Cognitive influences in language evolution: English data

Introduction

This is the model code for Monaghan & Roberts, "Cognitive influences in language evolution: Psycholinguistic predictors of loan word borrowing". It takes data from the WOLD database of borrowing for 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)

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)
}
```

Load data

```
dataloan <- read.csv("../data/loanword8.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)

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

dataloan2$AoA_objscaled = scale(dataloan2$AoA_obj)
```

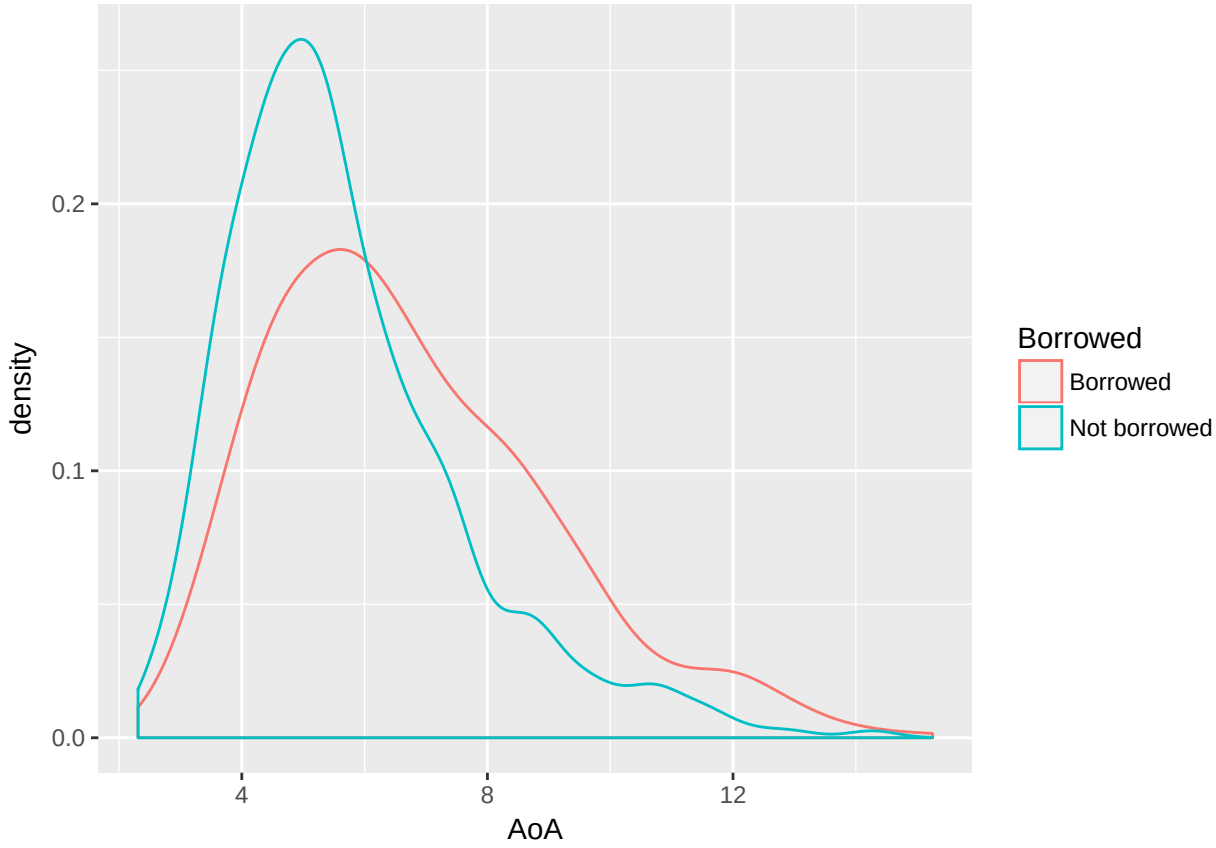
Identify Swadesh words:

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

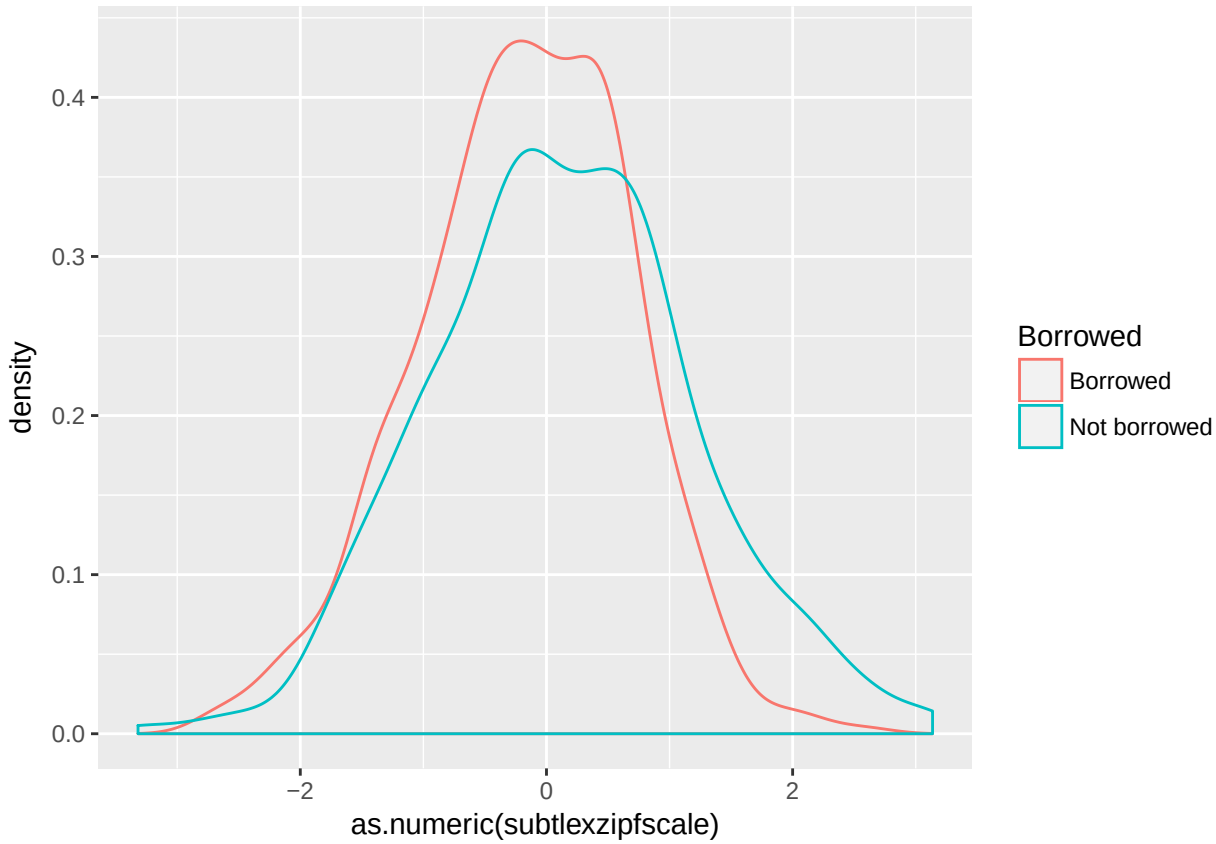
Plots

Raw data

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

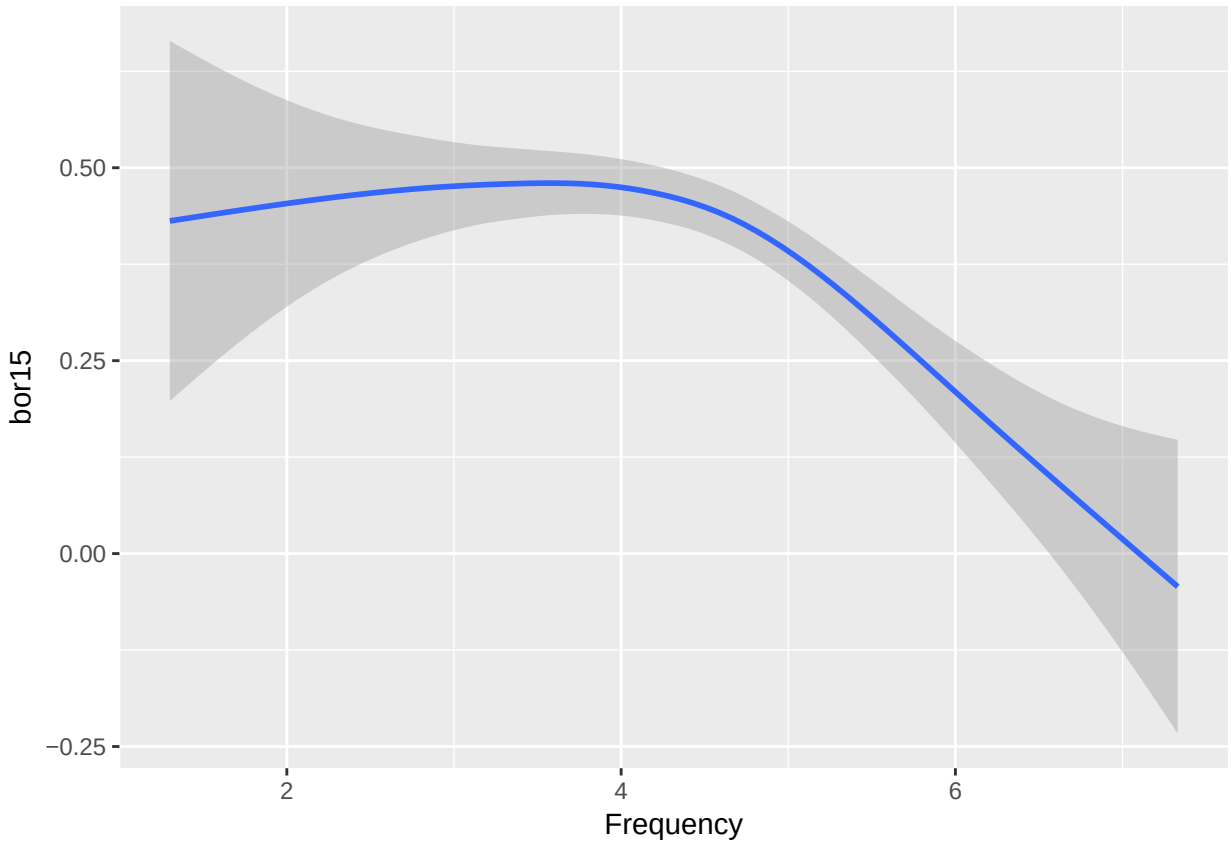



```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipfscale), colour=Borrowed)) +
  geom_density()
```



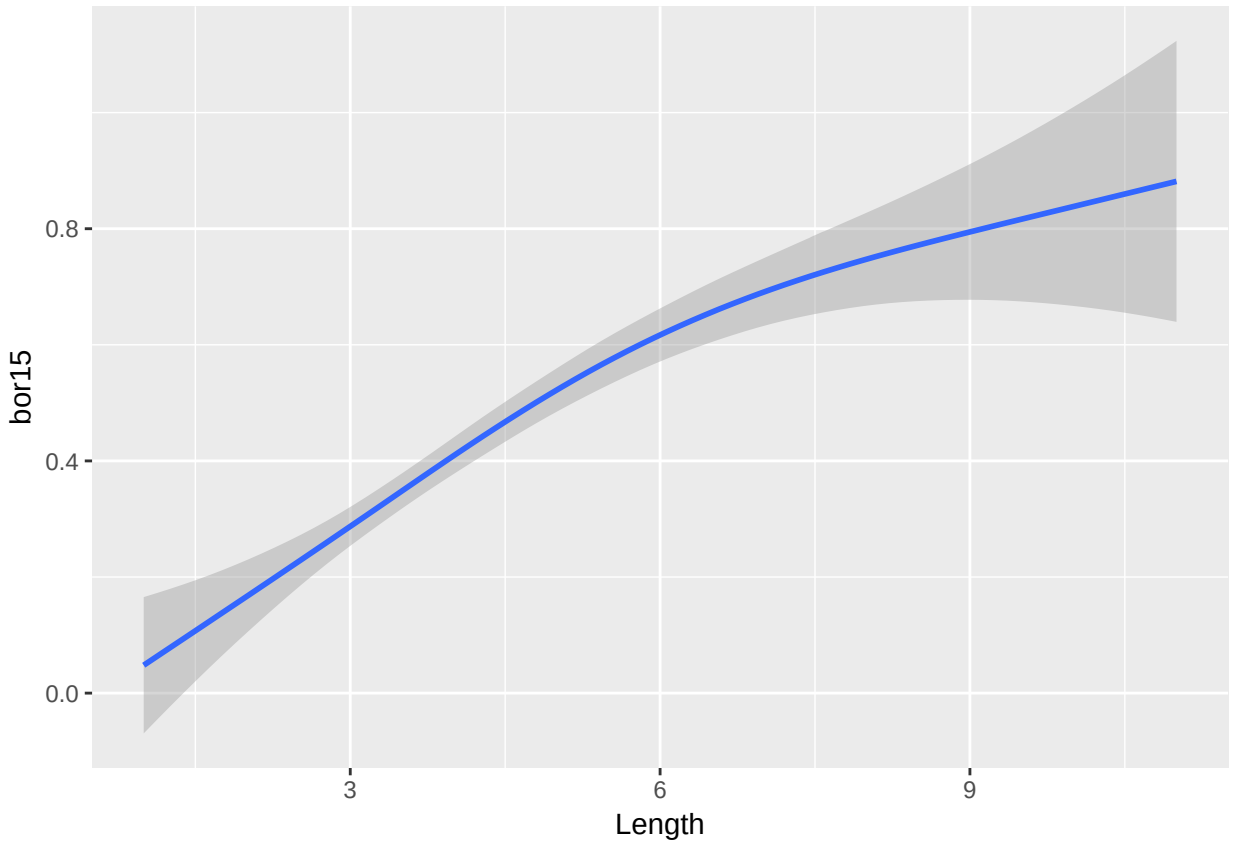

```
ggplot(dataloan2[!is.na(dataloan2$Borrowed),], aes(x=as.numeric(subtlexzipf), y=bor15)) +
  stat_smooth()+
  xlab("Frequency")
```

```
## `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:

```
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)
)
catx = catx[order(catx$mean, decreasing = T),]
catx$PoS = factor(catx$PoS, levels = catx[order(catx$mean, 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)

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

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

```
catx$mean= catx$mean*100
write.csv(catx, "../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
```

No significant improvement.

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)
```

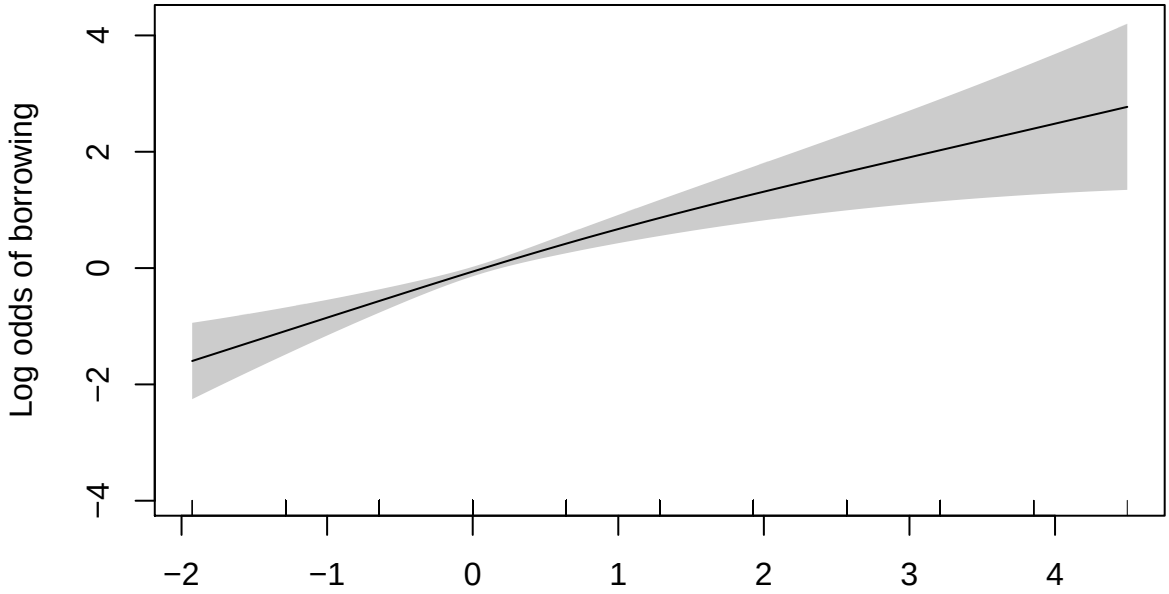
```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ s(phonlengthscale) + s(AoAscale) + s(subtlexzipfscale) +
##     s(concscale) + s(cat, bs = "re") + s(cat, phonlengthscale,
##     bs = "re") + s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale,
##     bs = "re") + s(cat, concscale, bs = "re")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

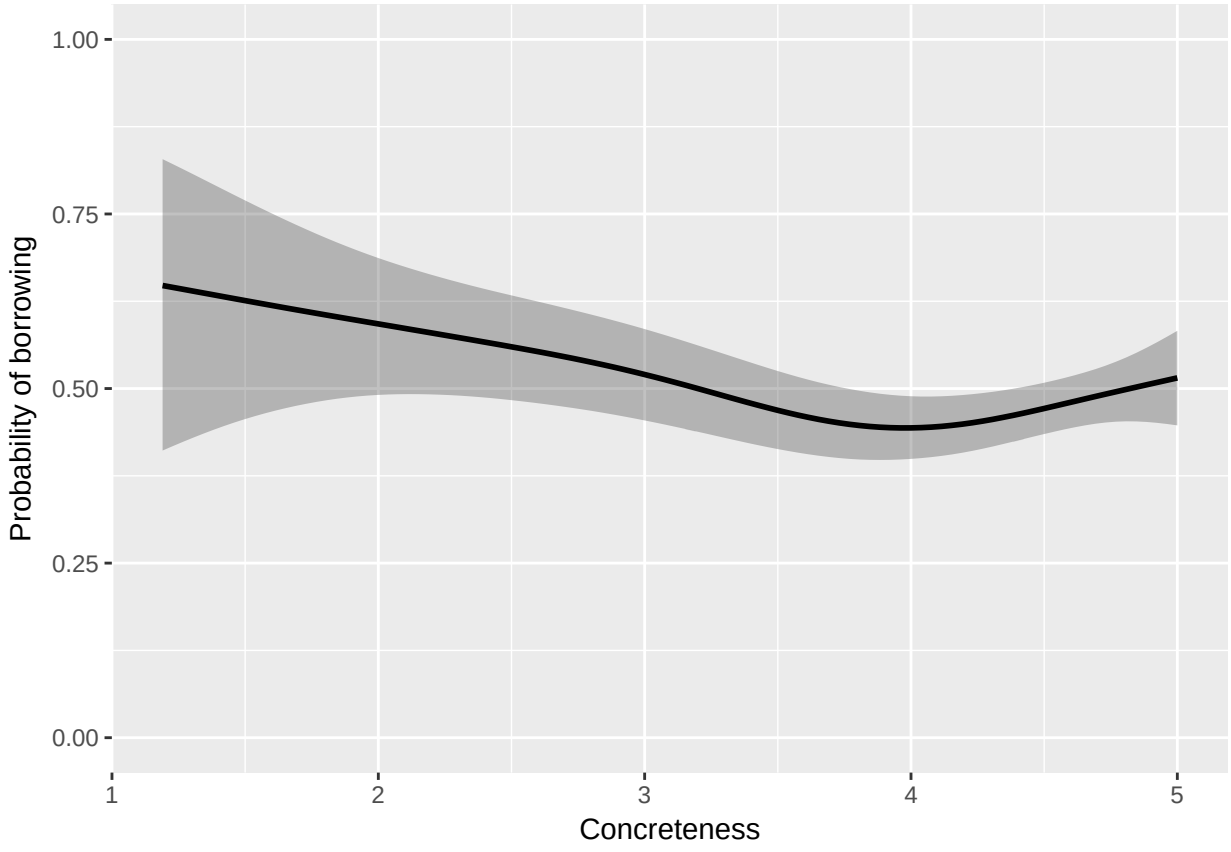
No significant improvement.

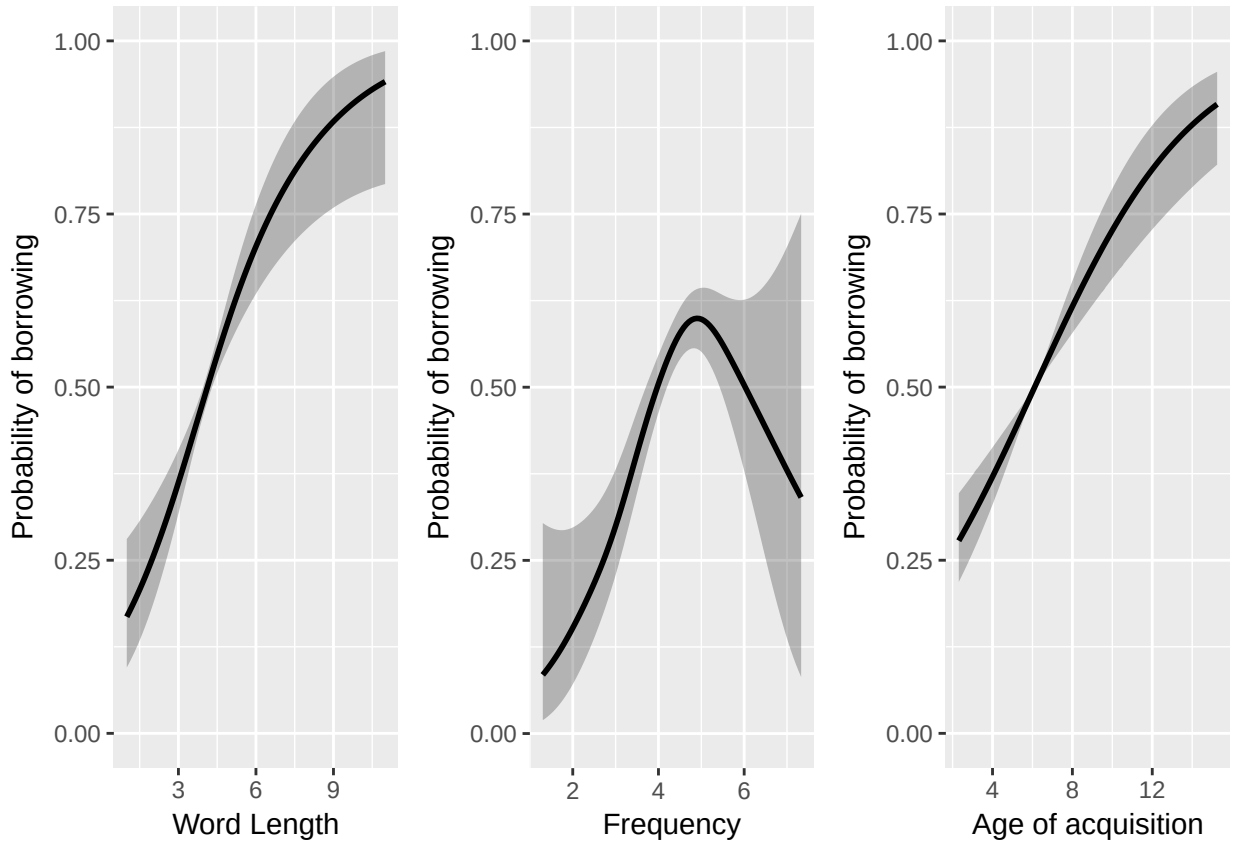
So no interactions are necessary.

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.

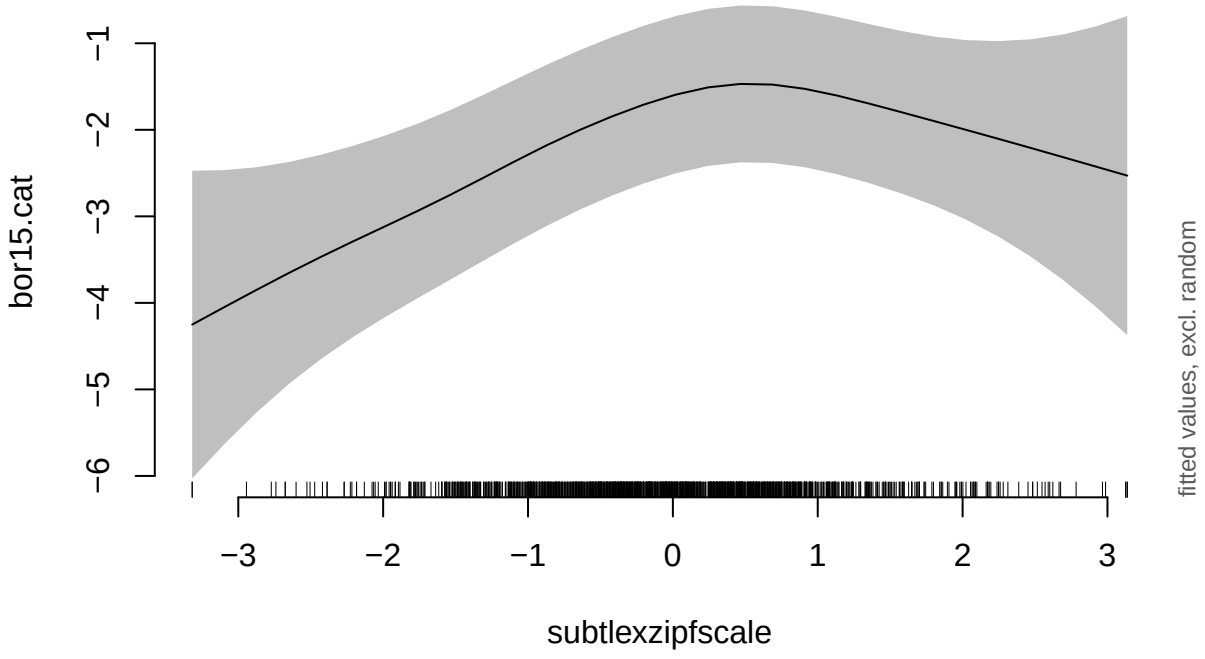


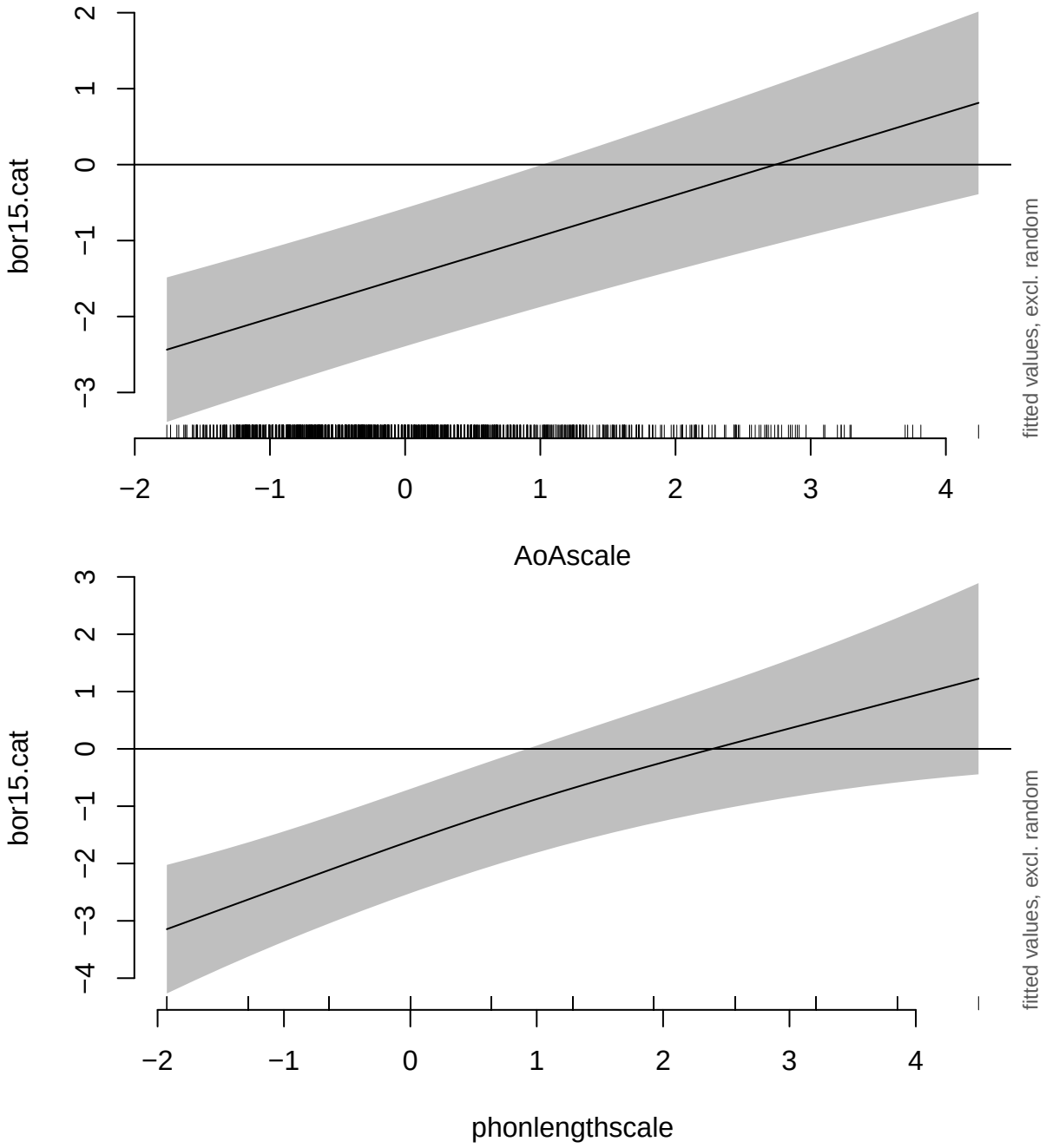


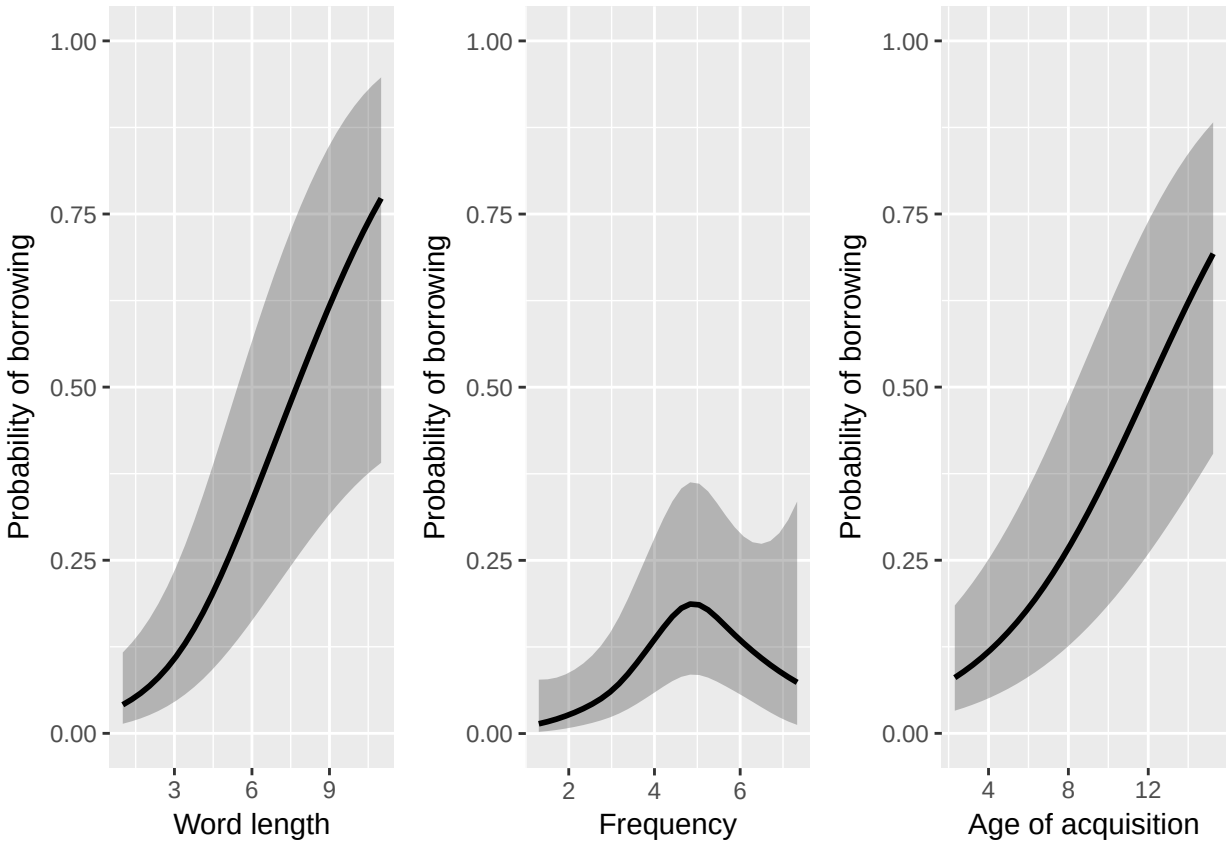



```
## pdf
##   2
```

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.




bor15.cat
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
AoAscale
fitted values, excl. random
bor15.cat
3
2
1
0
−1
−2
−3
−4
−2
−1
0
1
2
3
4
phonlengthscale
fitted values, excl. random


1.00
0.75
0.50
0.25
0.00
Probability of borrowing
3
6
9
Word length
2
4
6
Frequency
4
8
12
Age of acquisition

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

Objective measures of AoA

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

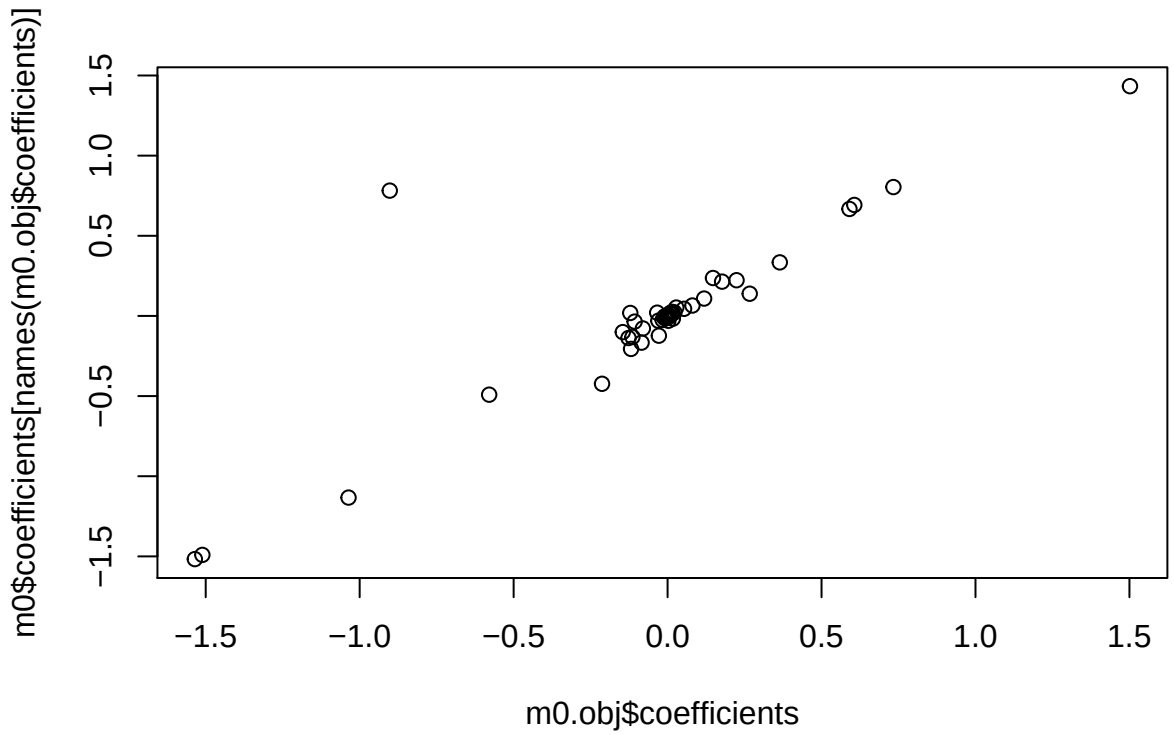
```
m0.obj = bam(bor15.cat ~
      s(phonlengthscale) +
      s(AoA_objscaled, k=3) +
      s(subtlexzipfscale) +
      s(concscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoA_objscaled,bs='re')+
      s(cat,subtlexzipfscale,bs='re')+
      s(cat,concscale,bs='re'),
    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`.

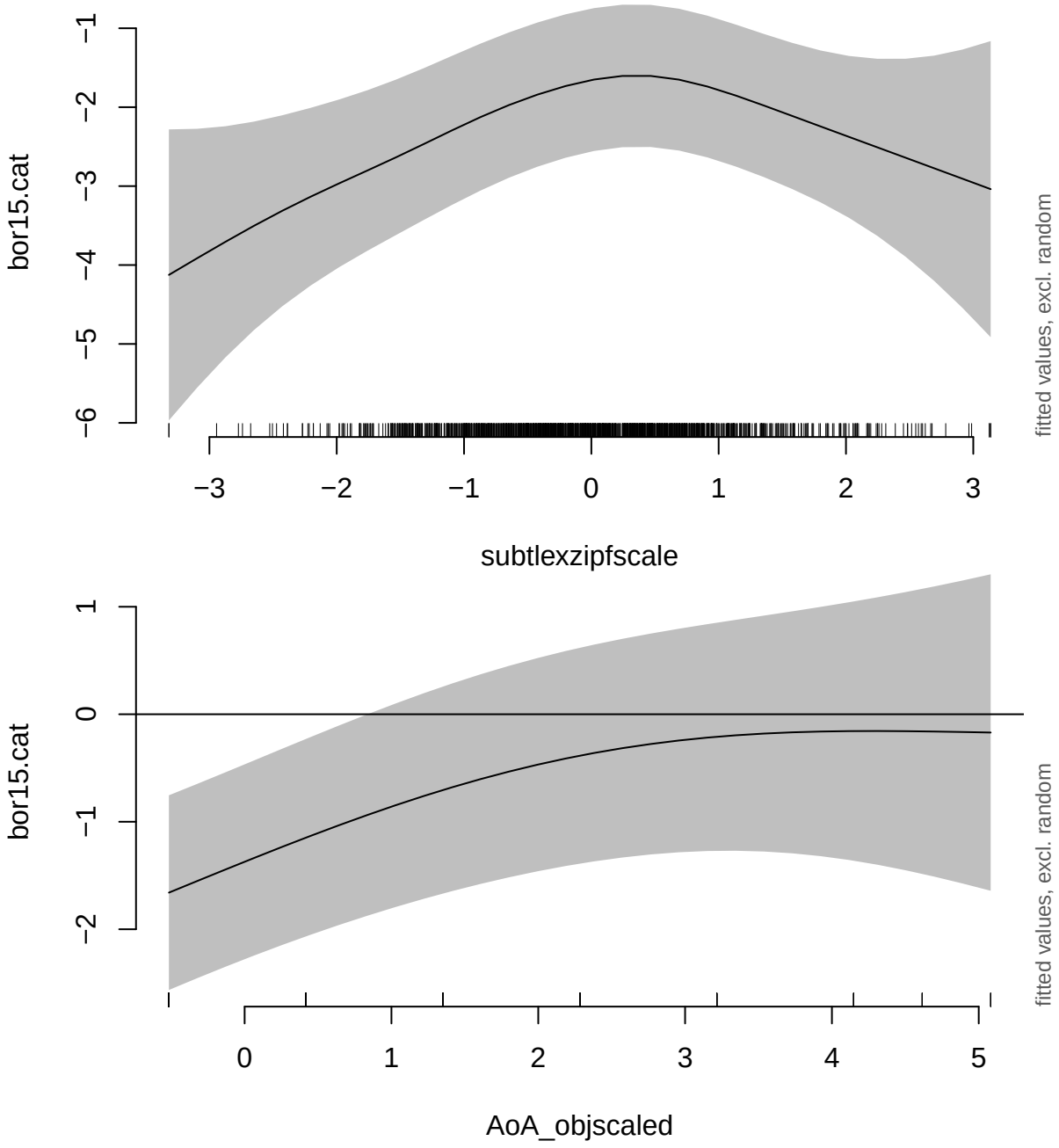
And chi squared terms are similar:

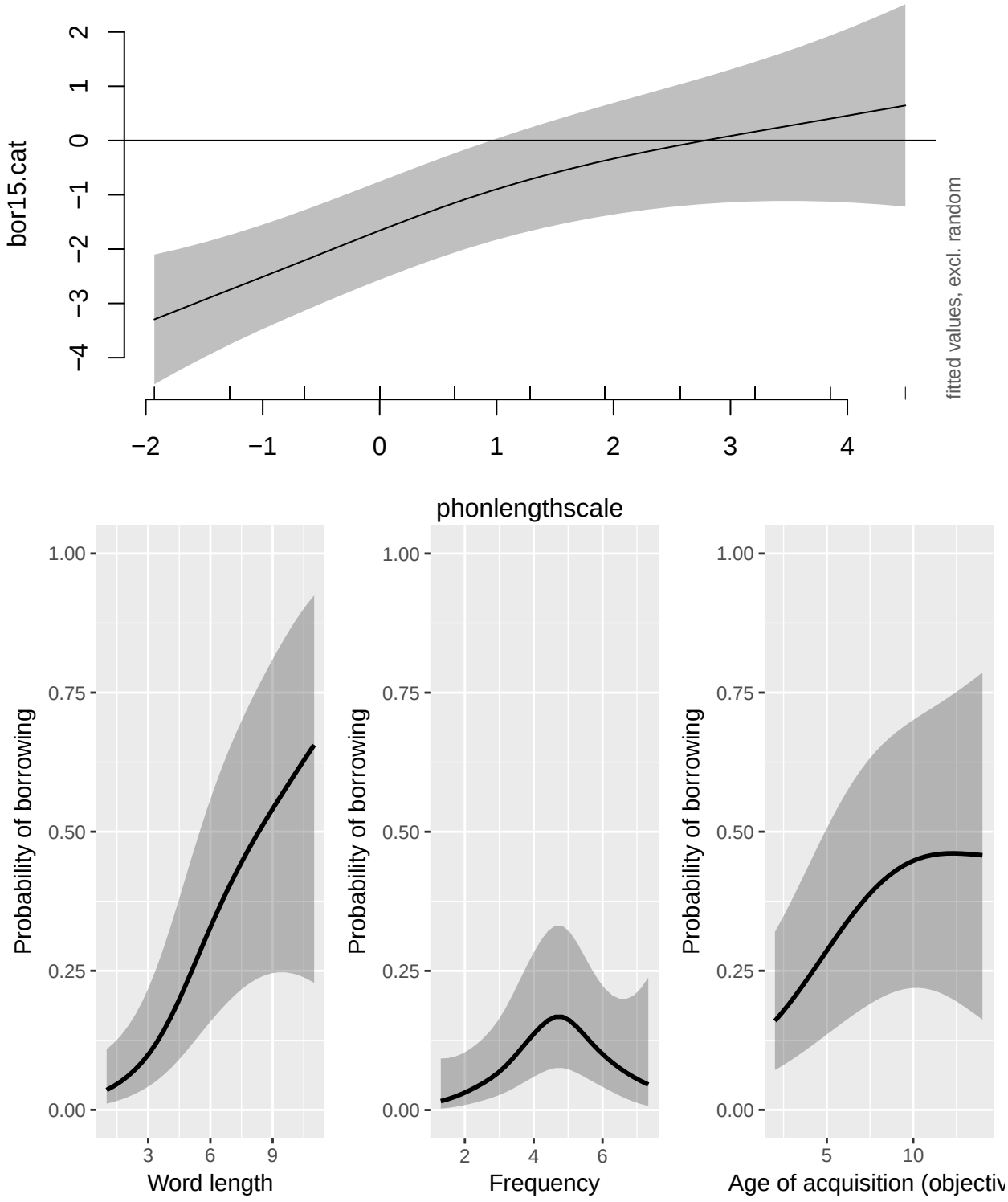
```
m0S = summary(m0)
m0.objS = summary(m0.obj)
cbind(m0=m0S$chi.sq,m0.obj=m0.objS$chi.sq)[1:4,]
```

```
##                             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.


bor15.cat
−1
−2
−3
−4
−5
−6
−3
−2
−1
0
1
2
3
subtlexzipfscale
fitted values, excl. random
bor15.cat
1
0
−1
−2
0
1
2
3
4
5
AoA_objscaled
fitted values, excl. random


bor15.cat
2
1
0
−1
−2
−3
−4
−2
−1
0
1
2
3
4
phonlengthscale
fitted values, excl. random
Probability of borrowing
1.00
0.75
0.50
0.25
0.00
3
6
9
Word length
Frequency
5
10
Age of acquisition (objectiv

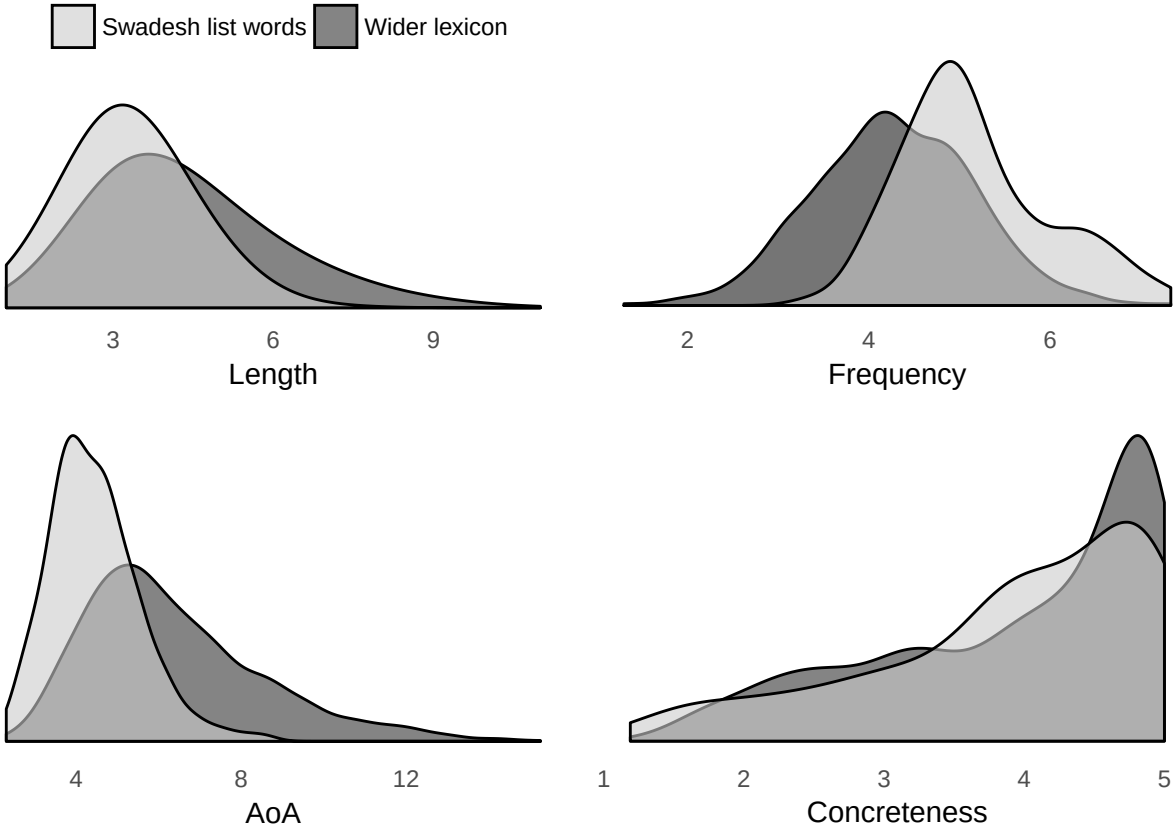
```
## pdf
##   2
```

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
```

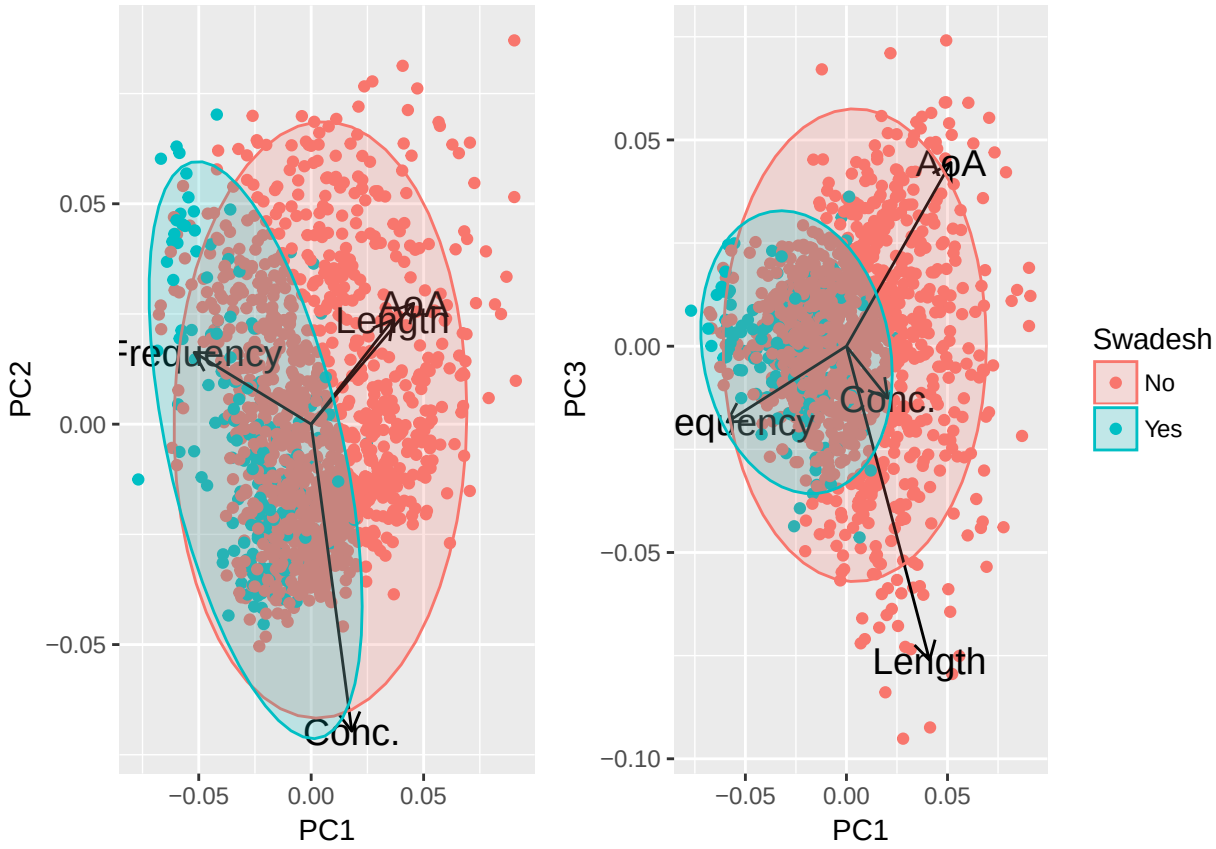
Try a PCA plot:

```
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),]
```

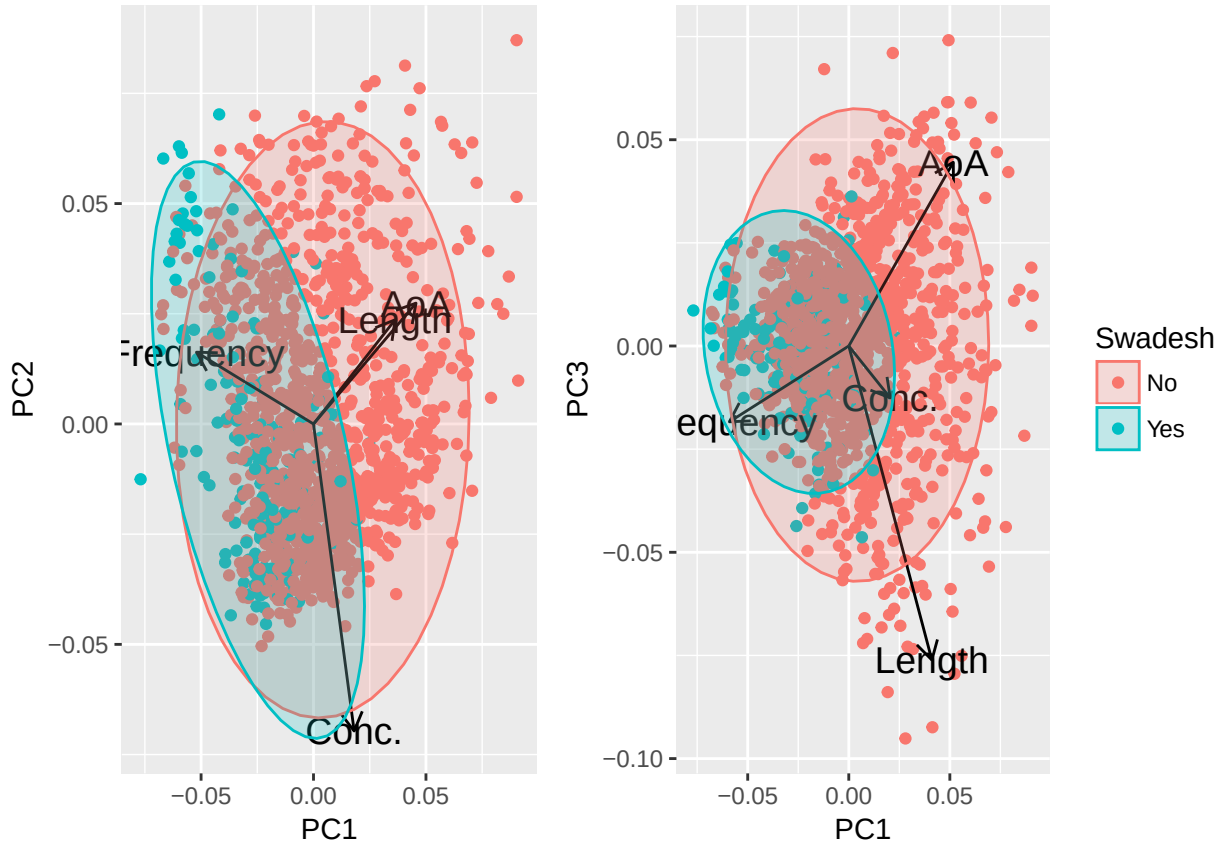
```
pc = prcomp(dx[, 1:4])

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
```