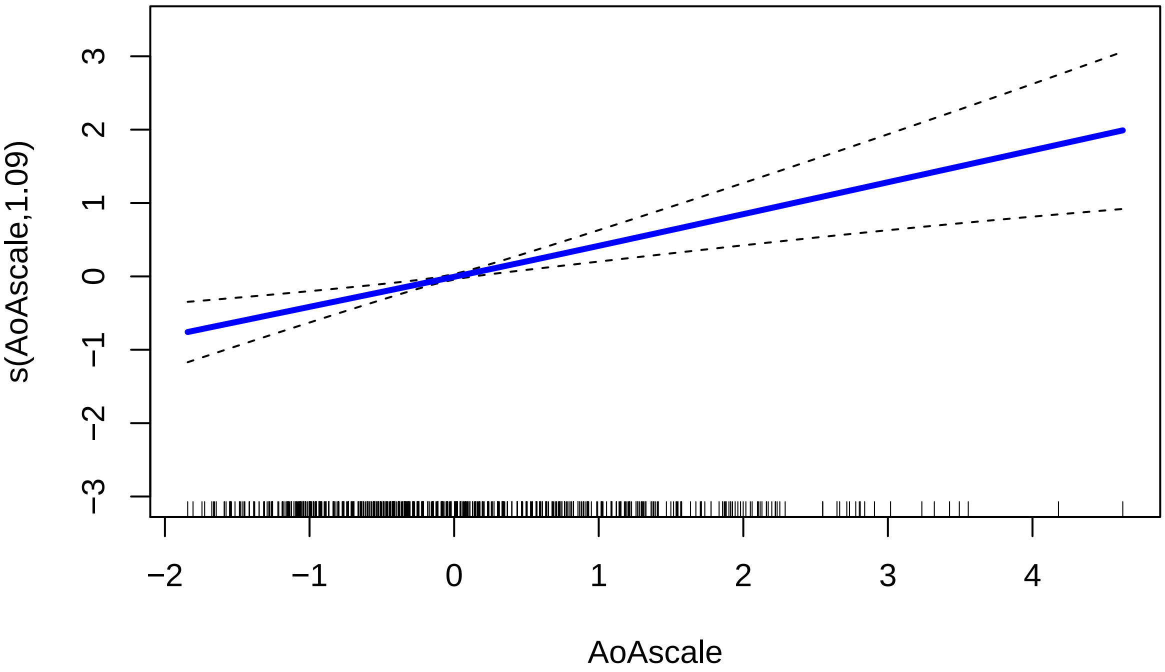
```
pfreq = rescaleGam(px,1,dataloan2$subtlexzipfscale, "Frequency",
                   xlim = c(2,8), breaks=c(2,4,6,8,10,12))
paoa = rescaleGam(px,2,dataloan2$AoAscale, "Age of acquisition",
                  xlim=c(2,13),breaks=c(2,4,6,8,10,12,14))
plen = rescaleGam(px,3,dataloan2$phonlengthscale, "Length",
                  xlim=c(2,10),breaks=c(2,4,6,8,10))
pconc = rescaleGam(px,4,dataloan2$concscale, "Concreteness",
                   xlim = c(1,5), breaks=1:5)
picon = rescaleGam(px,5,dataloan2$iconscale, "Iconicity",
                   xlim = c(0,4), breaks=c(0,1,2,3,4))

pSigIcon = plotGAMSignificantSlopes(m1,"iconscale","Iconicity")
```



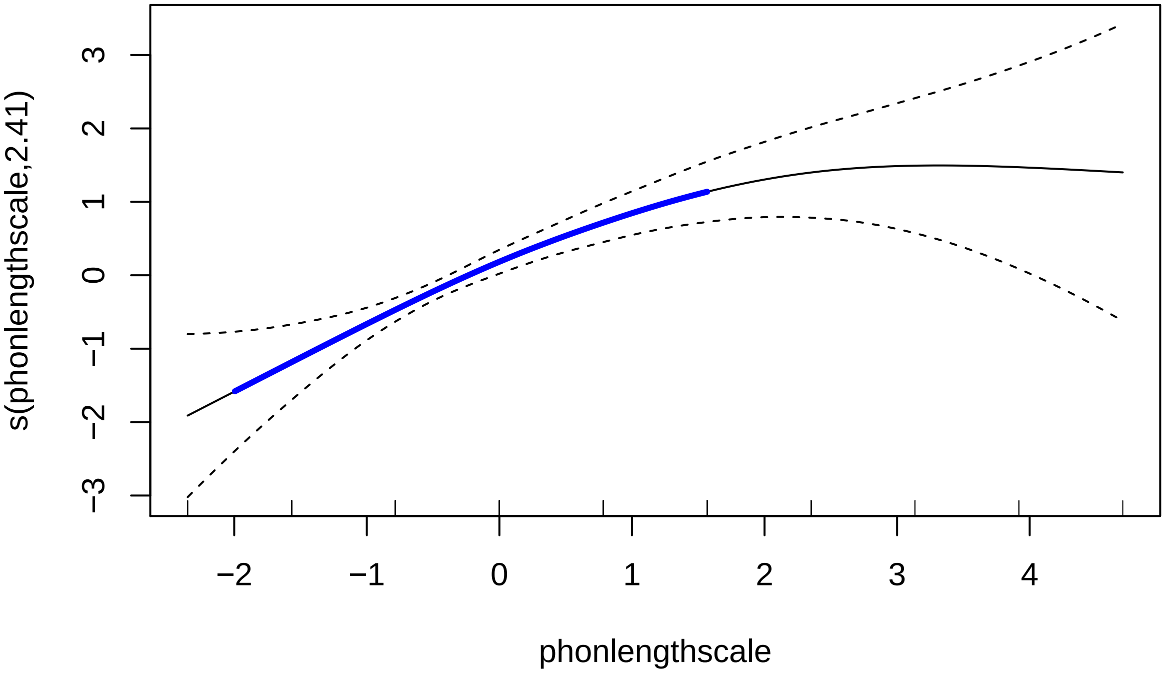
```
pSigFreq = plotGAMSignificantSlopes(m1,"subtlexzipfscale","Frequency")
```



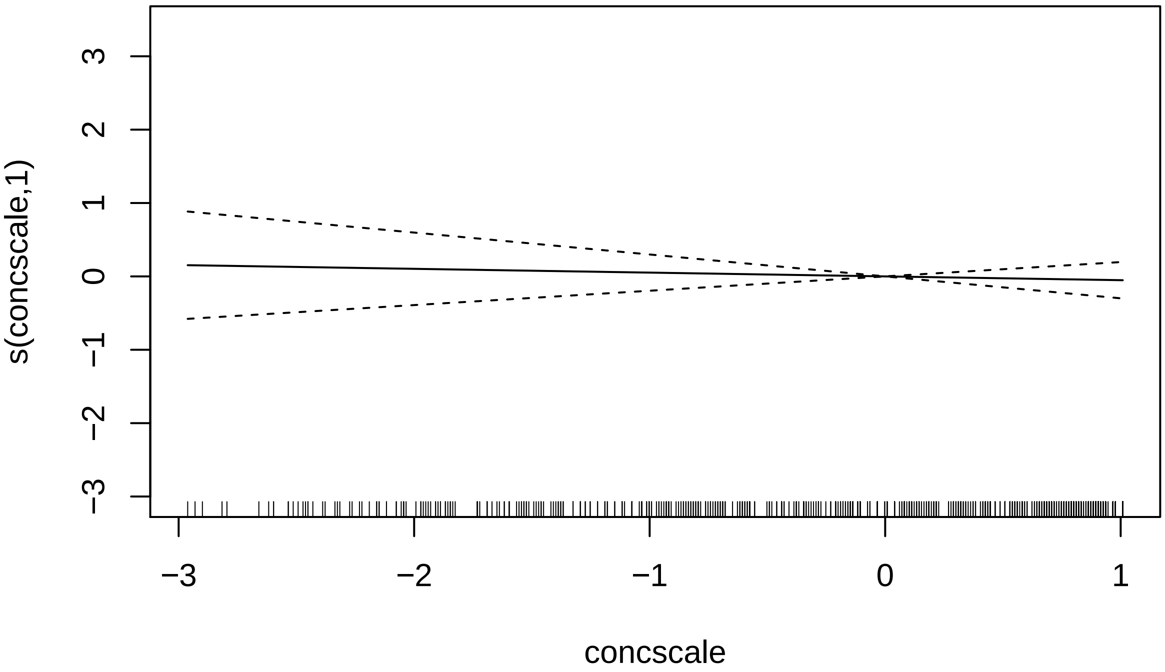
```
pSigAoA = plotGAMSignificantSlopes(m1,"AoAscale","AoA")
```



```
pSigLen = plotGAMSignificantSlopes(m1,"phonlengthscale","Word Length")
```

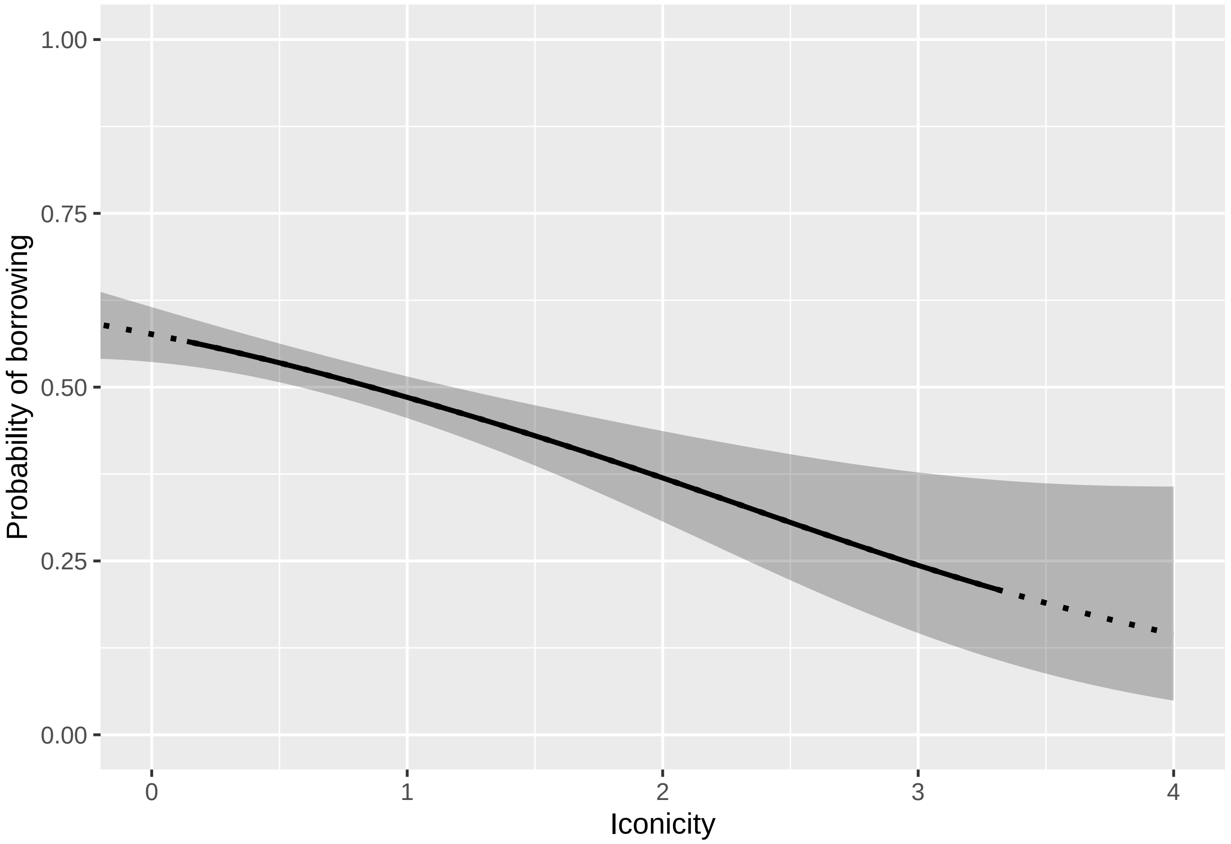


```
pSigConc = plotGAMSignificantSlopes(m1,"concscale","Concreteness")
```

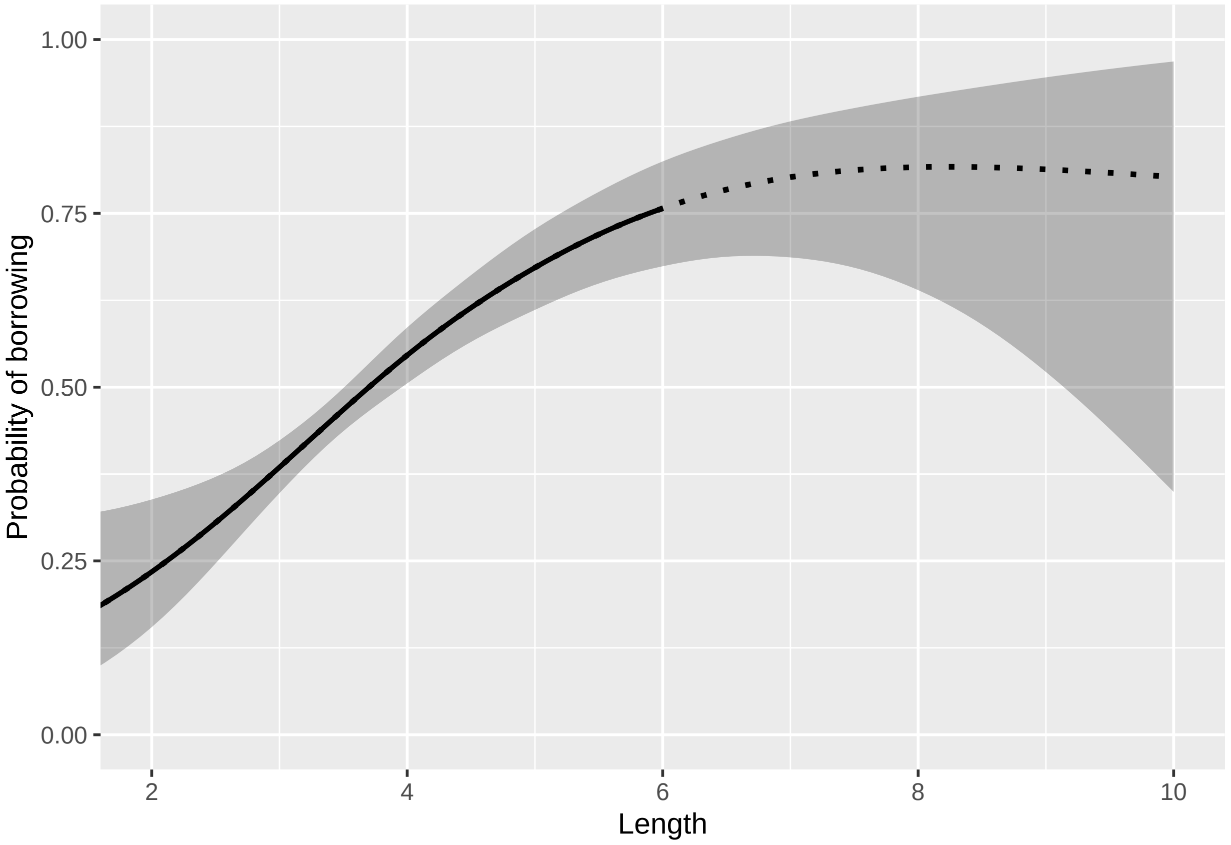


Plot increasing/decreasing curves onto the paoa,pfreq,picon,plen figures above.

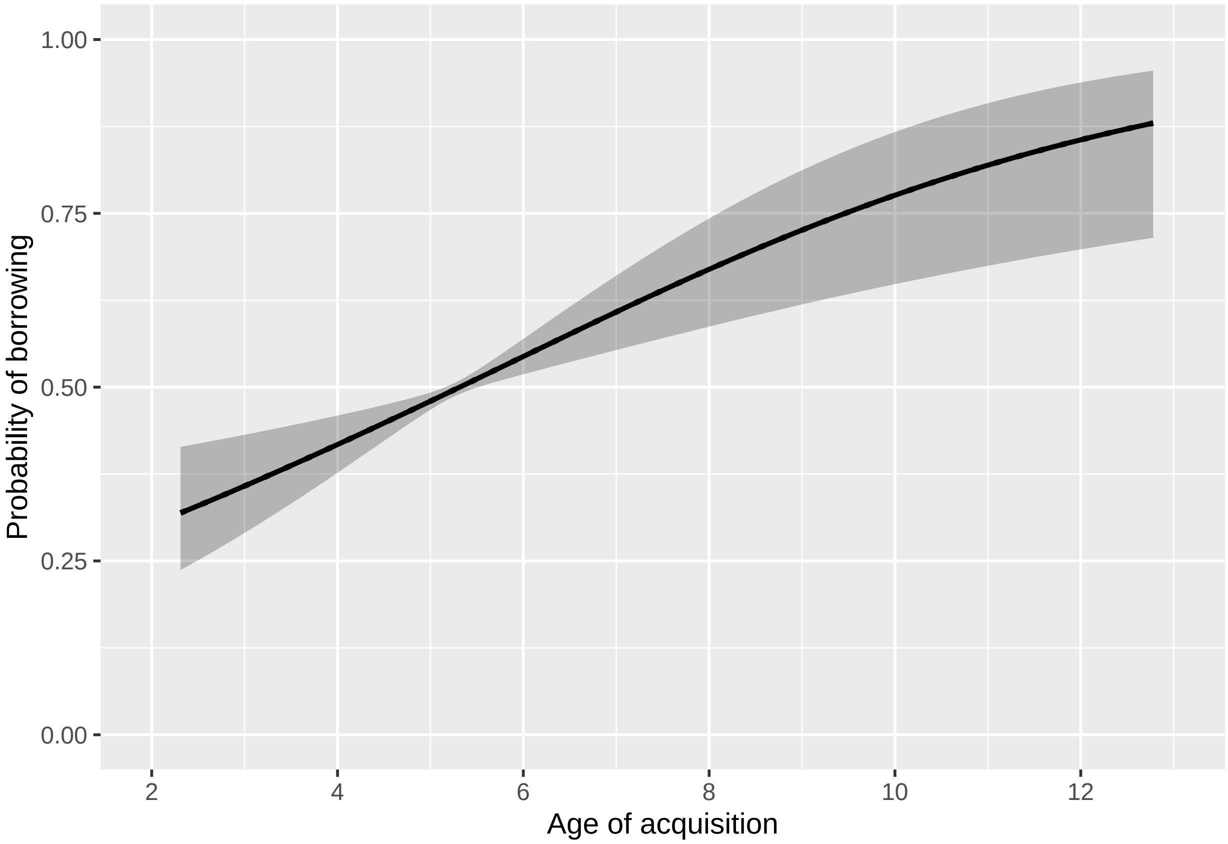
```
rescaleDerivitiesPlot = function(origPlot,sigCurveData){
  sigCurveData$curve.x = origPlot$data$x
  sigCurveData$m2.dsig.incr = logit2per(sigCurveData$m2.dsig.incr)
  sigCurveData$m2.dsig.decr = logit2per(sigCurveData$m2.dsig.decr)

  ret = origPlot +
    geom_path(data = sigCurveData,
              aes(x = curve.x,
                  y = m2.dsig.incr),
              size=0.9) +
    geom_path(data = sigCurveData,
              aes(x = curve.x,
                  y = m2.dsig.decr),
              size=0.9) +
    theme(legend.position = 'none')
  return(ret)
}

pSigIcon2 = rescaleDerivitiesPlot(picon,pSigIcon)
pSigLen2 = rescaleDerivitiesPlot(plen,pSigLen)
pSigAoA2 = rescaleDerivitiesPlot(paoa,pSigAoA)
pSigFreq2 = rescaleDerivitiesPlot(pfreq,pSigFreq)
pSigConc2 = rescaleDerivitiesPlot(pconc,pSigConc)

pSigIcon2
```


1.00
0.75
0.50
0.25
0.00
Probability of borrowing
0
1
2
3
4
Iconicity

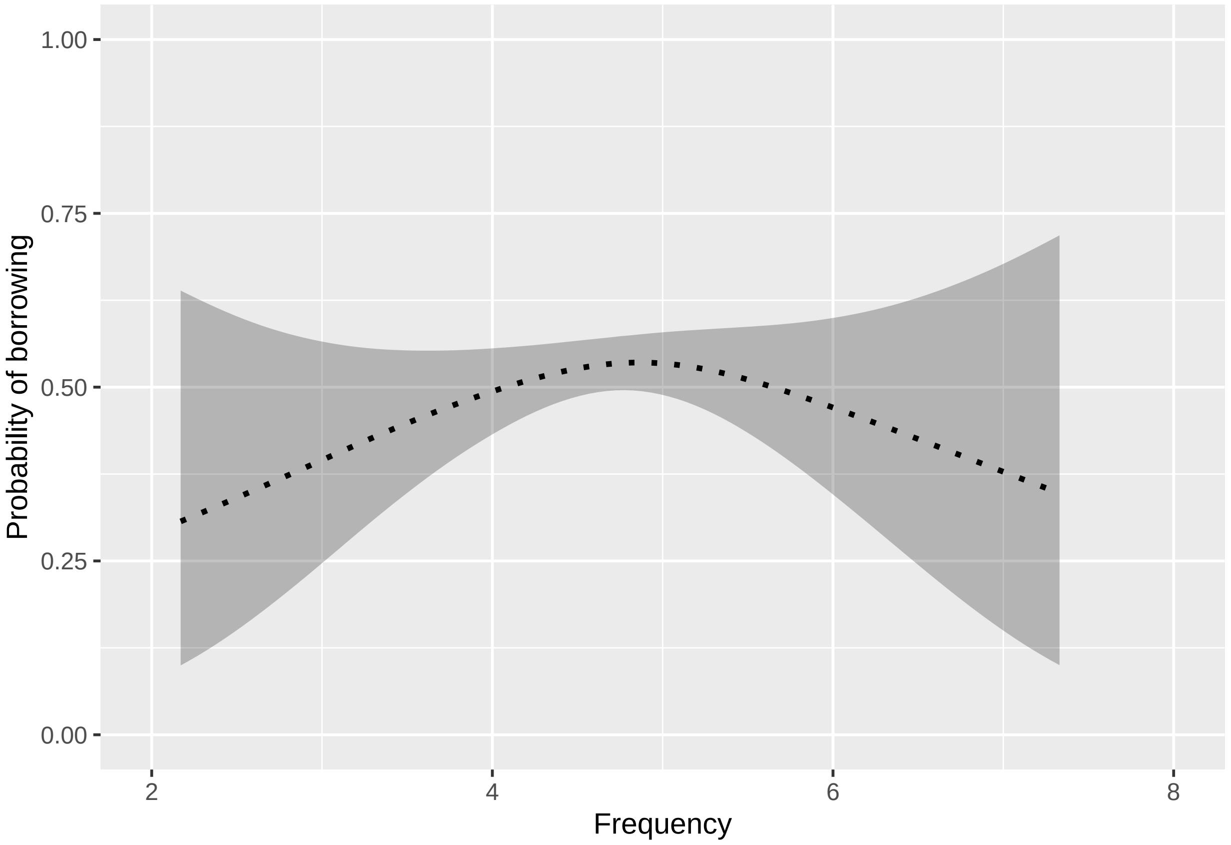
```
pSigLen2
```



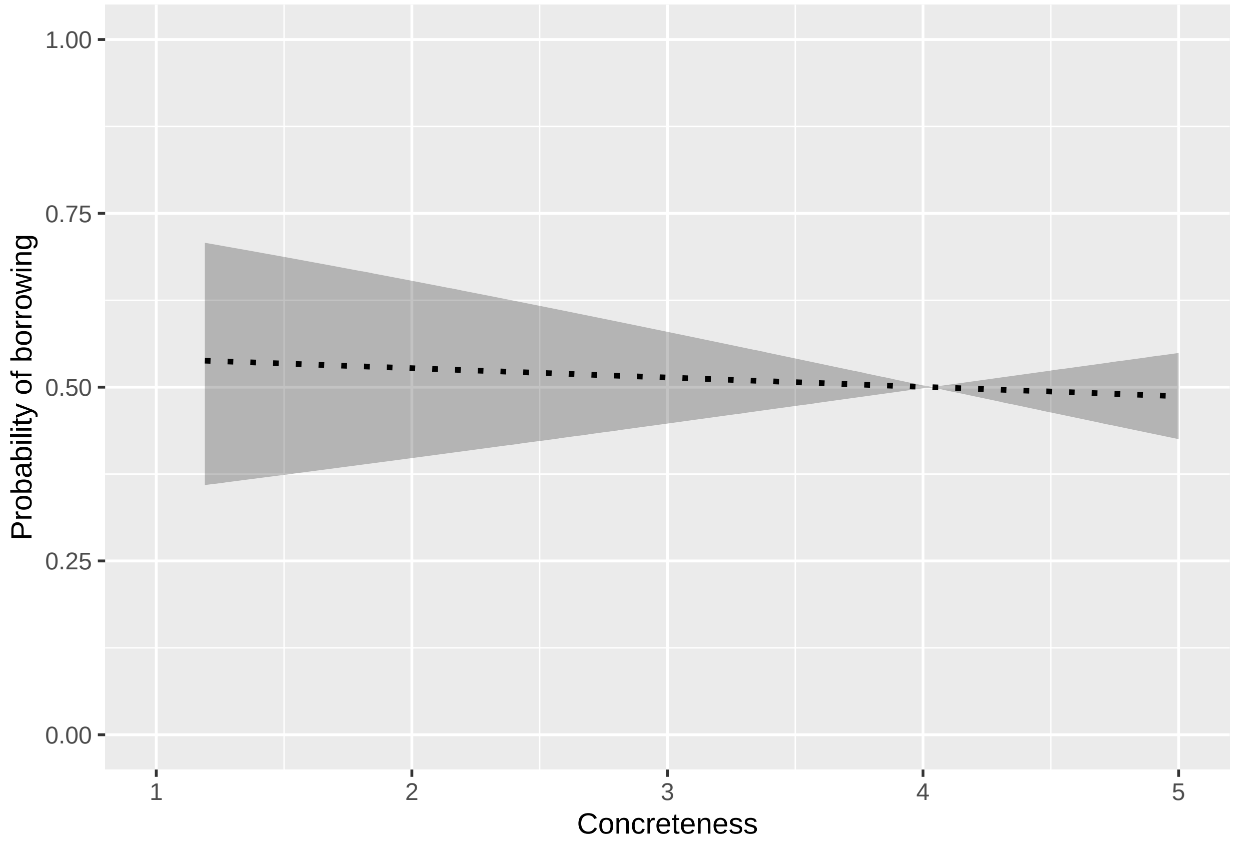
```
pSigAoA2
```


1.00
0.75
0.50
0.25
0.00
Probability of borrowing
2
4
6
8
10
12
Age of acquisition

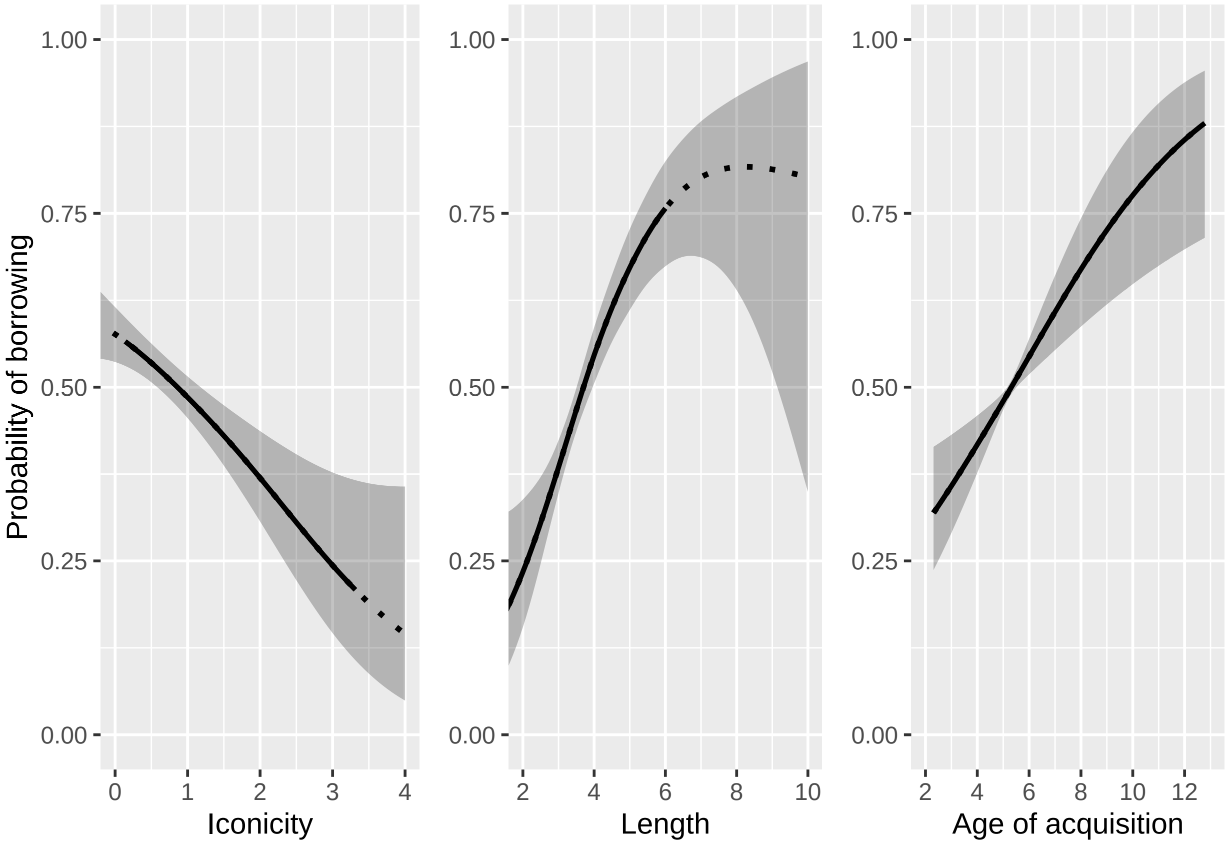
```
pSigFreq2
```


1.00
0.75
0.50
0.25
0.00
Probability of borrowing
2
4
6
8
Frequency

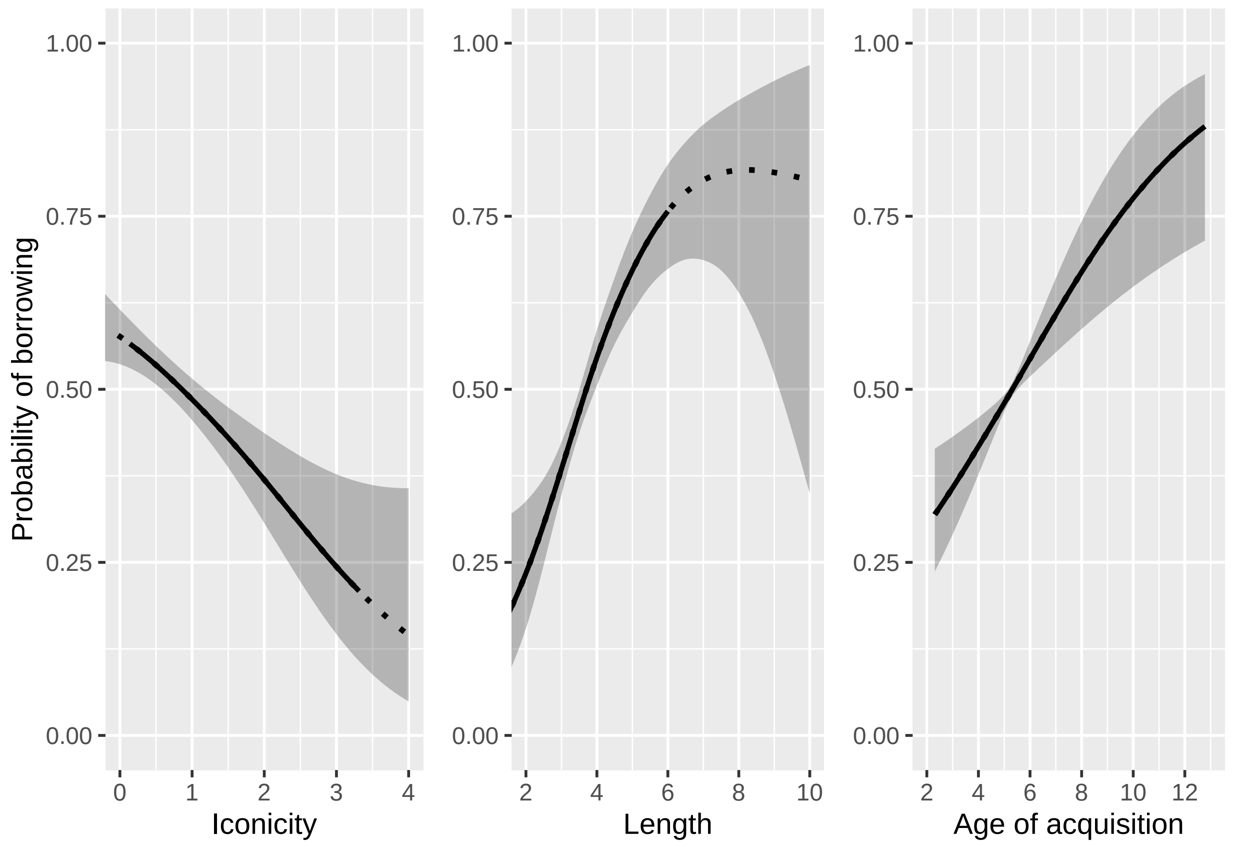
```
pSigConc2
```



```
# All together:
gx = grid.arrange(pSigIcon2,
             pSigLen2 + theme(axis.title.y = element_blank()),
             pSigAoA2 + theme(axis.title.y = element_blank()),
             layout_matrix=matrix(1:3,nrow = 1),
             widths=c(1.1,1,1))
```


Probability of borrowing
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
Iconicity
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
Length
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
Age of acquisition

```
plot(gx)
```



Output to file:

```
pdf("../results/Results_Iconicity.pdf",width=3,height=3)
  pSigIcon2
dev.off()
```

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

```
pdf("../results/Results.pdf", width=6,height=3)
  plot(gx)
dev.off()
```

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

Variance explained

The method here follows Wood's advice (https://stat.ethz.ch/pipermail/r-help/2007-October/142811.html). It compares a model with and without the key variable, keeping the smoothing parameters the same for smooth terms that remain in each model so that extra variance isn't accounted for in the adjustment of the more minimal model.

```
m1.min = gam(bor15.cat ~1,data = dataloan2, family='binomial')
rsq= c()
dev = c()
sTerms = c("s(phonlengthscale)","s(AoAscale)","s(subtlexzipfscale)",
           "s(iconscale)", "s(concscale)","s(cat)")
for(sx in sTerms){
  sxF = formula(paste("~.-",sx))
  m1Adj = update(m1, sxF, sp=m1$sp[names(m1$sp!=sx)])
  rsq = c(rsq,summary(m1)$r.sq - summary(m1Adj)$r.sq)

  dev = c(dev, (deviance(m1Adj)-deviance(m1))/deviance(m1.min))

}
names(rsq) = sTerms
names(dev) = sTerms
```

Percentage of variance explained:

```
t(t(round(100*rsq,3)))
```

```
##                        [,1]
## s(phonlengthscale)  7.434
## s(AoAscale)         1.710
## s(subtlexzipfscale) 1.280
## s(iconscale)        0.441
## s(concscale)        0.185
## s(cat)              0.000
```

Percentage of deviance explained:

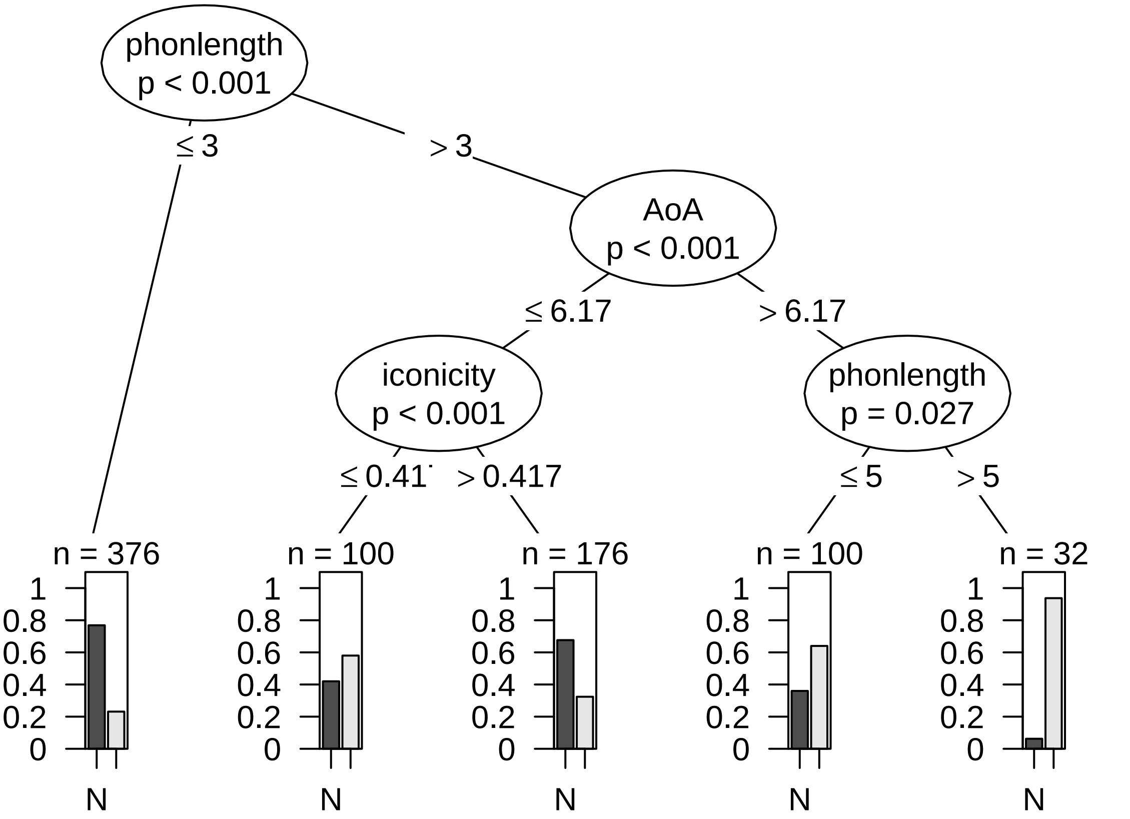
```
t(t(round(100*dev,3)))
```

```
##                        [,1]
## s(phonlengthscale)  5.483
## s(AoAscale)         1.402
## s(subtlexzipfscale) 1.005
## s(iconscale)        0.286
## s(concscale)        0.078
## s(cat)              0.000
```

Decision tree

Use a decision tree to look at interactions and importance. There's no need to use scaled variables with decision trees.

```
set.seed(3289)
t = ctree(factor(bor15,levels=c(0,1),labels=c("N","Y"))~
                    phonlength + AoA +
                    subtlexzipf + conc +
                    iconicity + cat,
          data = dataloan2)
plot(t,terminal_panel = node_barplot(t,beside=T,id=F),
     inner_panel = node_inner(t,id=F))
```



```
# output to file
pdf("../results/ResultsDecisionTree.pdf",width=7,height=5)
plot(t,terminal_panel = node_barplot(t,beside=T,id=F),
     inner_panel = node_inner(t,id=F))
dev.off()
```

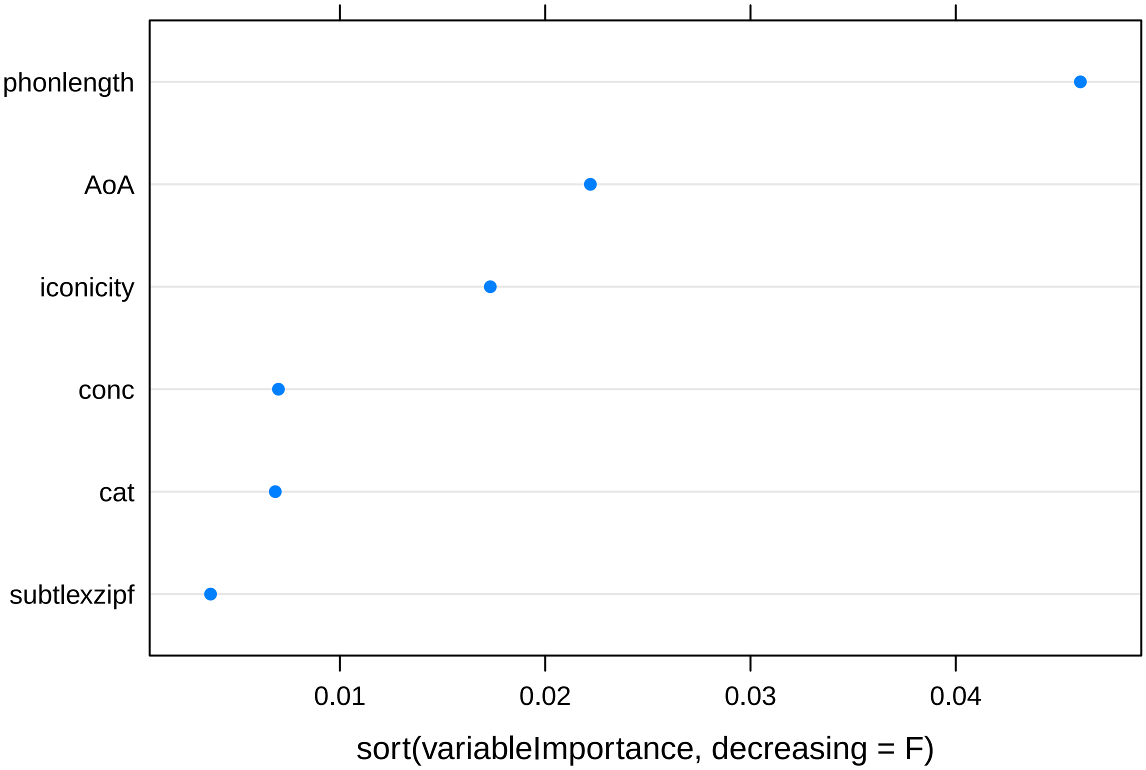
```
## pdf
##   2
```

The decision tree above suggests that the biggest effect of iconicity is for longer words learned later in life. Next, we use random forests to estimate relative importance of measures:

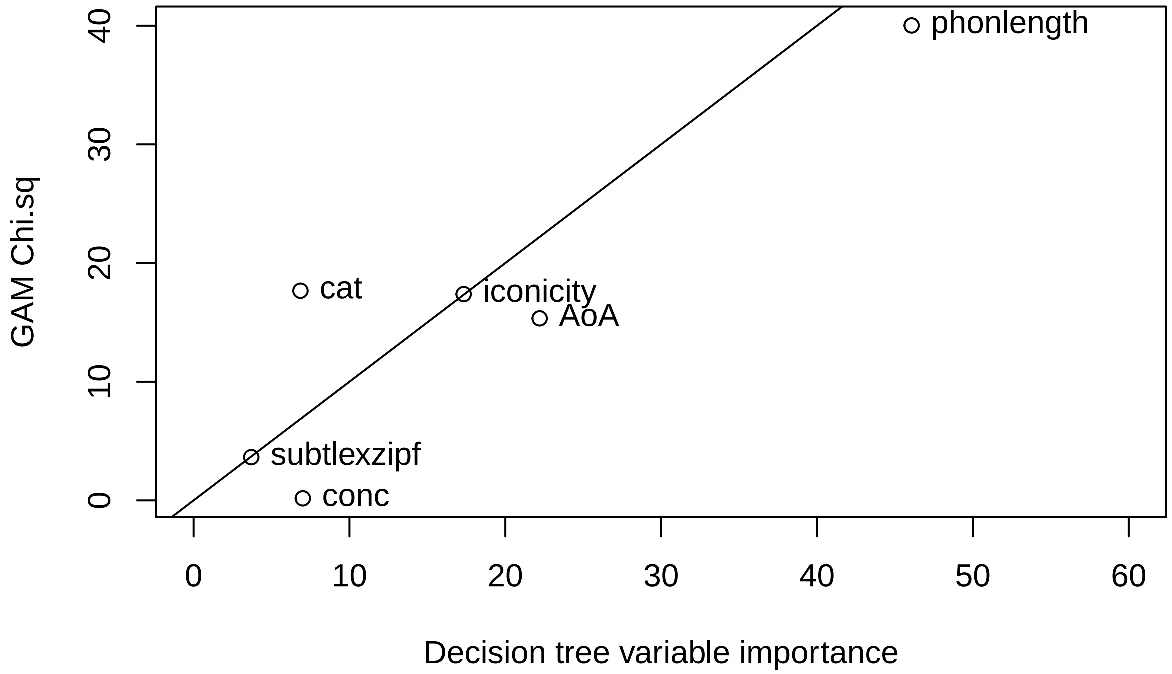
```
set.seed(3289)
f = cforest(bor15.cat~phonlength + AoA +
              subtlexzipf + conc +
              iconicity + cat,
          data = dataloan2)
variableImportance = varimp(f)
variableImportance
```

```
##   phonlength          AoA subtlexzipf        conc   iconicity         cat
## 0.046069444 0.022201389 0.003701389 0.007006944 0.017326389 0.006854167
```

```
dotplot(sort(variableImportance,decreasing = F))
```



```
ctreeOrder = c("subtlexzipf","AoA","phonlength","conc","iconicity","cat")
plot(variableImportance[ctreeOrder]*1000,
     summary(m1)$s.table[1:length(ctreeOrder),3],
     xlab="Decision tree variable importance",
     ylab="GAM Chi.sq",
     xlim=c(0,60))
text(variableImportance[ctreeOrder]*1000,summary(m1)$s.table[1:length(ctreeOrder),3],ctreeOrder,pos=4)
abline(0,1)
```



The importance values agree with the Chi.sq values from the GAM (r = 0.9. The effect of iconicity looks like it applies after the effects of length and AoA.

Distinguish borrowing type information

There are different kinds of borrowing scenario, including a word replacing an existing word, coexisting with an existing word or being inserted into the language (e.g. for a new concept or artefact that doesn't yet have a name). Do the psycholinguistic predictors relate differently to each of these types?

There are a few ways of testing the relationships between them. The first is to check whether the correlations are robust when running the model only on words that belong to just one category of borrowing (vs. not borrowed).

The analyses below show broad consistencies:

- For all types, lower iconicity is associated with greater probability of borrowing, though this was not significant for replacements (p = 0.5).
- For all types, longer words are associated with greater probaility of borrowing.
- For replacements, words learned later in life are associated with grater probability of borrowing. The relationship with AoA is similar but not significant for coexisting words (p = 0.12) or insertions (p = 0.13).

The second approach is to try to tell the difference between the different borrowing types. The main models above try to estimate whether a word is borrowed or not based on psycholinguistic predictors (frequency, iconicity etc.). The model below tries to predict which fine-grained category a word belongs to (not borrowed, replaced, coexisting with an existing word, or inserted), based on the psycholinguistic predictors. This requires a multinomial GAM. We use non-borrowed words as the intercept condition. The model estimates smooth curve parameters for each category. A significant factor indicates that the given category (e.g. replacements) has a significanlty different relation with a given factor (e.g. age of acquisition) compared to the intercept condition.

The analyses below show:

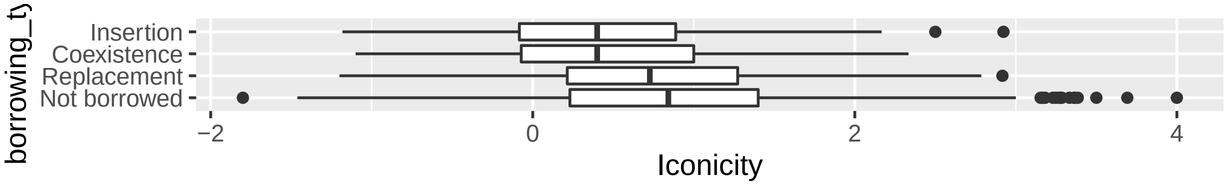
- Length predicts likelihood of borrowing for all three types.
- AoA predicts likelihood of borrowing for all three types.
- Iconicity predicts likelihood of borrowing for insertions and coexisting words, but not significantly for replacements.

The only interesting differences between the three borrowing types are:

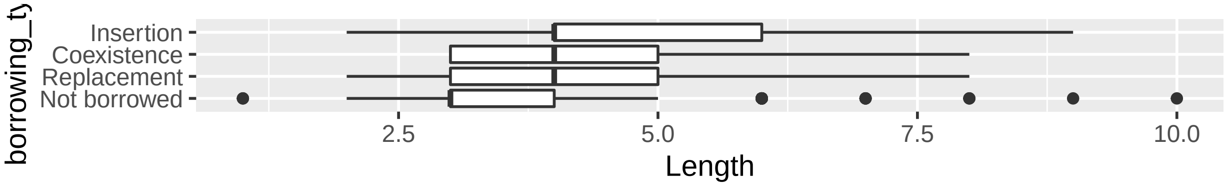
- High concreteness is a predictor of insertion (but not other types of borrowing). This makes sense: many of the words that did not exist before the borrowing are technological innovations.
- Iconicity is not a predictor for words that replace others.

Below are the distributions of various psycholinguistic predictors by different borrowing types:

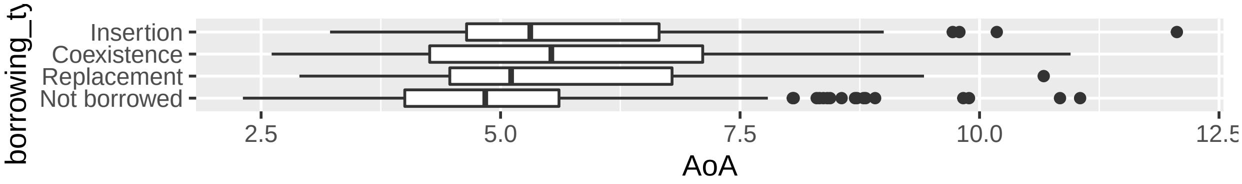
```
dataloan2$borrowing_type = factor(dataloan2$borrowing_type,
          levels=c("Not borrowed","Replacement","Coexistence","Insertion"))
ggplot(dataloan2[!is.na(dataloan2$borrowing_type),],aes(x=borrowing_type,y=iconicity)) +
  geom_boxplot()+ylab("Iconicity")+coord_flip()
```



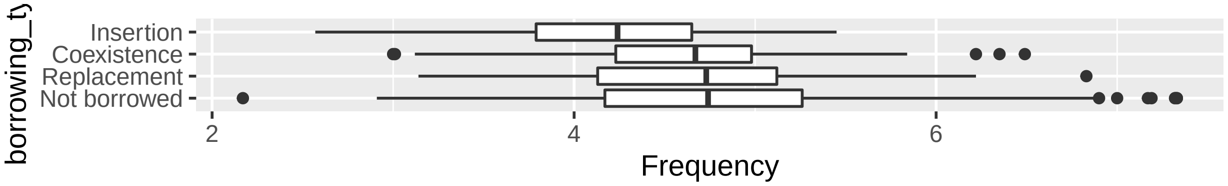
```
ggplot(dataloan2[!is.na(dataloan2$borrowing_type),], aes(x=borrowing_type,y=phonlength))+
  geom_boxplot()+ylab("Length")+coord_flip()
```



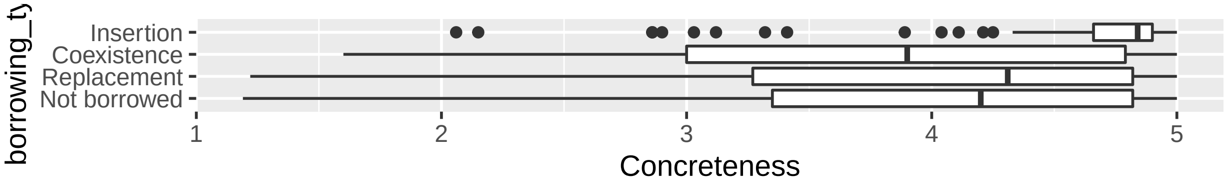
```
ggplot(dataloan2[!is.na(dataloan2$borrowing_type),],aes(x=borrowing_type,y=AoA))+
  geom_boxplot()+coord_flip()
```



```
ggplot(dataloan2[!is.na(dataloan2$borrowing_type),],aes(x=borrowing_type,y=subtlexzipf))+
  geom_boxplot()+ylab("Frequency")+coord_flip()
```



```
ggplot(dataloan2[!is.na(dataloan2$borrowing_type),],aes(x=borrowing_type,y=conc))+
  geom_boxplot()+ylab("Concreteness")+coord_flip()
```



Models for each borrowing type

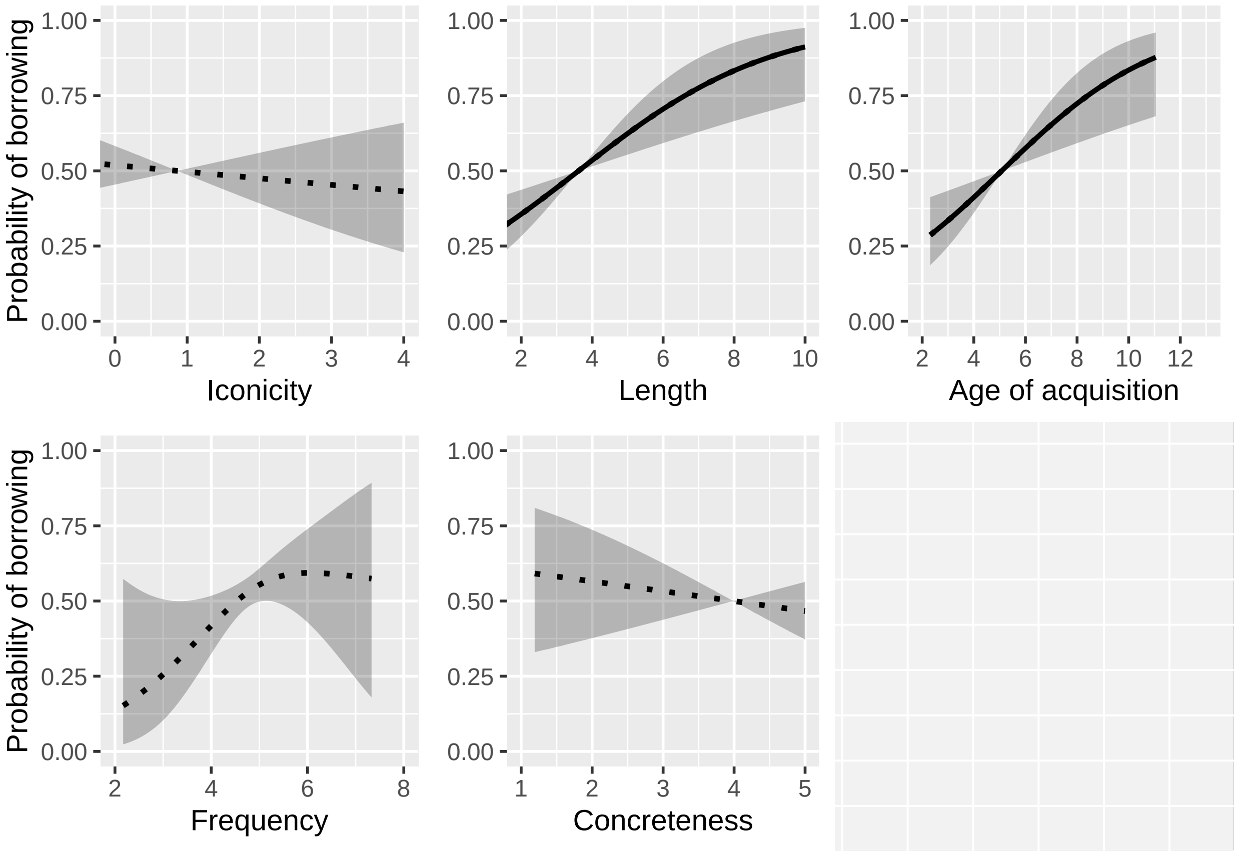
Replacements: Repeat GAM for only those words that are classified as Replacements vs. not borrowed:

```
m1r= update(m1, data = dataloan2[dataloan2$borrowing_type %in% c("Not borrowed","Replacement"),])
cat(paste(capture.output(summary(m1r))[18:32],collapse="\n"))
```

```
## Approximate significance of smooth terms:
##                                  edf Ref.df Chi.sq  p-value
## s(subtlexzipfscale)      1.871e+00  2.381  4.829 0.129084
## s(AoAscale)              1.000e+00  1.000 10.637 0.001109 **
## s(phonlengthscale)       1.000e+00  1.000 12.203 0.000477 ***
## s(concscale)             1.000e+00  1.000  0.470 0.492779
## s(iconscale)             1.000e+00  1.000  0.342 0.558596
## s(cat)                   3.157e+00 10.000 10.366 0.005327 **
## s(cat,iconscale)         4.210e-02 10.000  0.043 0.342077
## s(cat,phonlengthscale)   1.537e-06 10.000  0.000 1.000000
## s(cat,AoAscale)          4.377e-04 10.000  0.000 0.343354
## s(cat,subtlexzipfscale)  1.835e-06 10.000  0.000 0.937030
## s(cat,concscale)         3.504e-06 10.000  0.000 0.566372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Visualise effects (duplicated visualisation code is hidden, see the RMD file):

```
plot(m1r.vis)
```



Coexistence: Repeat GAM only for coexistence words vs words that are not borrowed:

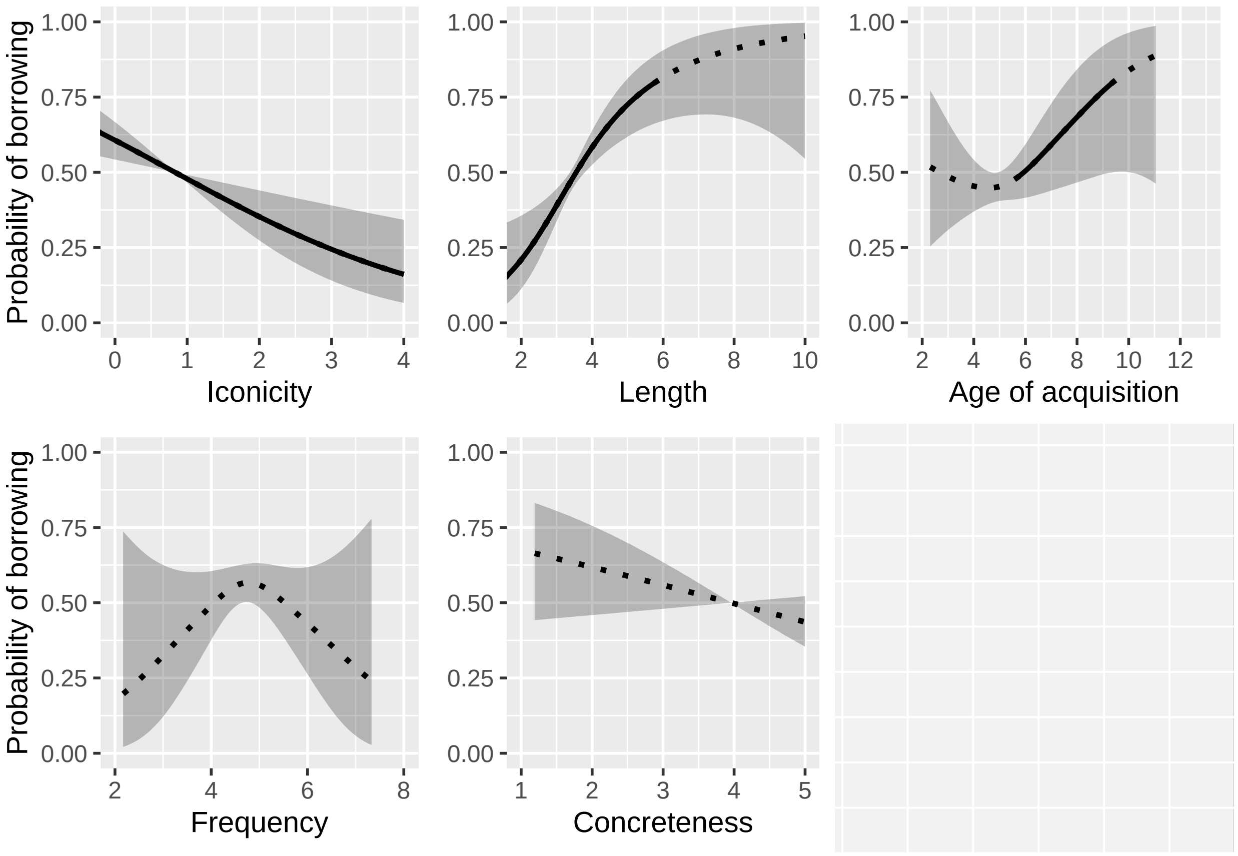
```
m1c= update(m1, data = dataloan2[
  dataloan2$borrowing_type %in% c("Not borrowed","Coexistence"),])
cat(paste(capture.output(summary(m1c))[18:32],collapse="\n"))
```

```
## Approximate significance of smooth terms:
##                                edf Ref.df Chi.sq  p-value    
## s(subtlexzipfscale)      2.422e+00  3.110  4.476 0.229039    
## s(AoAscale)              2.287e+00  2.893  5.295 0.115062    
## s(phonlengthscale)       1.861e+00  2.362 17.224 0.000388 ***
## s(concscale)             1.000e+00  1.000  2.228 0.135547    
## s(iconscale)             1.000e+00  1.000 10.963 0.000930 ***
## s(cat)                   2.233e-06 10.000  0.000 0.580392    
## s(cat,iconscale)         1.228e-05 10.000  0.000 0.411613    
## s(cat,phonlengthscale)   1.022e+00 10.000  1.693 0.164013    
## s(cat,AoAscale)          1.204e+00 10.000  3.452 0.068835 .  
## s(cat,subtlexzipfscale)  7.512e-01 10.000  0.971 0.283084    
## s(cat,concscale)         1.046e-05 10.000  0.000 0.404642    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m1c.vis = visGam(m1c)
```

Visualise effects:

```
plot(m1c.vis)
```



Insertion: Repeat GAM only for insertions vs words that are not borrowed:

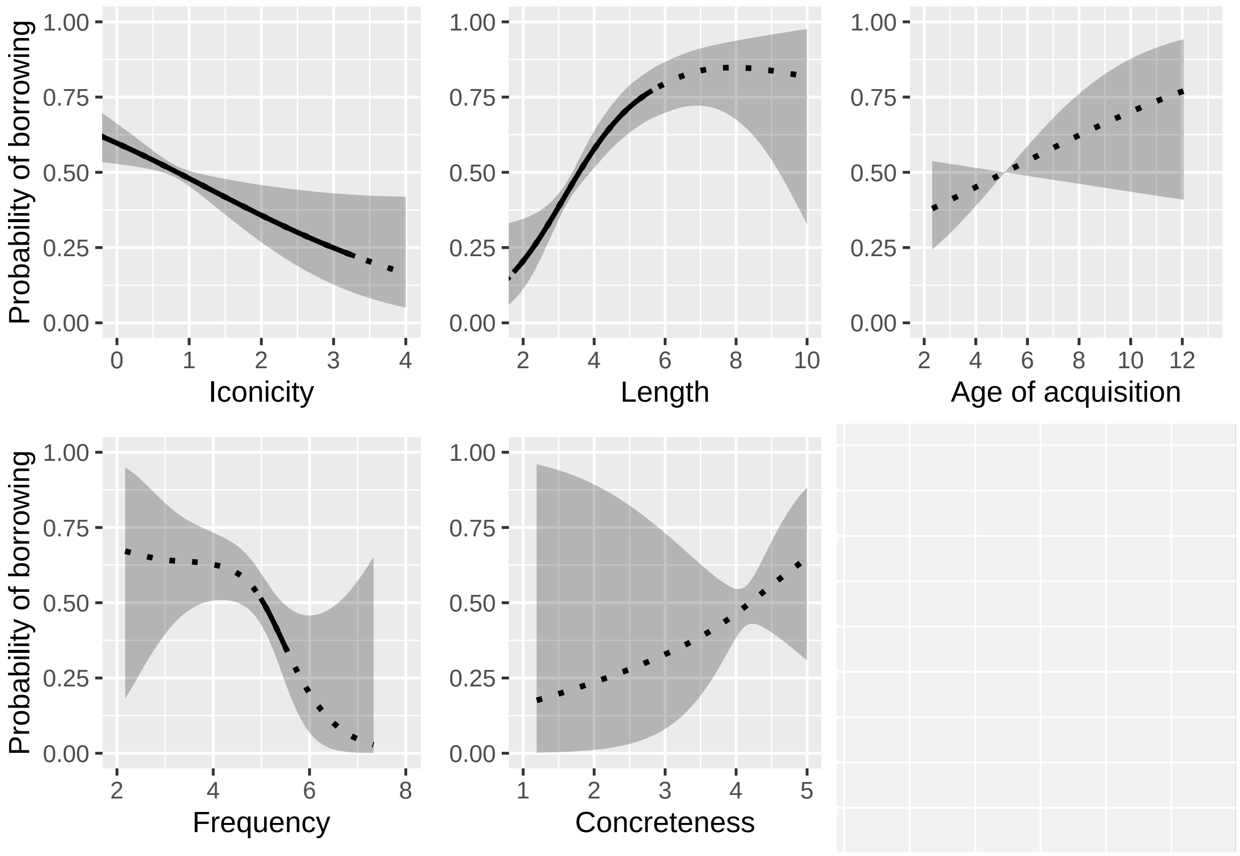
```
m1i= update(m1, data = dataloan2[
  dataloan2$borrowing_type %in% c("Not borrowed","Insertion"),])
cat(paste(capture.output(summary(m1i))[18:32],collapse="\n"))
```

```
## Approximate significance of smooth terms:
##                                edf Ref.df Chi.sq  p-value
## s(subtlexzipfscale)      2.291e+00  2.951  5.454  0.13936
## s(AoAscale)              1.000e+00  1.000  2.352  0.12510
## s(phonlengthscale)       2.366e+00  3.009 34.524 1.55e-07 ***
## s(concscale)             1.309e+00  1.555  1.103  0.55415
## s(iconscale)             1.153e+00  1.291  7.229  0.00930 **
## s(cat)                   1.429e+00 10.000  5.062  0.02904 *
## s(cat,iconscale)         7.345e-02 10.000  0.078  0.30688
## s(cat,phonlengthscale)   2.619e-06 10.000  0.000  1.00000
## s(cat,AoAscale)          1.212e-01 10.000  0.134  0.30272
## s(cat,subtlexzipfscale) 6.090e-06 10.000  0.000  0.55840
## s(cat,concscale)         3.102e+00 10.000 11.586  0.00413 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m1i.vis = visGam(m1i)
```

Visualise effects:

```
plot(m1i.vis)
```



Multinomial model of all types

```
# Code borrowing type as a number 0-3 so that we can run the GAM
dataloan2$borrowing_type_num = as.numeric(dataloan2$borrowing_type) -1

# Run a multinomial GAM
mRCI = gam(list(
          borrowing_type_num ~
          # Model for Replacements
           s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
           s(concscale) + s(iconscale) + s(cat,bs='re'),
          # Model for Coexistence
          ~s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
            s(concscale) + s(iconscale) + s(cat,bs='re'),
          # Model for Insertions
          ~s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
            s(concscale) + s(iconscale) + s(cat,bs='re')
          ),
         data = dataloan2[!is.na(dataloan2$borrowing_type_num),],
         family=multinom(K=3))
```

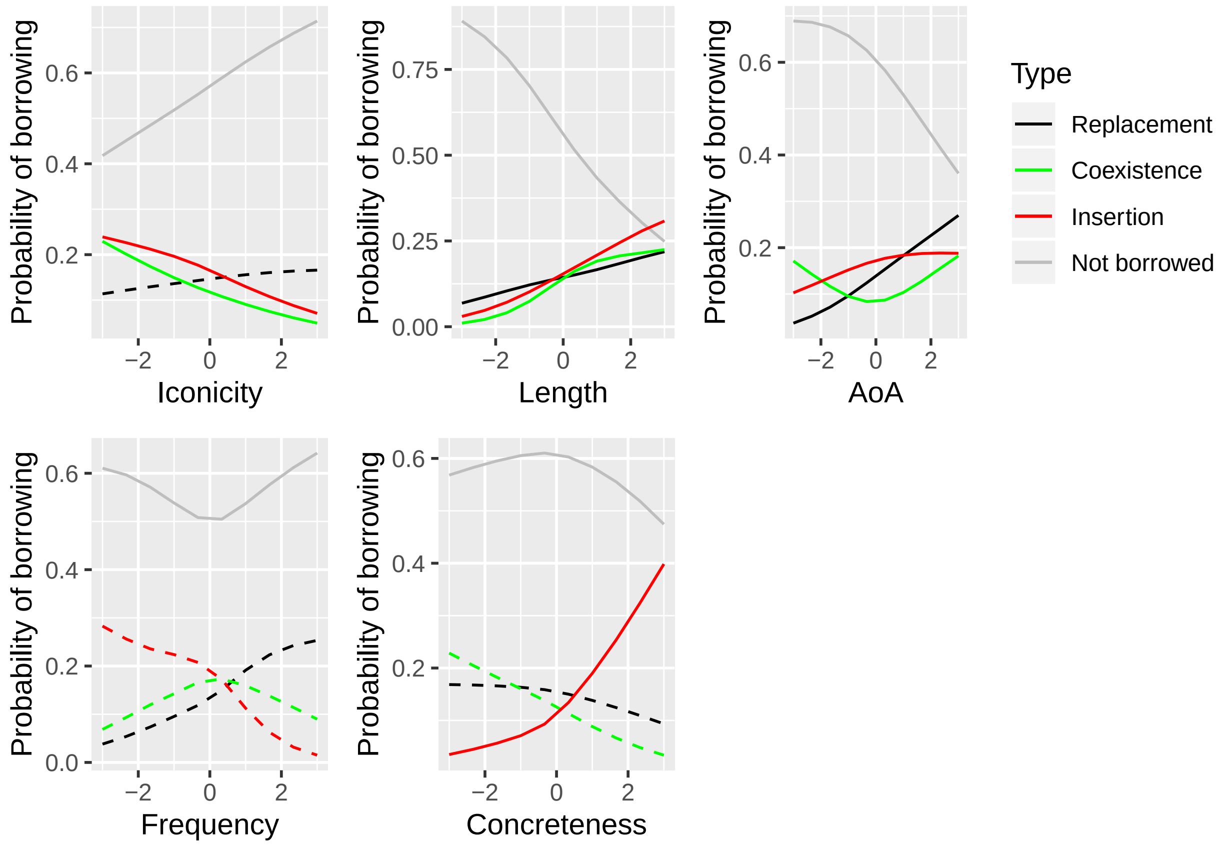
In the summary below, the first set of numbers "s" are for replacements, the second "s.1" are for coexistence and the third "s.2" are for insertions.

```
summary(mRCI)
```

```
##
## Family: multinom
## Link function:
##
## Formula:
## borrowing_type_num ~ s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) +
##     s(concscale) + s(iconscale) + s(cat, bs = "re")
## ~s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) + s(concscale) +
##     s(iconscale) + s(cat, bs = "re")
## ~s(subtlexzipfscale) + s(AoAscale) + s(phonlengthscale) + s(concscale) +
##     s(iconscale) + s(cat, bs = "re")
##
## Parametric coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.2800     0.3426  -6.656 2.82e-11 ***
## (Intercept).1  -1.9664     0.1452 -13.540  < 2e-16 ***
## (Intercept).2  -2.4039     0.3851  -6.242 4.31e-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(subtlexzipfscale)   1.901e+00  2.420  4.026 0.164470
## s(AoAscale)           1.000e+00  1.000  9.575 0.001973 **
## s(phonlengthscale)    1.000e+00  1.000 12.179 0.000484 ***
## s(concscale)          1.000e+00  1.000  0.232 0.630295
## s(iconscale)          1.000e+00  1.001  0.353 0.552863
## s(cat)                2.816e+00 10.000  8.470 0.012179 *
## s.1(subtlexzipfscale) 2.457e+00  3.158  4.160 0.260628
## s.1(AoAscale)         2.421e+00  3.072 10.326 0.017472 *
## s.1(phonlengthscale)  2.132e+00  2.712 27.777 3.78e-06 ***
## s.1(concscale)        1.000e+00  1.000  3.056 0.080437 .
## s.1(iconscale)        1.000e+00  1.000  9.866 0.001684 **
## s.1(cat)              1.272e-05 10.000  0.000 0.577444
## s.2(subtlexzipfscale) 2.545e+00  3.263  4.716 0.215985
## s.2(AoAscale)         1.000e+00  1.000  4.128 0.042171 *
## s.2(phonlengthscale)  1.546e+00  1.927 41.016 9.01e-09 ***
## s.2(concscale)        1.527e+00  1.885  6.757 0.022753 *
## s.2(iconscale)        1.389e+00  1.692 12.741 0.003188 **
## s.2(cat)              1.537e+00 10.000  4.197 0.049064 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Deviance explained = 15.5%
## -REML = 203.28  Scale est. = 1         n = 738
```

The exact relationships are hard to determine from the model output, so we also visalise the model. Note that these graphs are not representations of the data, but representations of the model estimates.

```r
getPred = function(gModel){
  # Get model predictions for range of variable values
  nx = 10
  gr = expand.grid(
    subtlexzipfscale = seq(-3,3,length.out = nx),
    AoAscale = seq(-3,3,length.out = nx),
    phonlengthscale = seq(-3,3,length.out = nx),
    concscale = seq(-3,3,length.out = nx),
    iconscale = seq(-3,3,length.out = nx)
  )
  gr$cat = "Noun"
  pred = predict(gModel,newdata=gr, type="response")
  colnames(pred) = c("Not borrowed","Replacement","Coexistence","Insertion")
  gr = cbind(gr,pred)
  return(gr)

}

visualiseMultinomGamModel = function(gr,fac,xlabel,rType,cType,iType){

  grs = gr %>% group_by(!!as.name(fac)) %>%
    summarise(R=mean(Replacement),
              C=mean(Coexistence),
              I=mean(Insertion),
              NB=mean(`Not borrowed`))
  names(grs)[1] = "subtlexzipfscale"

  gx =  ggplot(grs,aes(x=subtlexzipfscale,y=R)) +
    geom_line(mapping = aes(y=NB,colour="gray")) +
    geom_line(aes(colour="black"),linetype = rType) +
    geom_line(mapping = aes(y=C,colour="green"),linetype = cType) +
    geom_line(mapping = aes(y=I,colour="red"),linetype = iType) +
    ylab("Probability of borrowing") + xlab(xlabel) +
    scale_color_identity(name = "Type", breaks=c("black","green","red","gray"),
          labels = c("Replacement", "Coexistence", "Insertion","Not borrowed"),
                  guide="legend") #+ ylim(0,0.3)
  return(gx)
}

gr = getPred(mRCI)

grid.arrange(
  visualiseMultinomGamModel(gr,"iconscale","Iconicity","dashed","solid","solid") +
    theme(legend.position = "none"),
  visualiseMultinomGamModel(gr,"subtlexzipfscale","Frequency","dashed","dashed","dashed") +
    theme(legend.position = "none"),
  visualiseMultinomGamModel(gr,"phonlengthscale","Length","solid","solid","solid") +
    theme(legend.position = "none"),
  visualiseMultinomGamModel(gr,"concscale","Concreteness","dashed","dashed","solid") +
    theme(legend.position = "none"),
  visualiseMultinomGamModel(gr,"AoAscale","AoA","solid","solid","solid"),
layout_matrix = matrix(1:6,nrow=2),
widths = c(1,1,1.6)
)
```



The figure above shows the model esimates from multinomial model. The probabilities on the y-axis are the model's estimated likelihood of a word belonging to each of the three borrowed categories. The lines show these probabilities change according to different values of the x-axis. Lines are solid if the effect is significant and dotted if the effect is not significant. For example, the probabilities of all three types of borrowing increase as length increases. Logically, this implies that the probability of being non-borrowed decreases as word length increases.

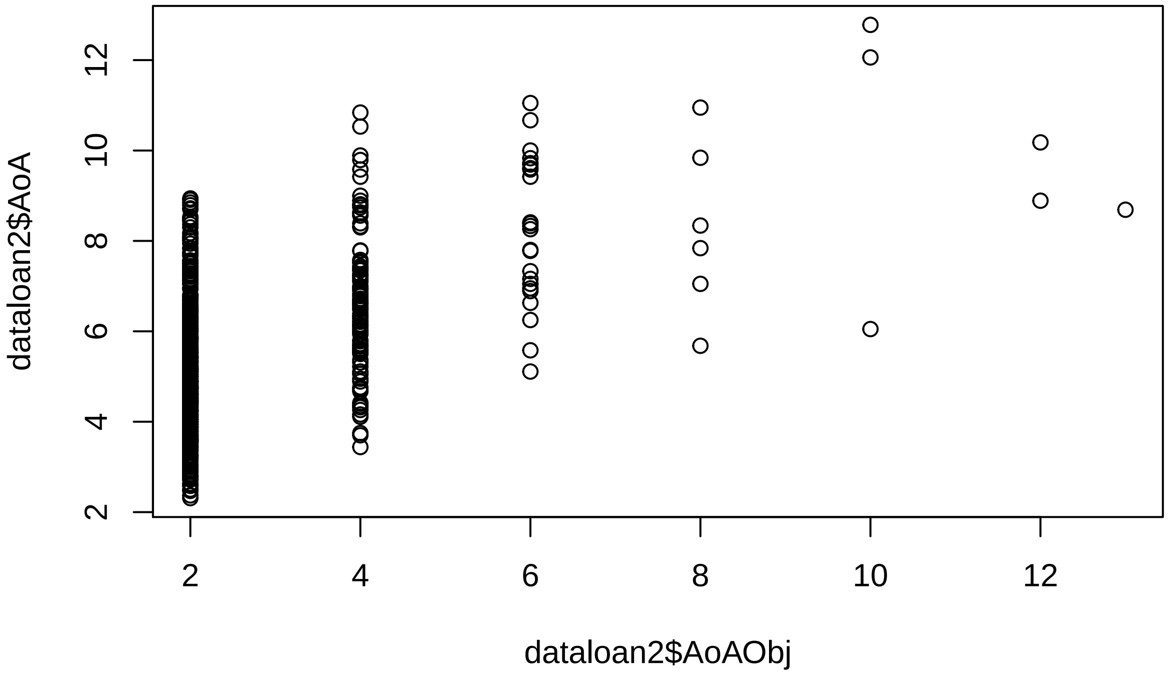
Objective age of acquisition

Run a model with objective age of acqusition instead of the ratings measure. Objective measure taken from test-based age of acquisition from Brysbaert & Biemiller (2017).

```
obj = read.csv("../data/Brysbaert_Biemiller_TestBasedAoAMeasures.csv",stringsAsFactors = F, encoding =
obj[obj$WORD=="sew up",]$WORD = "sew"
dataloan2$AoAObj = obj[match(dataloan2$word,obj$WORD),]$AoAtestbased
dataloan2$AoAObjScale = scale(dataloan2$AoAObj)
```

The correlation between objective and ratings-based AoA is high:

```
plot(dataloan2$AoA~dataloan2$AoAObj)
```



```
cor.test(dataloan2$AoA,dataloan2$AoAObj,na.rm=T)
```

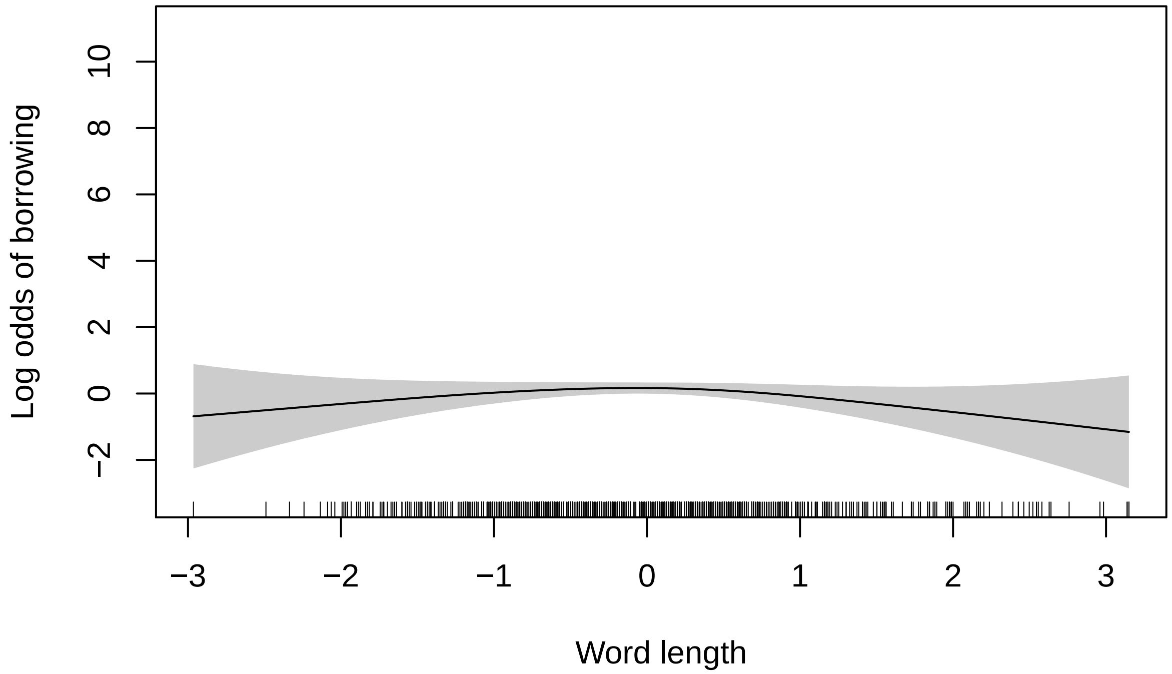
```
##
##  Pearson's product-moment correlation
##
## data:  dataloan2$AoA and dataloan2$AoAObj
## t = 18.334, df = 782, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4973704 0.5954528
## sample estimates:
##       cor
## 0.5482943
```

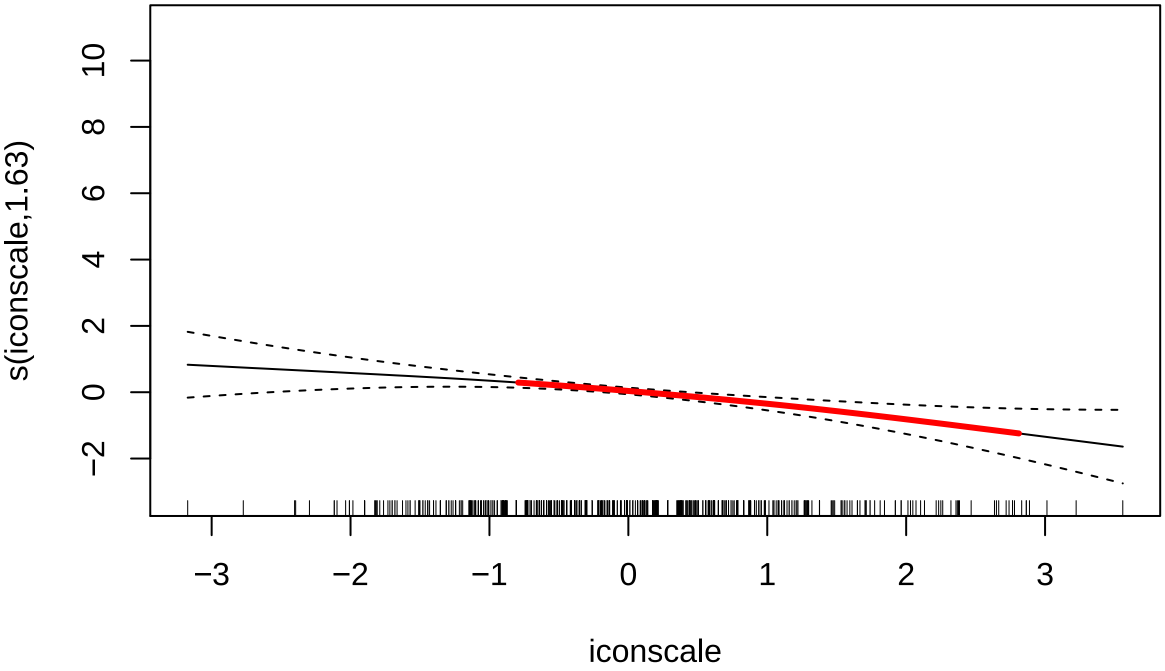
Run model with objective AoA. Note that the results are qualitatively the same as the main model.

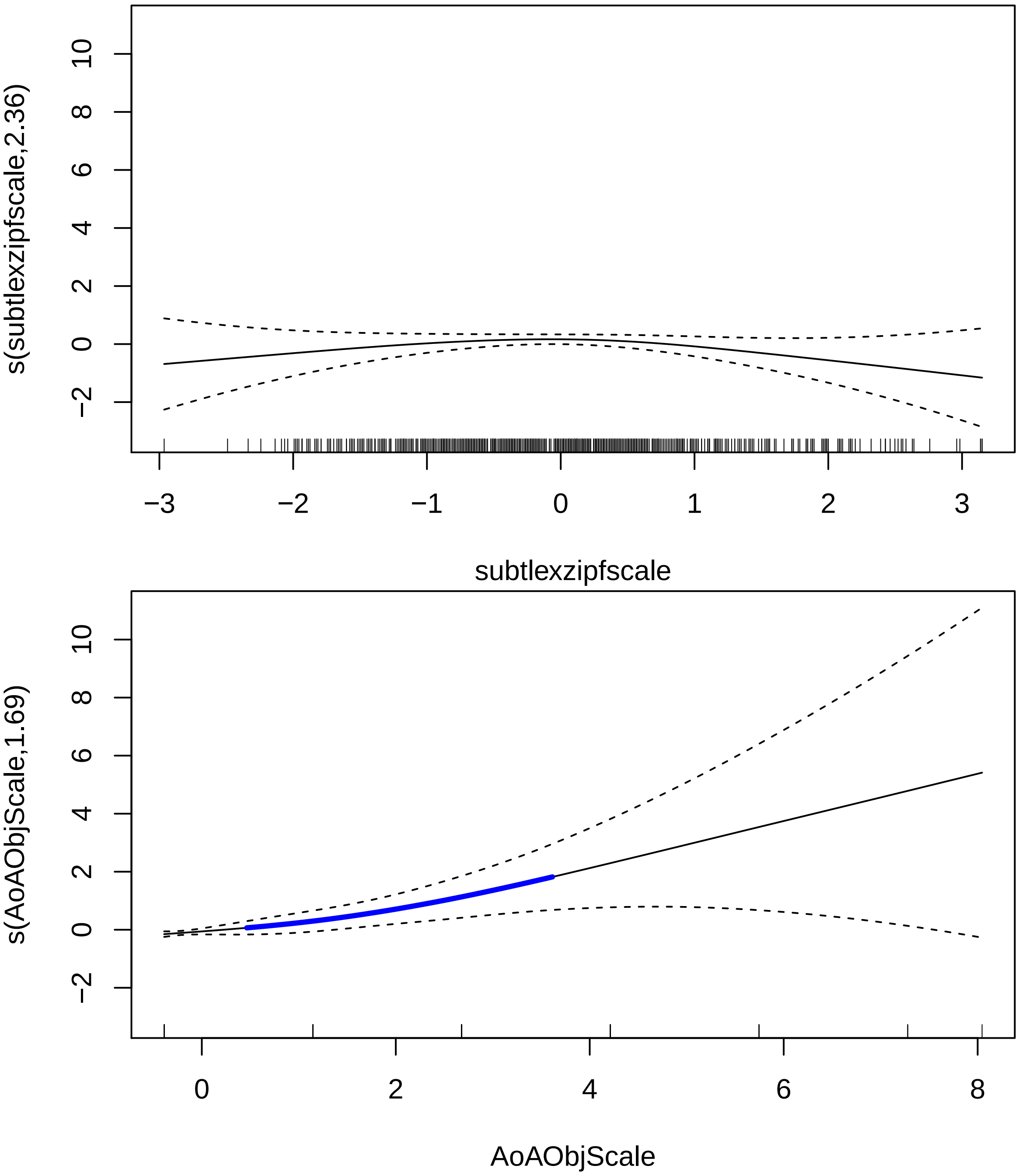
```
m1Obj = bam(bor15.cat ~
      s(subtlexzipfscale) +
      s(AoAObjScale, k = 6) +
      s(phonlengthscale) +
      s(concscale) +
      s(iconscale) +
      s(cat,bs='re')+
      s(cat,iconscale,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAObjScale,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    data = dataloan2,
    family='binomial')

summary(m1Obj)
```

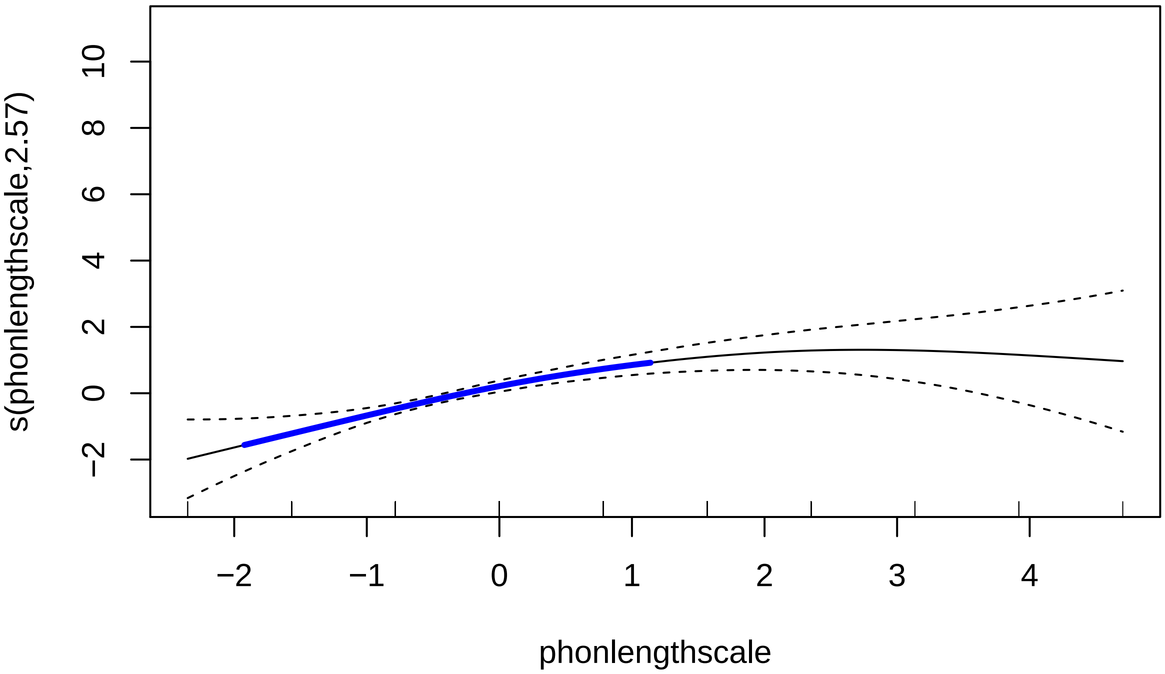
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(subtlexzipfscale) + s(AoAObjScale, k = 6) + s(phonlengthscale) +
##     s(concscale) + s(iconscale) + s(cat, bs = "re") + s(cat,
##     iconscale, bs = "re") + s(cat, phonlengthscale, bs = "re") +
##     s(cat, AoAObjScale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2967     0.3831  -3.384 0.000714 ***
## ---
## 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(subtlexzipfscale)     2.356e+00  3.024  4.656 0.201487
## s(AoAObjScale)          1.686e+00  2.046 11.702 0.003200 **
## s(phonlengthscale)      2.568e+00  3.276 39.509 2.92e-08 ***
## s(concscale)            1.000e+00  1.000  1.489 0.222317
## s(iconscale)            1.626e+00  2.052 16.204 0.000342 ***
## s(cat)                  4.169e+00 10.000 19.228 0.000160 ***
## s(cat,iconscale)        1.592e-04 10.000  0.000 0.352860
## s(cat,phonlengthscale)  2.521e-01 10.000  0.307 0.288516
## s(cat,AoAObjScale)      3.003e-06 10.000  0.000 0.660658
## s(cat,subtlexzipfscale) 8.654e-01 10.000  1.310 0.192095
## s(cat,concscale)        4.152e-06 10.000  0.000 0.713234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.198   Deviance explained = 17.6%
## fREML = 1113.4  Scale est. = 1         n = 784
```

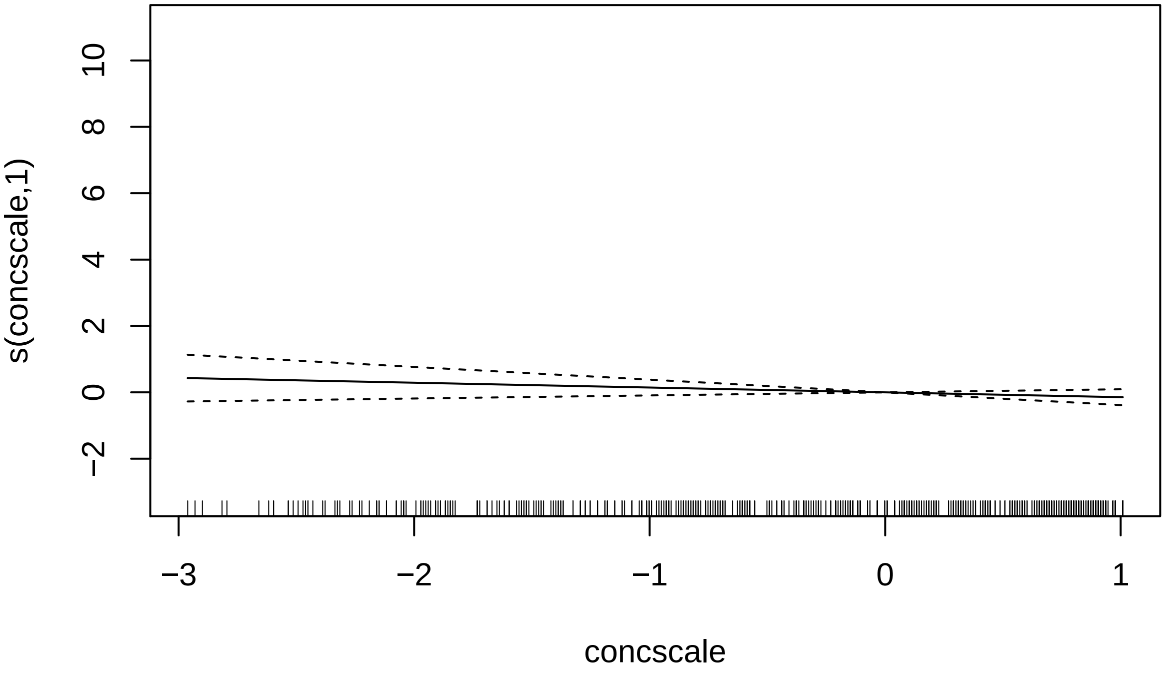
Visualise (code as above, but hidden). For age of acquisition, there is more uncertainty around the later ages, but in general the results are very similar.


Log odds of borrowing
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3
Word length

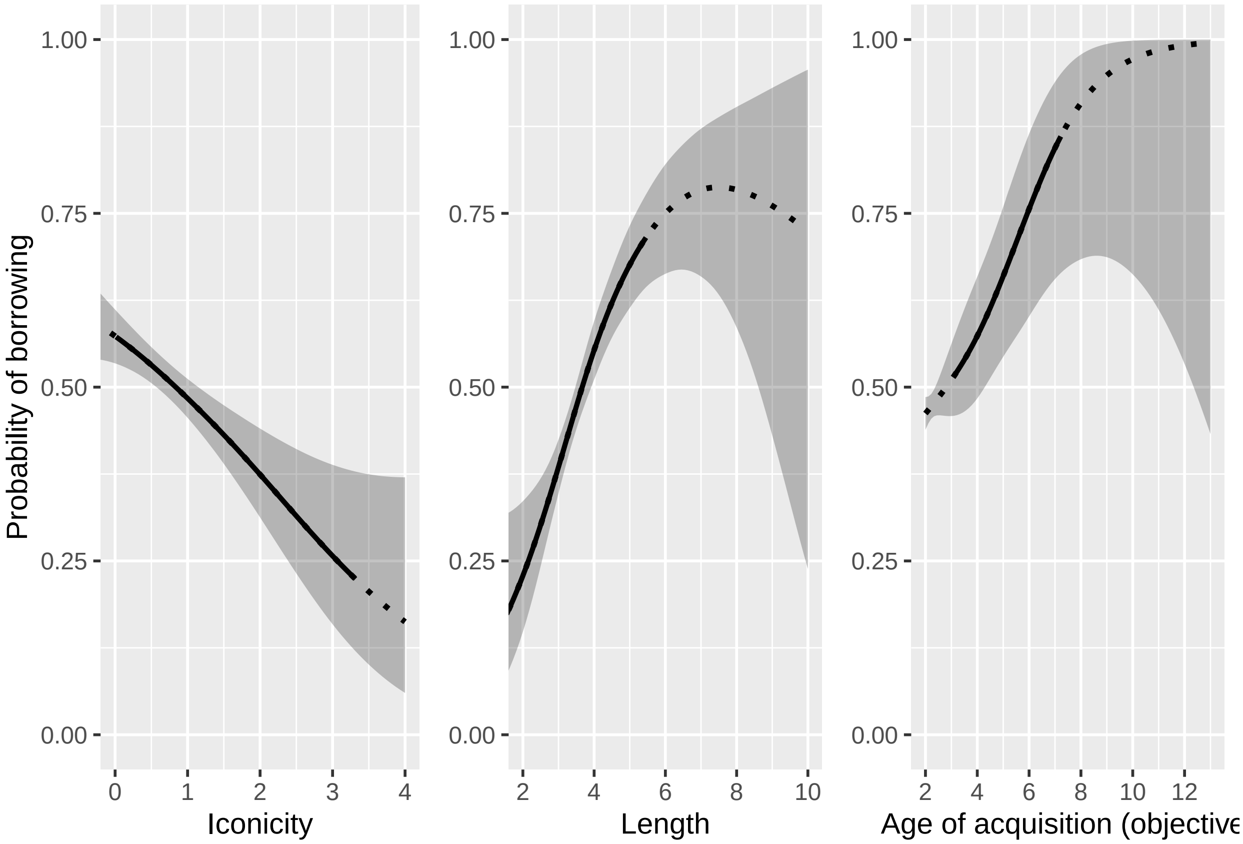

s(iconscale,1.63)
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3
iconscale

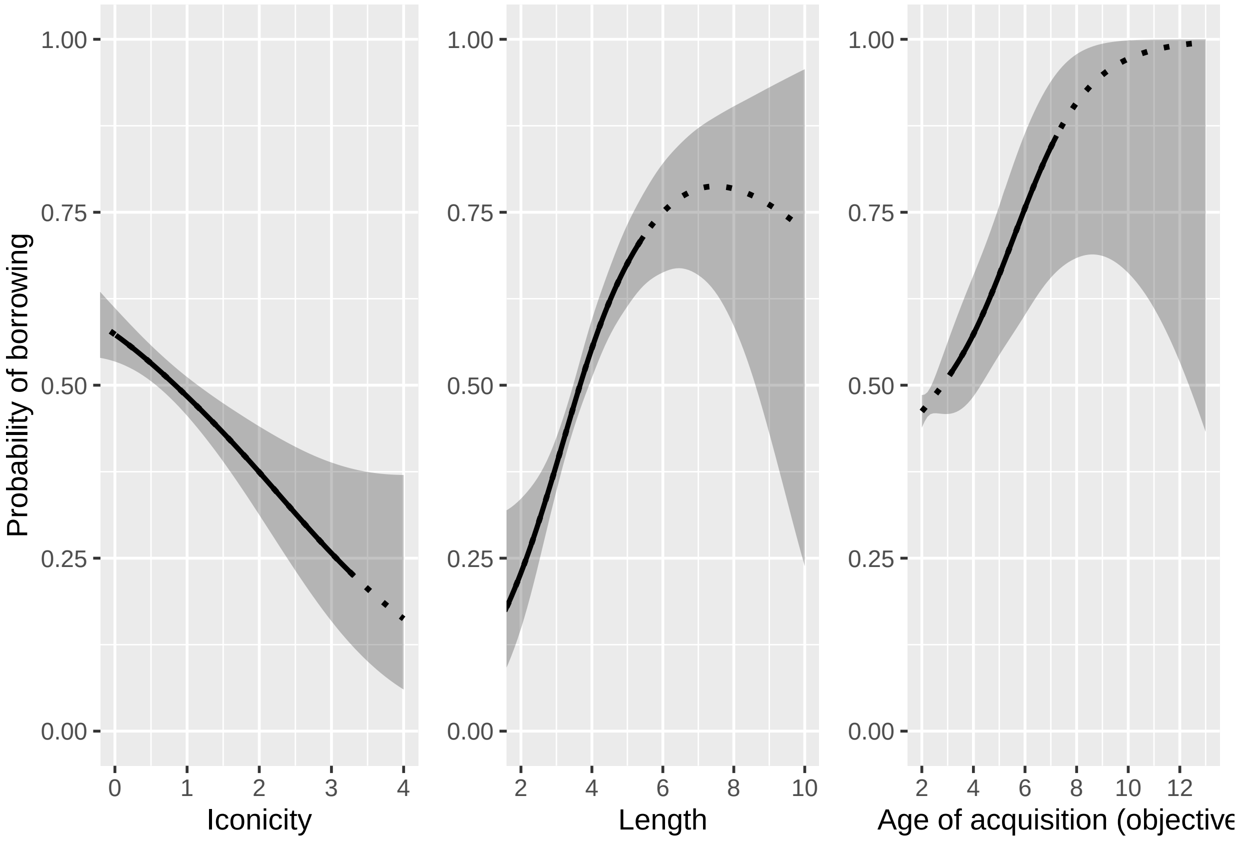

s(subtlexzipfscale,2.36)
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3
subtlexzipfscale
s(AoAObjScale,1.69)
10
8
6
4
2
0
−2
0
2
4
6
8
AoAObjScale





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

## Warning: Removed 46 row(s) containing missing values (geom_path).

## Warning: Removed 56 row(s) containing missing values (geom_path).

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

## Warning: Removed 62 row(s) containing missing values (geom_path).

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


Probability of borrowing
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
Iconicity
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
Length
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
Age of acquisition (objective


Probability of borrowing
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
Iconicity
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
Length
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
Age of acquisition (objective