

Supporting materials for Iconicity and diachronic language change

Introduction

This file shows the R code for the analyses in *Iconicity and diachronic language 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 = definitely 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).
- AoA_obj: Objective, test-based age of acquisition from Brysbaert & Biemiller (2017)
- sublexzipf: 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 estimate of date 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)

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

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

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.

```
# 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$r # Matrix of correlation coefficients
  p <- correlation_matrix$p # Matrix of p-value

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

  ## truncate 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 appropriate 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_derivatives.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.

```

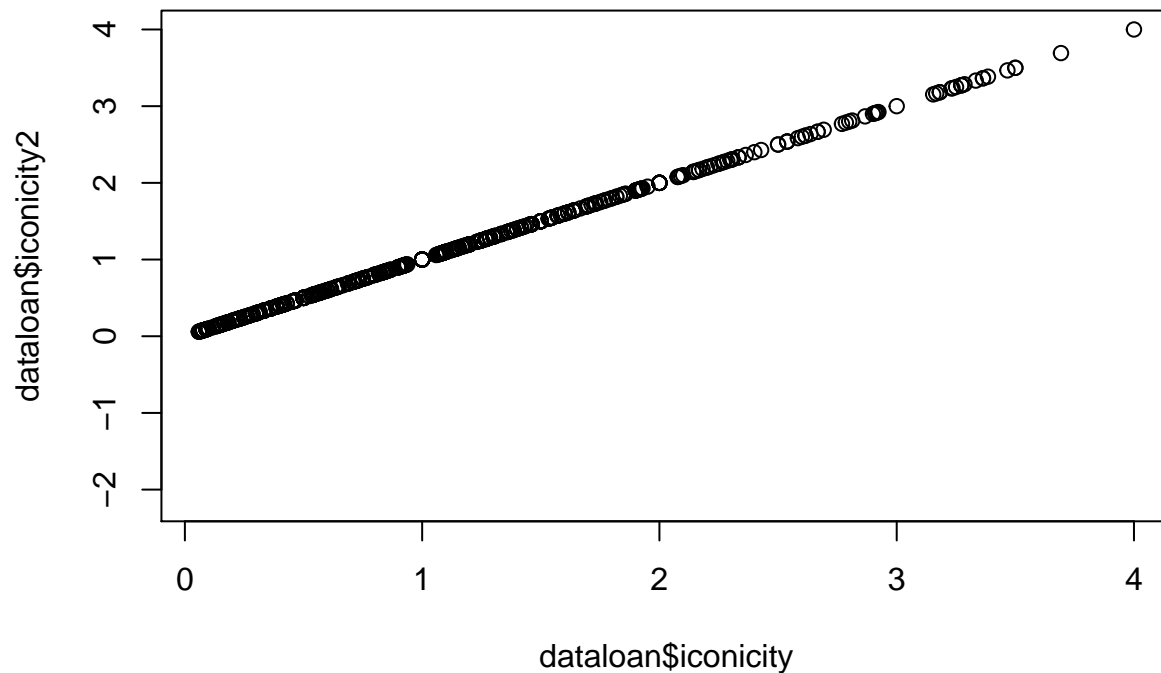
datafile = "loanwords_withiconicity_systematicity.csv"
dataloan <- read.csv(datafile,stringsAsFactors = F)
dataloan$pagel_rate = as.numeric(dataloan$pagel_rate)
dataloan$iconicity = as.numeric(dataloan$iconicity)

```

```
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 iconicity 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("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
## sublexzipf 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",
      "sublexzipf", "cat", "iconicity",
      'conc', 'bor15'))],]

```

This leaves 784 observations.

Make correlation table:

```

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

```

```

##           AoA      phonlength  conc      iconicity
## sublexzipf "-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("sublexzipf",
      "AoA", "phonlength",
      'conc',
      "iconicity")],
    removeTriangle = "lower")

```

```

##           AoA      phonlength  conc      iconicity
## sublexzipf "-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)

```

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

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

##
## Call:
## glm(formula = bor15.cat ~ subtlxzipf + 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
## subtlxzipf     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 main data. 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:

```
m0 = bam(bor15.cat ~
         s(phonlengthscale) +
         s(AoAscale) +
         s(subtlxzipfscale) +
         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)  -1.4141      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(phonlengthscale)      2.611e+00  3.330 50.595 1.74e-10 ***
## s(AoAscale)              1.386e+00  1.684 16.194 0.000163 ***
## s(subtlexzipfscale)      2.183e+00  2.799  4.614 0.203558
## 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, including iconicity as a predictor:

```

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="GAM_results.csv")
```

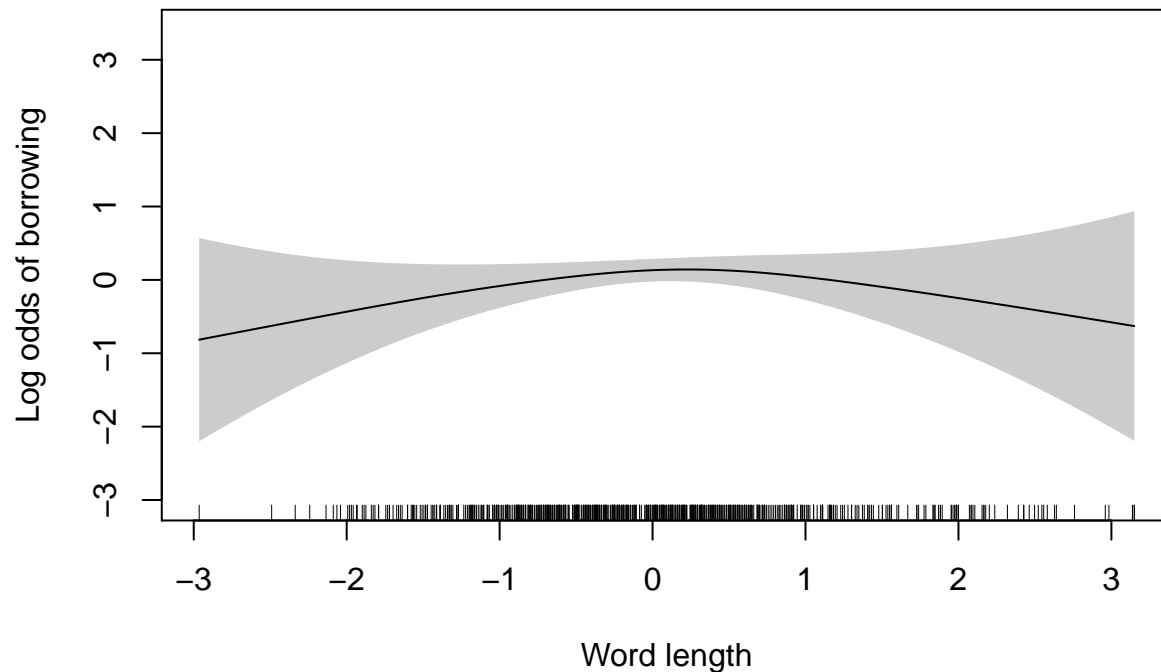
Test only iconicity to show that it is an independent predictor:

```
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

```
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))

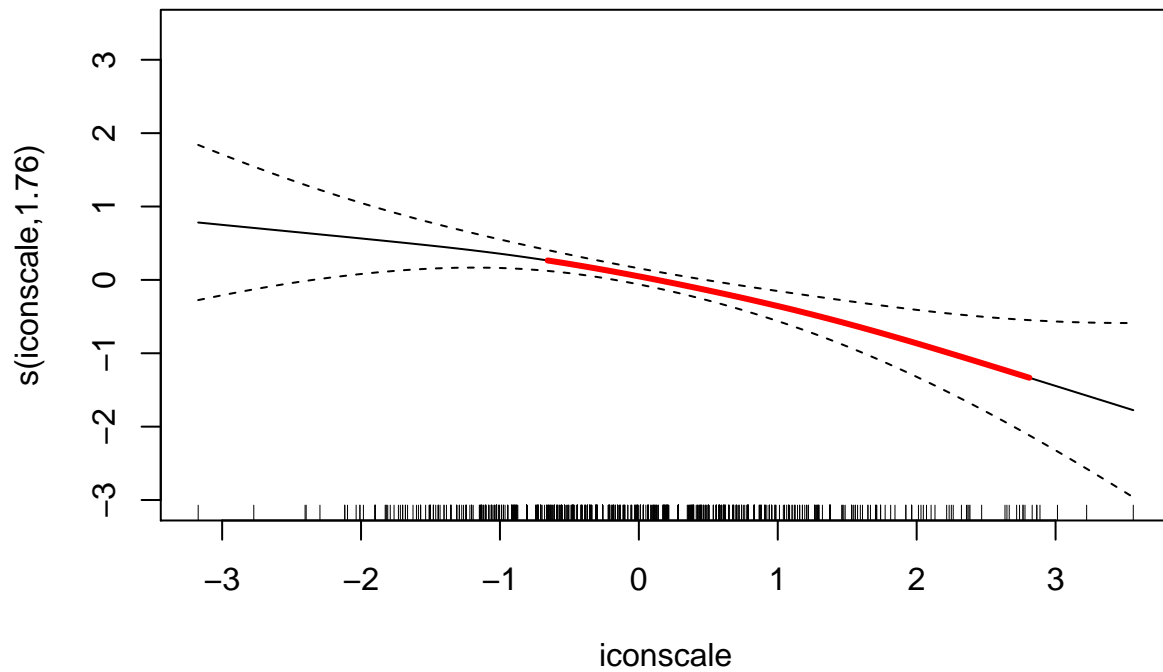
```

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.

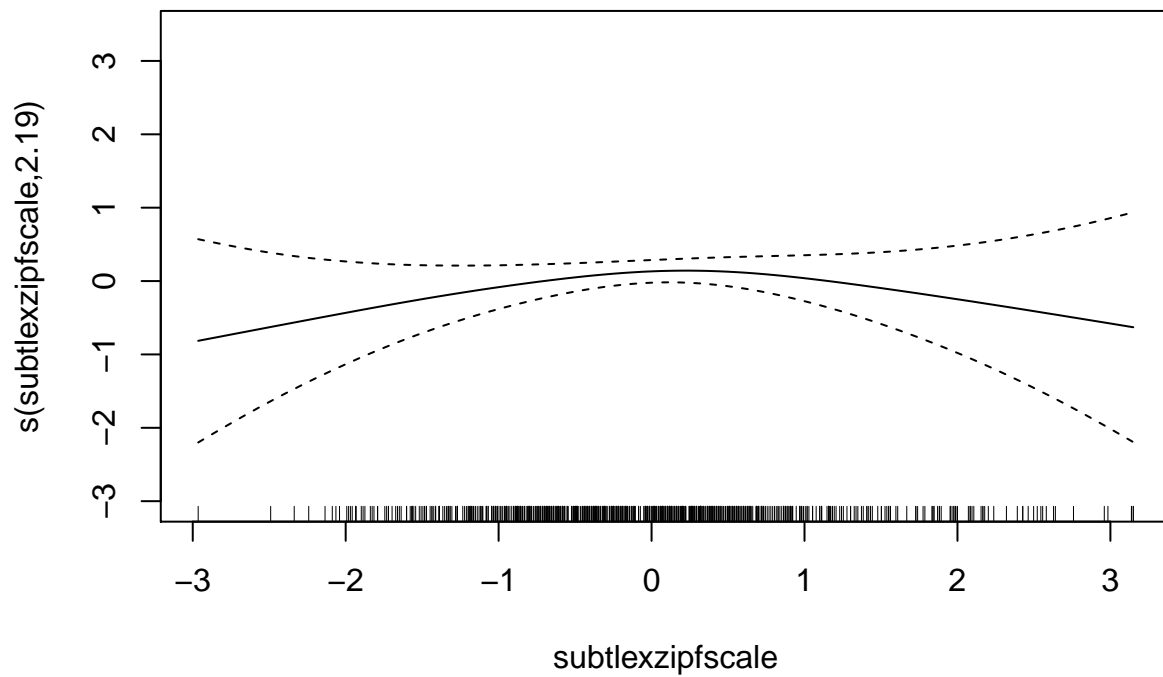
```

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

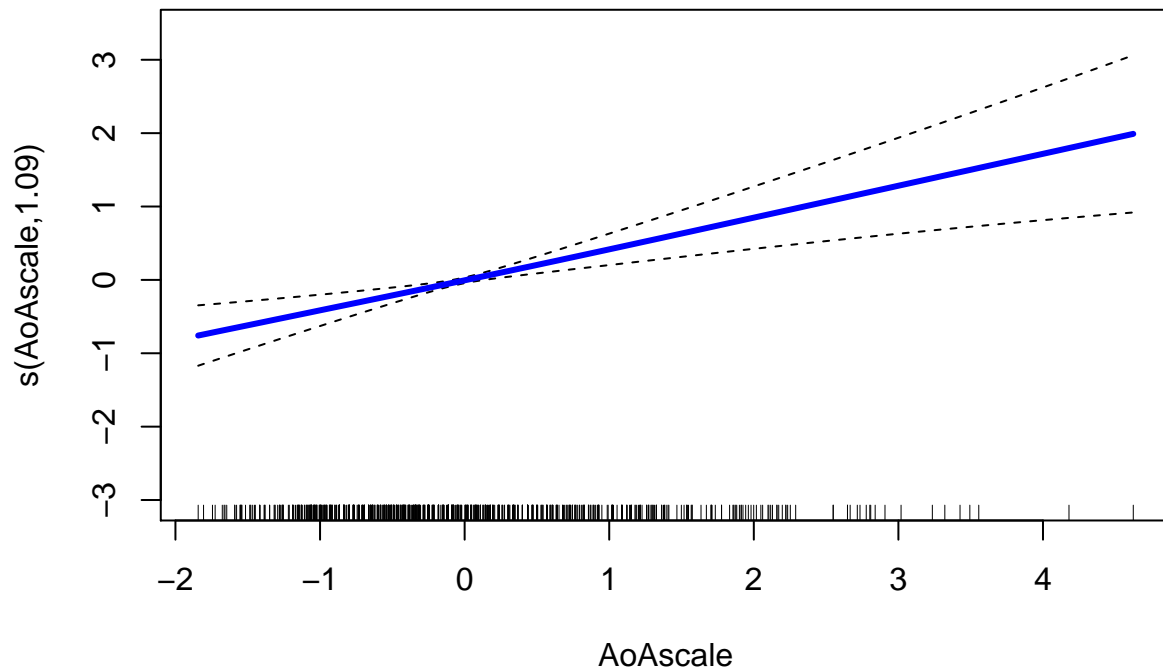
```



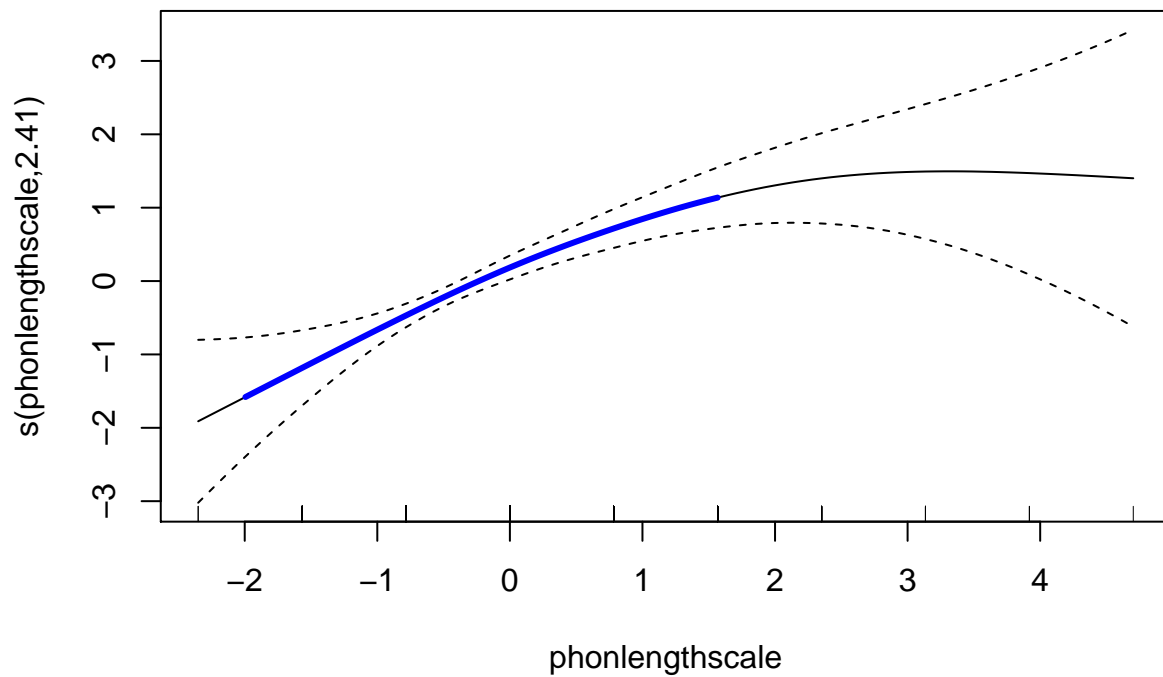
```
pSigFreq = plotGAMSignificantSlopes(m1,"subtlexzipfscale","Frequency")
```



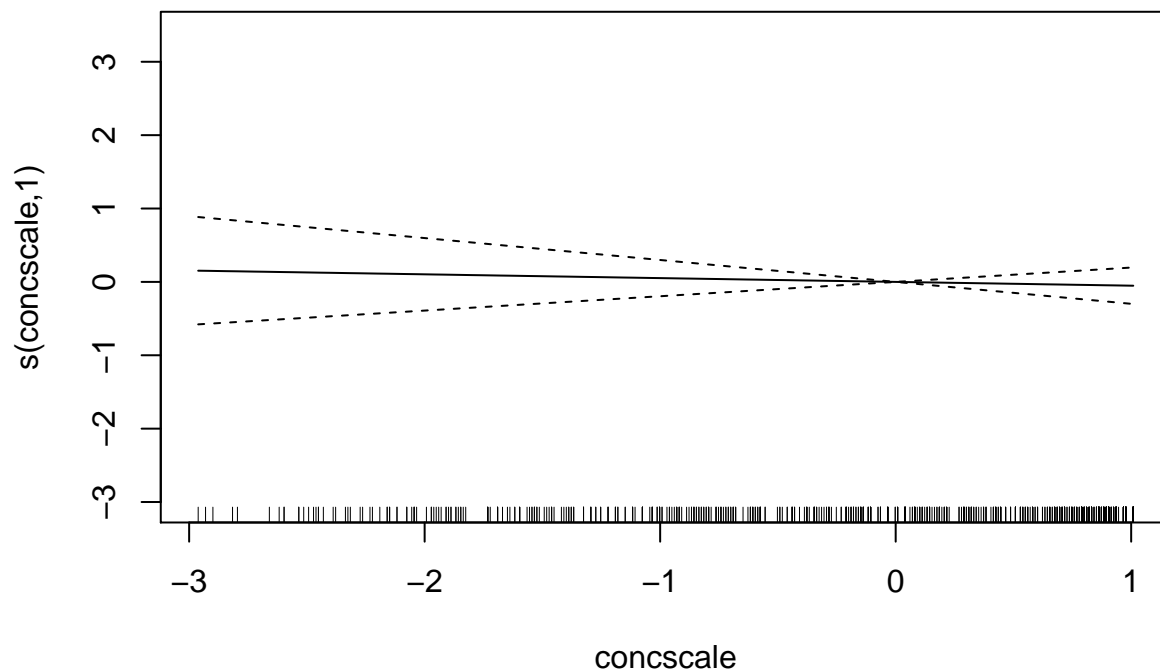
```
pSigAoA = plotGAMSignificantSlopes(m1,"AoAscale","AoA")
```



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



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



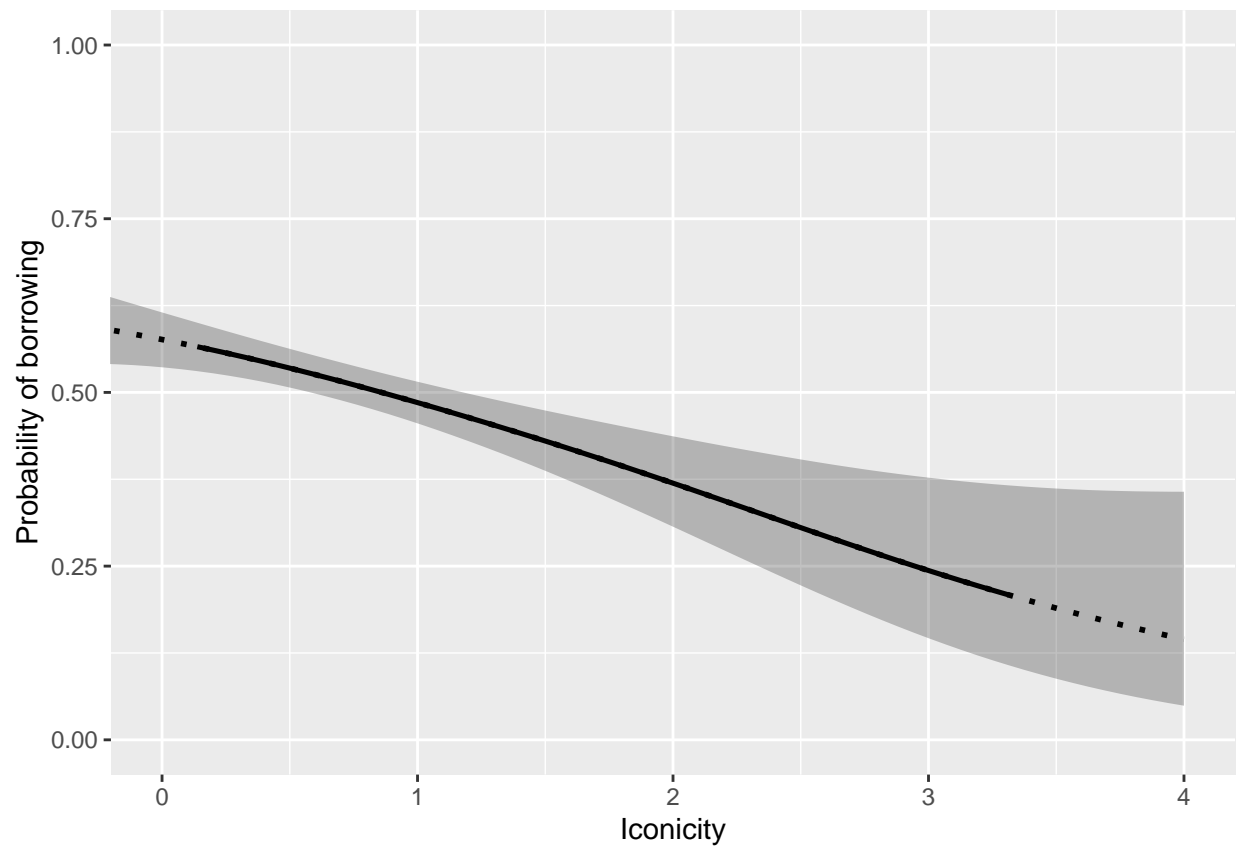
Plot increasing/decreasing curves onto the paoa,pfreq,picon,plen figures above.

```
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)

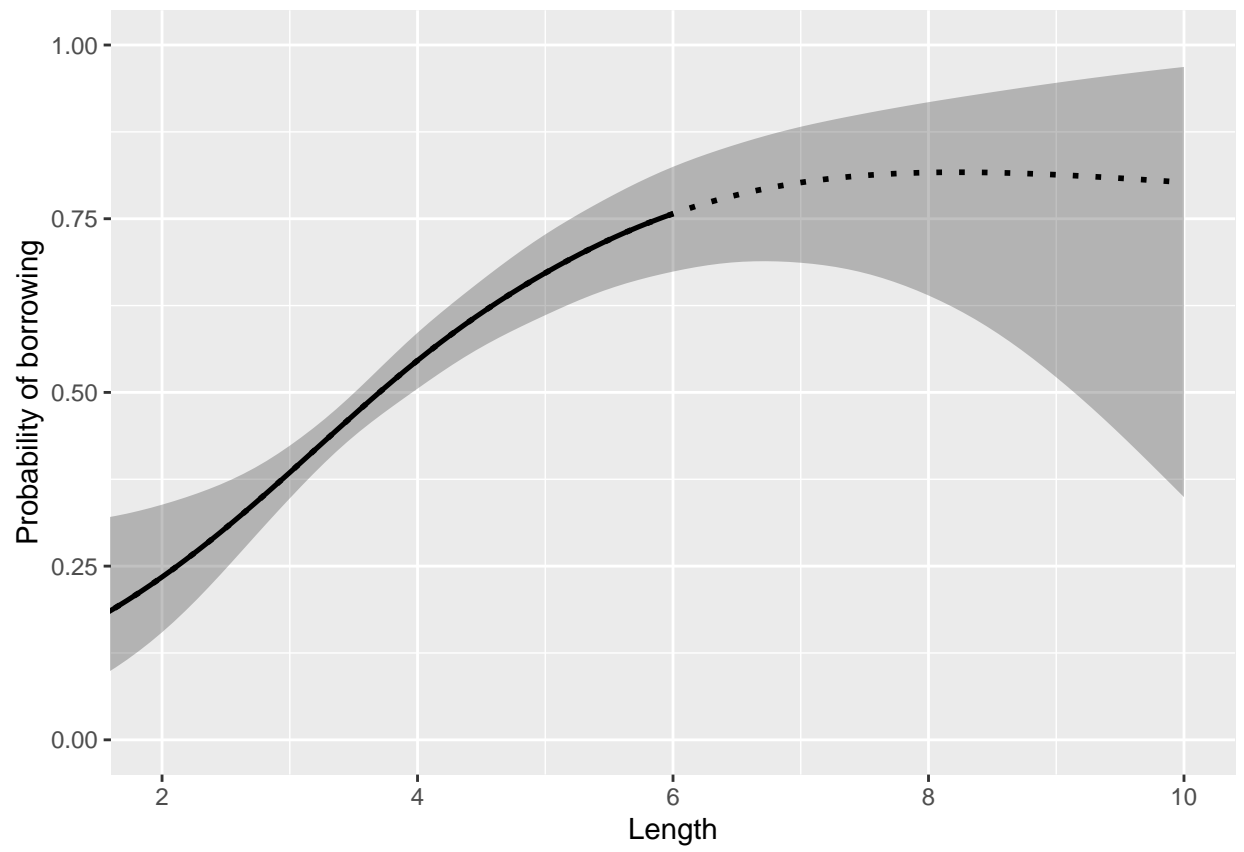
  ret = origPlot +
    geom_path(data = sigCurveData,
              aes(x = curve.x,
                  y = m2.dsig.incr),
              size=0.9) +
    geom_path(data = sigCurveData,
              aes(x = curve.x,
                  y = m2.dsig.decr),
              size=0.9) +
    theme(legend.position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

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

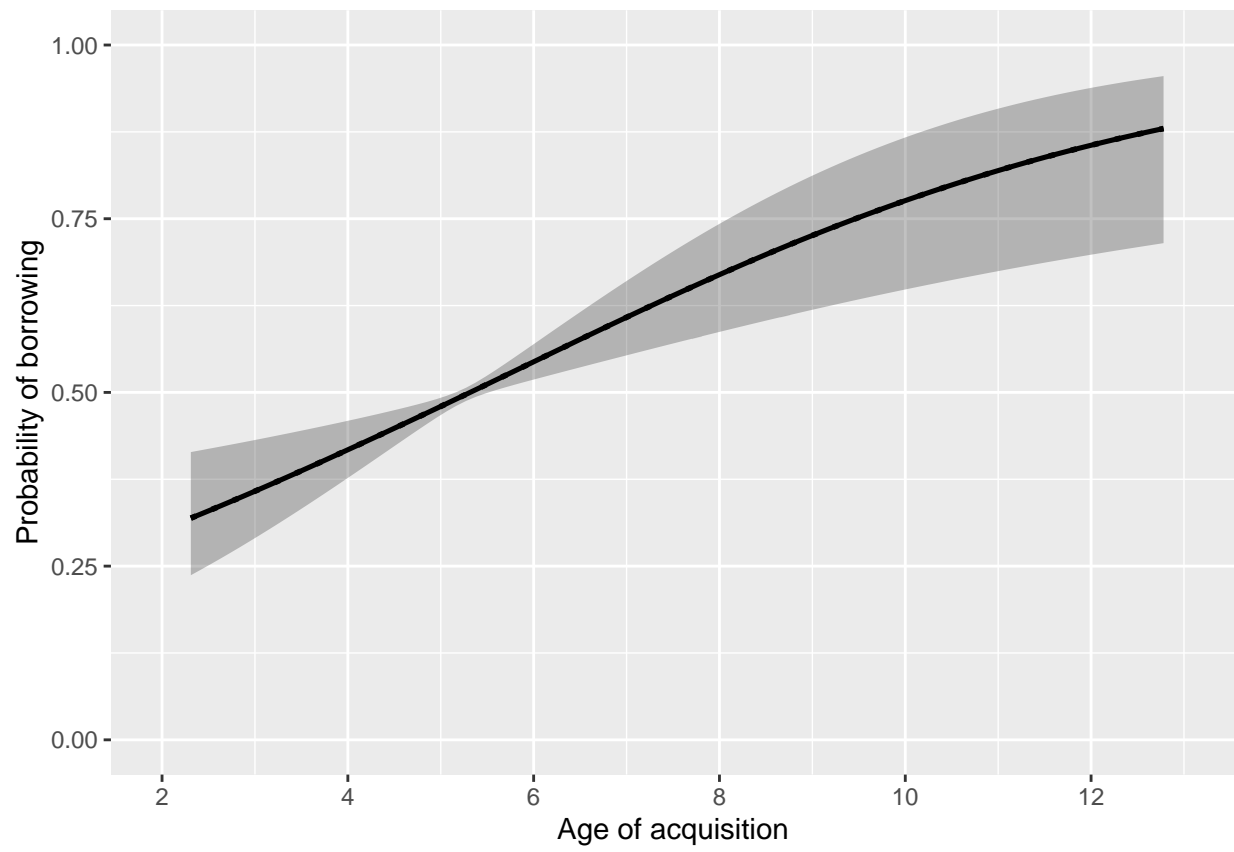
pSigIcon2
```



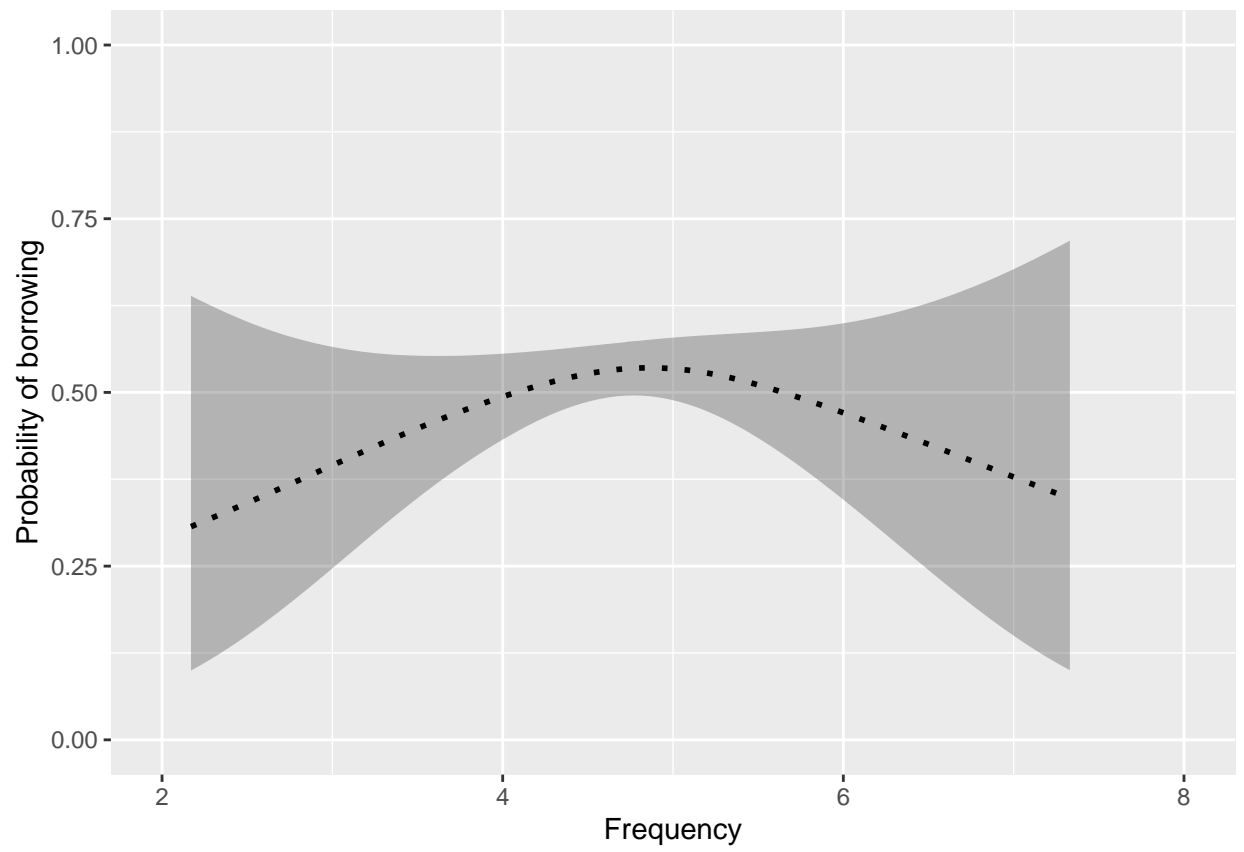
pSigLen2



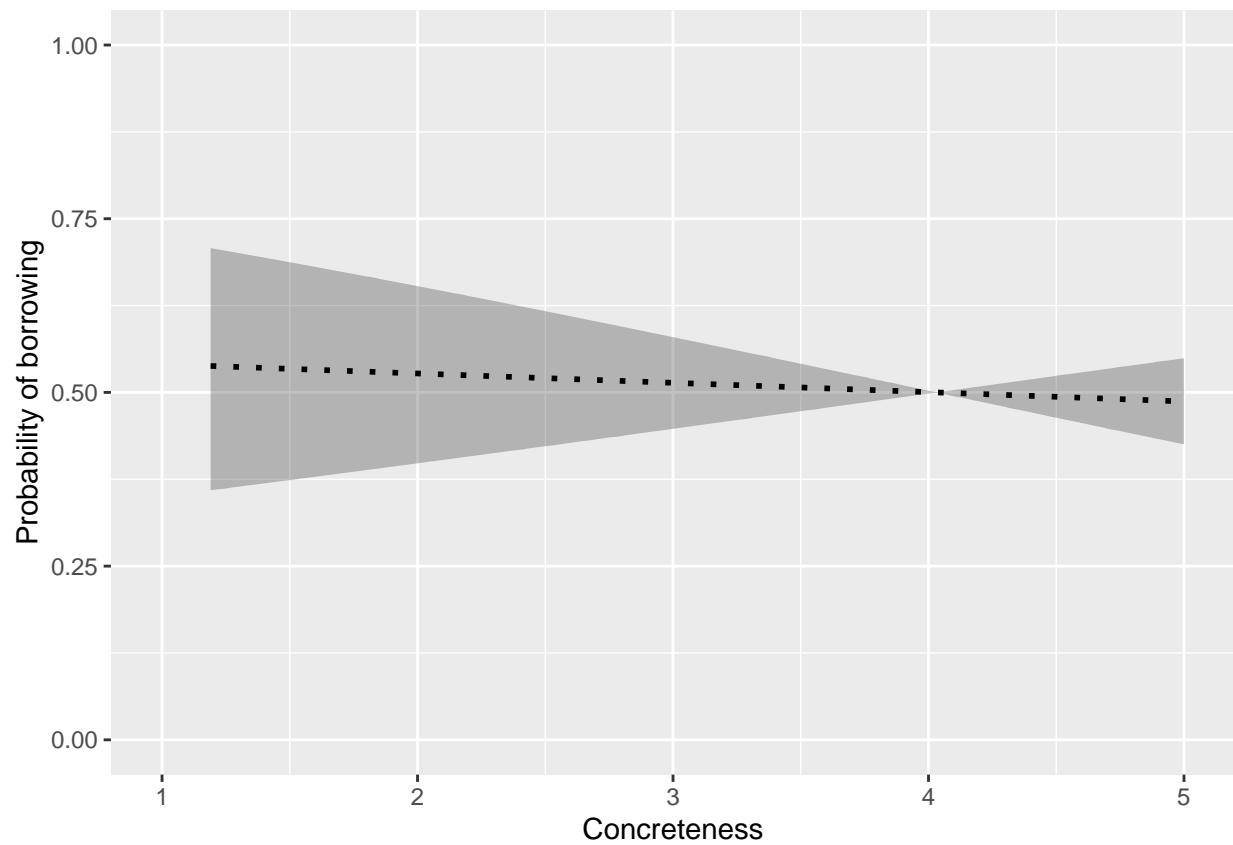
pSigAoA2



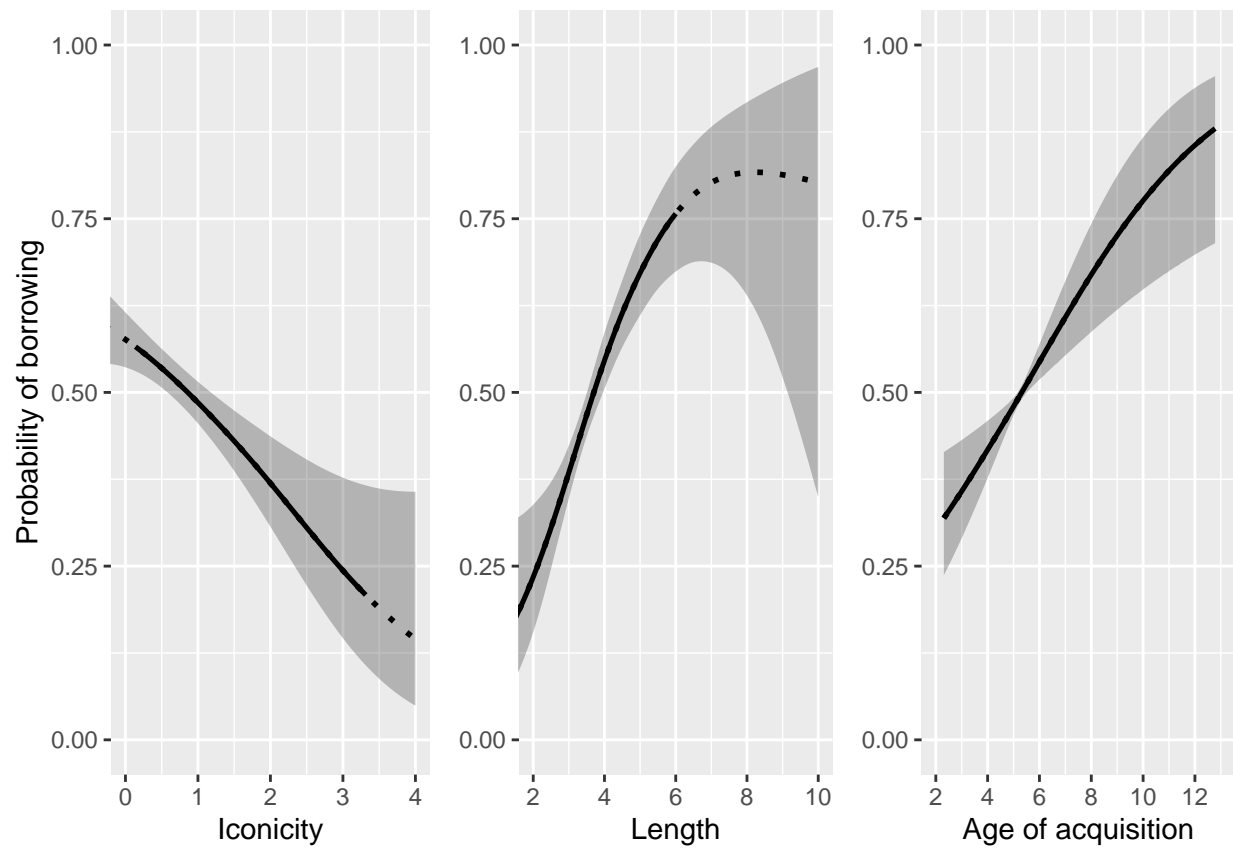
pSigFreq2



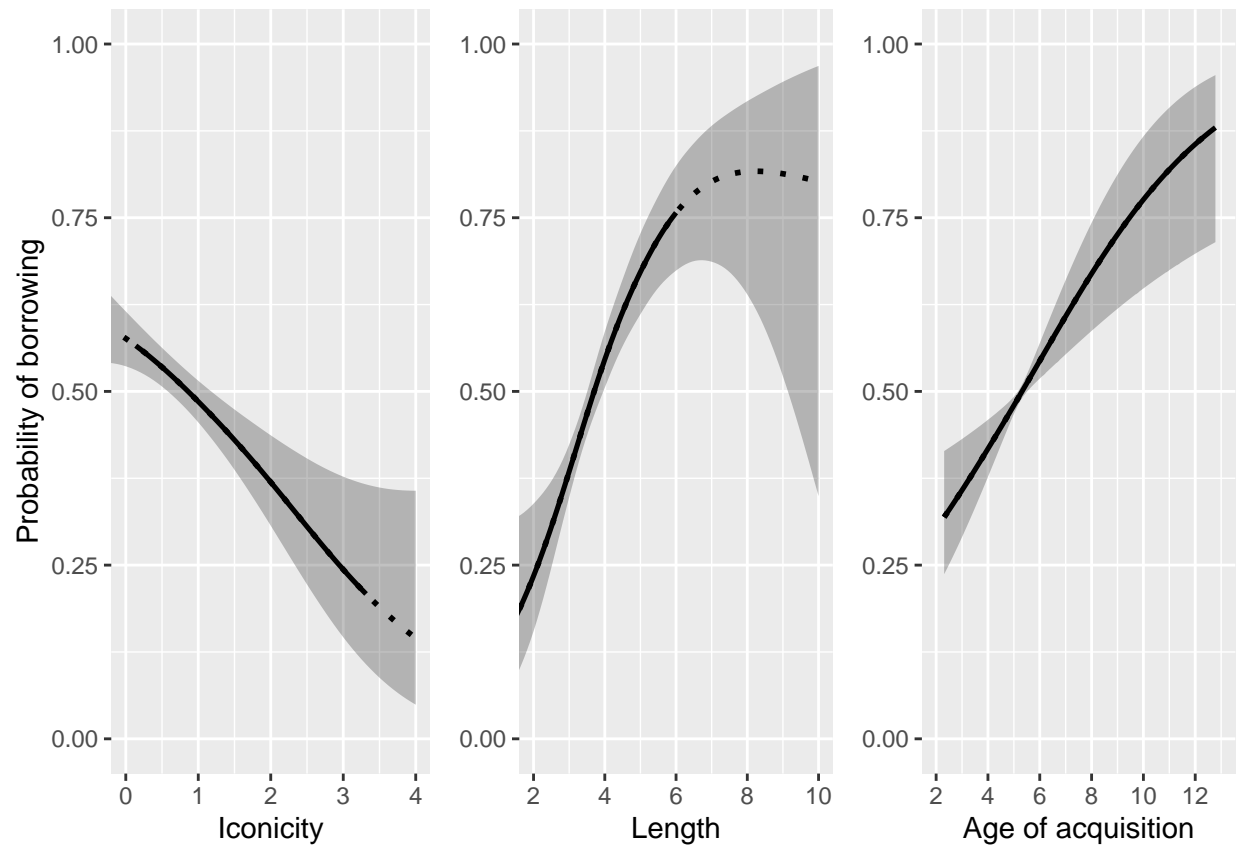
pSigConc2



```
# 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))
```



```
plot(gx)
```



Output to file:

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

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

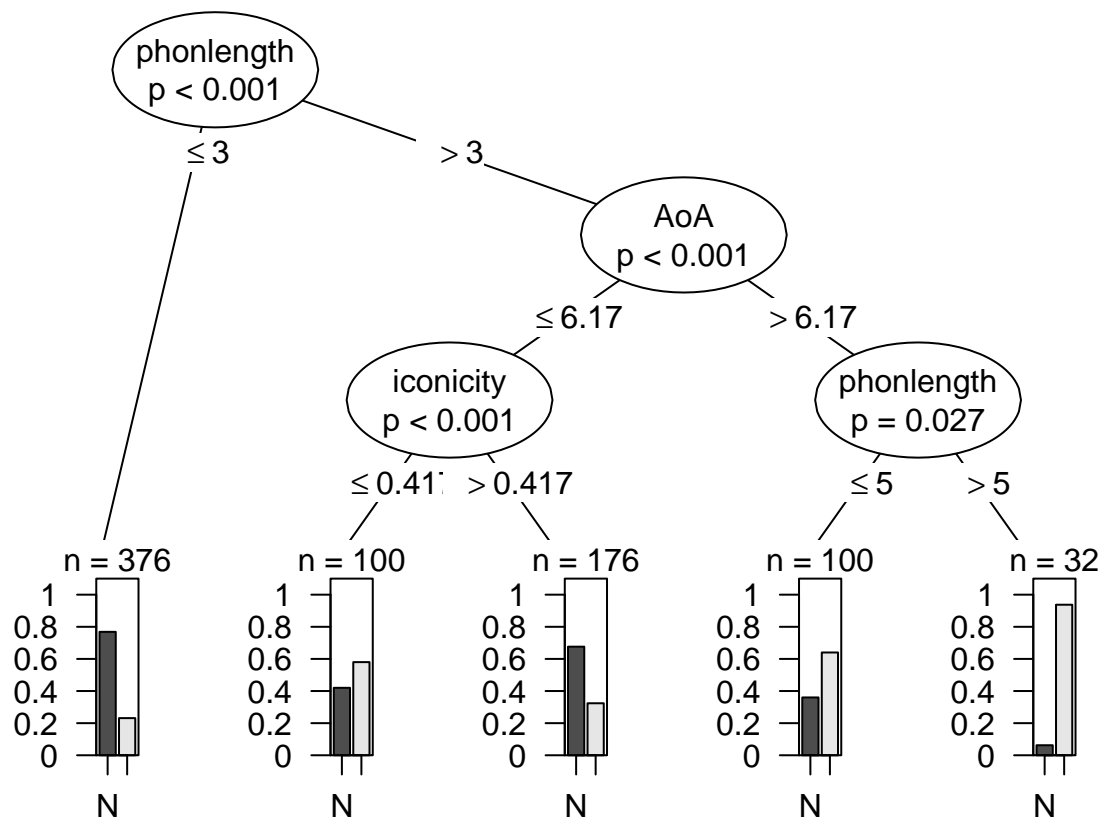
```
pdf("Results.pdf", width=6,height=3)
  plot(gx)
dev.off()
```

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

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 +
  subtlxzipf + 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("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
```

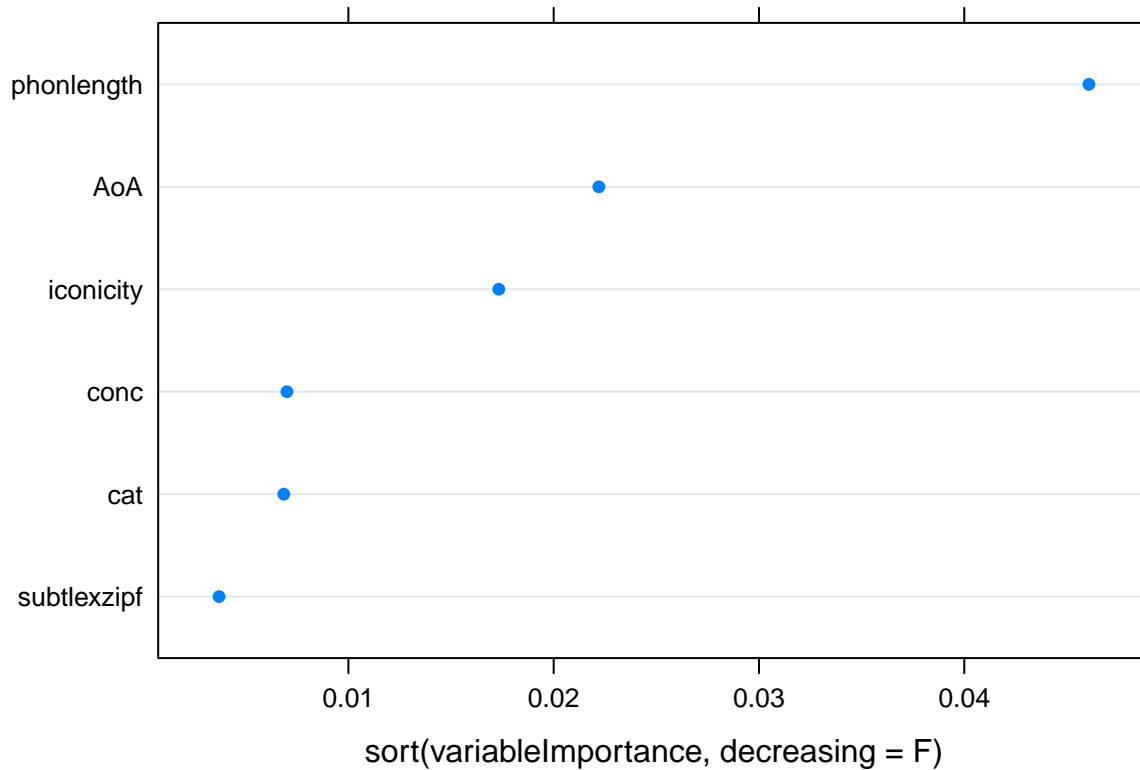
The decision tree above suggests that the biggest effect of iconicity is for longer words learned later in life. Next, we use random forests to estimate relative importance of measures:

```
set.seed(3289)
f = cforest(bor15.cat ~ phonlength + AoA +
  subtlxzipf + conc +
  iconicity + cat,
```

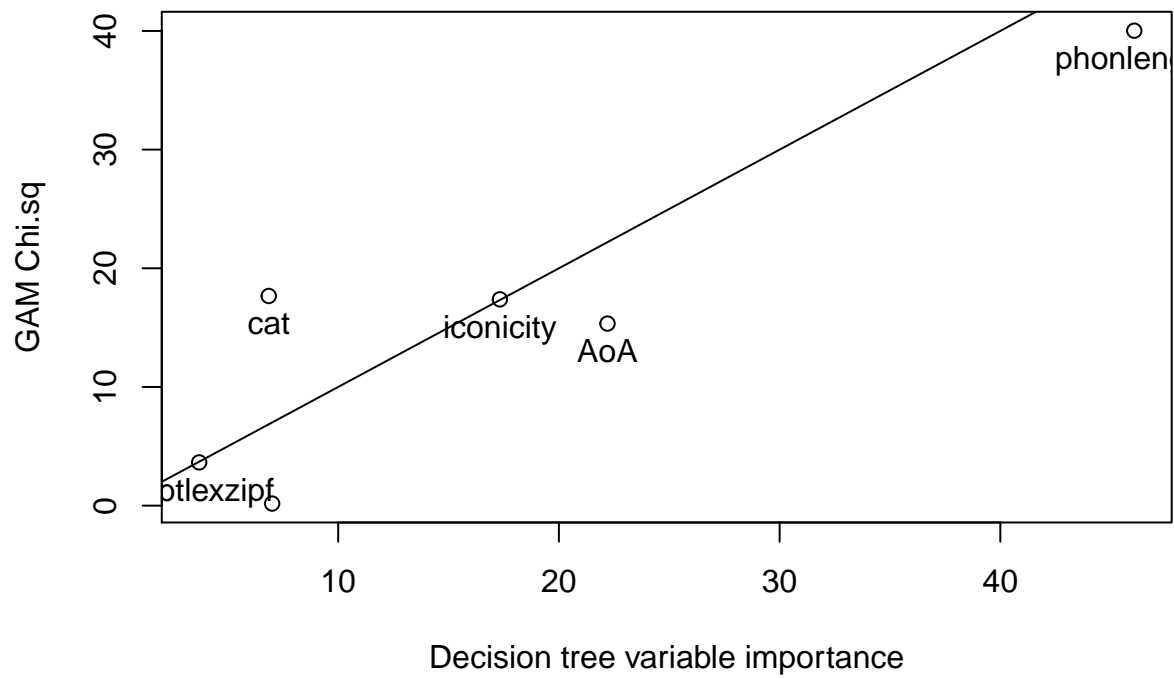
```
data = dataloan2)
variableImportance = varimp(f)
variableImportance
```

```
## 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)$s.table[1:length(ctreeOrder),3],
      xlab="Decision tree variable importance",
      ylab="GAM Chi.sq")
text(variableImportance[ctreeOrder]*1000,summary(m1)$s.table[1:length(ctreeOrder),3],ctreeOrder,pos=1)
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.