Age of acquisition and borrowing: French, Indonesian and Japanese

Introduction

Some data was available for French and Japanese. However, the number of datapoints for Japanese is very low, and the reliability of the loanword status for French is not good.

Load libraries

```
library(mgcv)
library(sjPlot)
library(lattice)
library(ggplot2)
library(gplots)
library(dplyr)
library(party)
library(lmtest)
library(gridExtra)
library(lme4)

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

```
load("../data/loanwords_French.Rdat")
load("../data/loanwords_Indonesian.Rdat")
load("../data/loanwords_Japanese.Rdat")
```

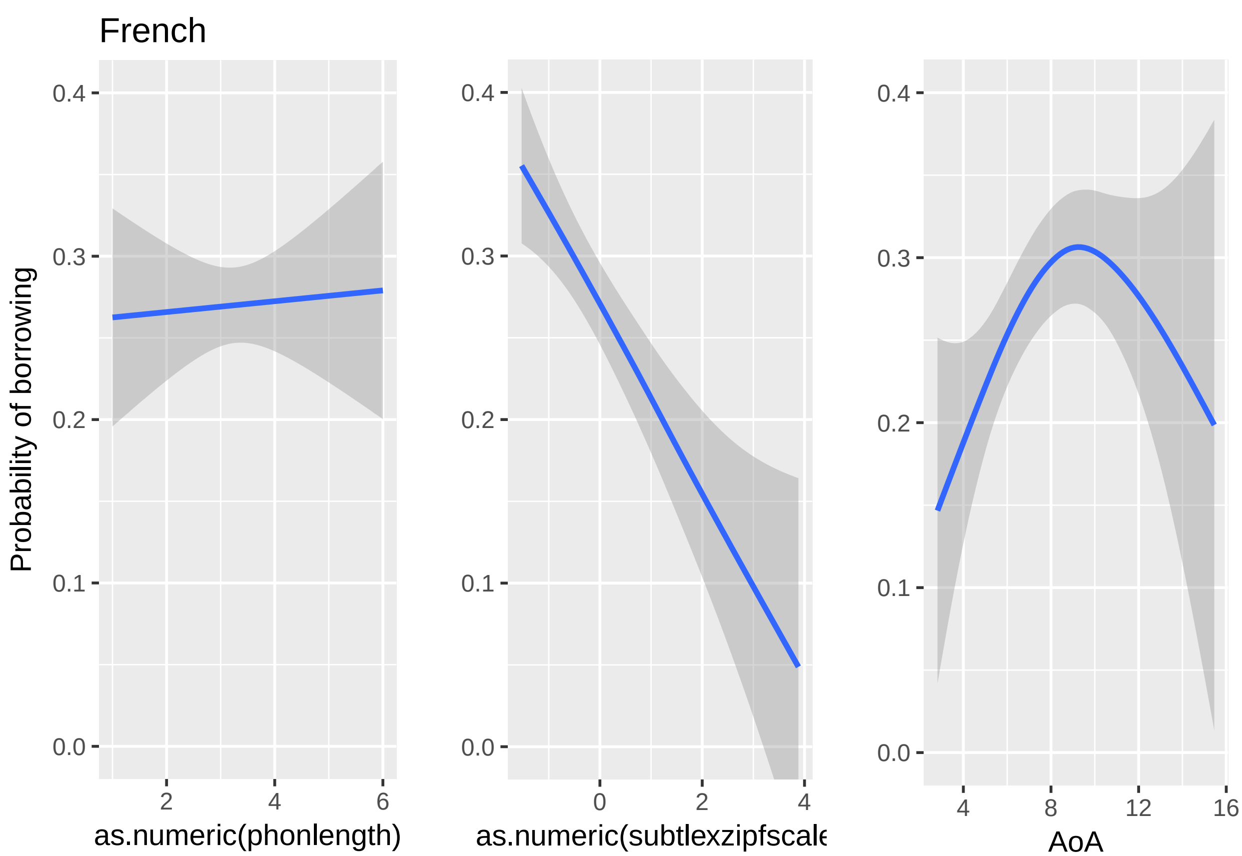
French

French data has 1422 datapoints. Note that the French data only includes monomorphemic words, and have a smaller length range than the Dutch or English data. The estimates of borrowing also come from lists of borrowed words from specific languages, rather than an expert judgement on each word in the data.

Raw data:

```
rd1 = ggplot(french, aes(as.numeric(subtlexzipfscale),bor15)) +
  geom_smooth() + ylab("") +
  coord_cartesian(ylim=c(0,0.4))+
  ggtitle("")
rd2 = ggplot(french, aes(AoA,bor15)) +
  geom_smooth()+ ylab("")+
  coord_cartesian(ylim=c(0,0.4))+
  ggtitle("")
rd3 = ggplot(french, aes(as.numeric(phonlength),bor15)) +
  geom_smooth(method = 'lm')+
  ylab("Probability of borrowing")+
  ggtitle("French")+
  coord_cartesian(ylim=c(0,0.4))
grid.arrange(rd3,rd1,rd2, nrow=1)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

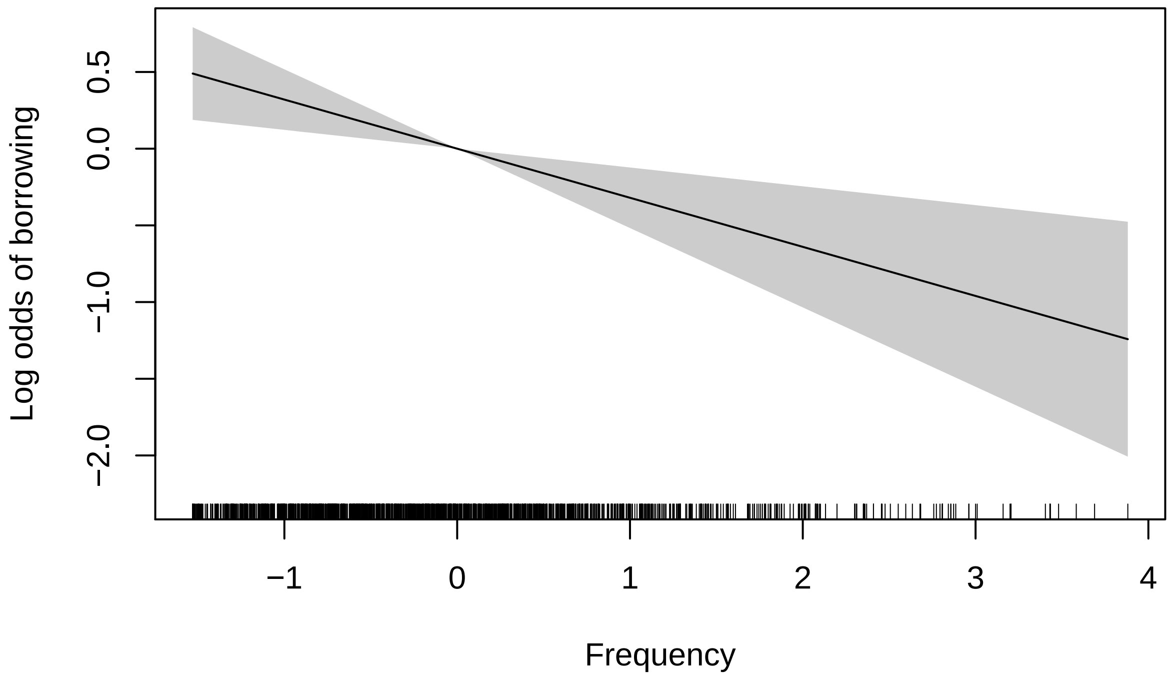


GAM:

```r
m0.french = bam(bor15.cat ~
      s(phonlengthscale,k=3) +
      s(AoAscale) +
      s(subtlexzipfscale) +
      s(cat,bs='re')+
      s(cat,phonlengthscale,bs='re')+
      s(cat,AoAscale,bs='re')+
      s(cat,subtlexzipfscale,bs='re'),
    data = french,
    family='binomial')
summary(m0.french)
```

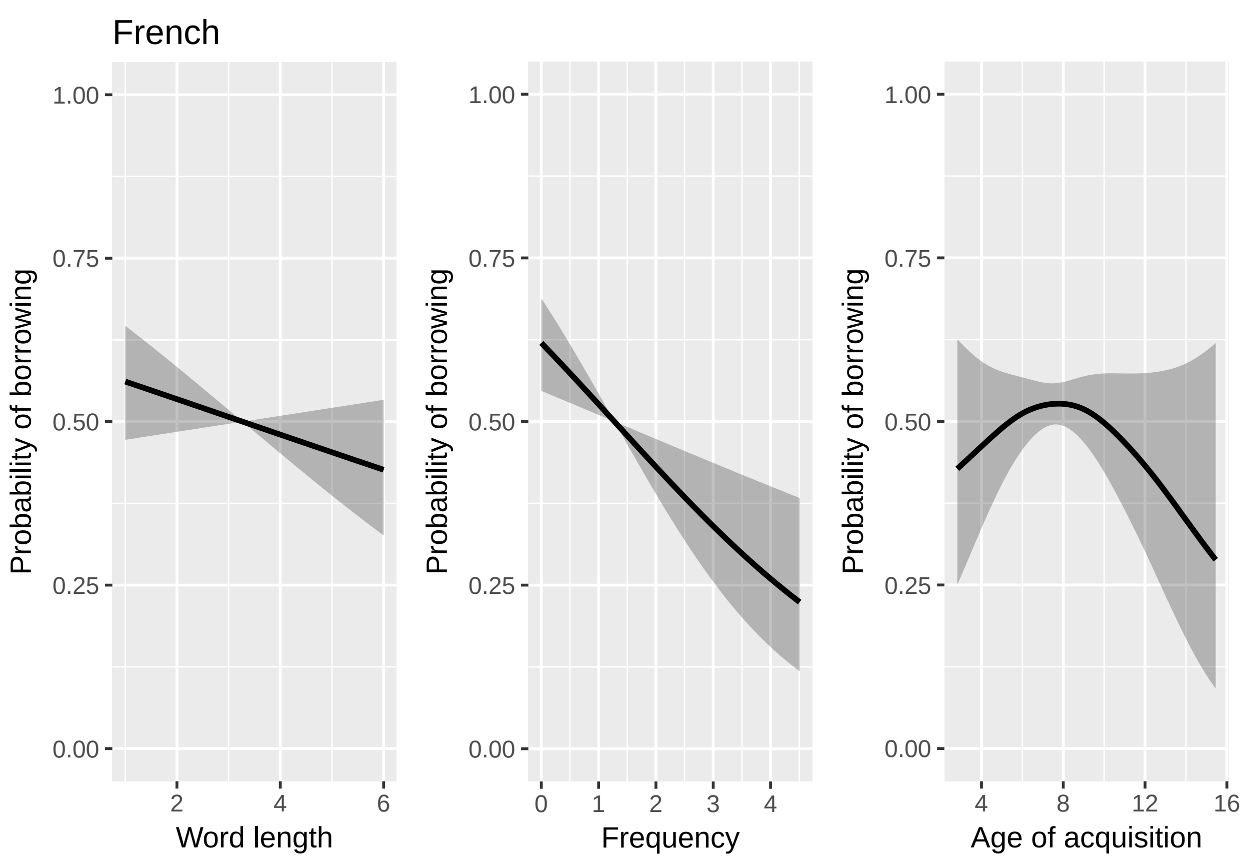
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## bor15.cat ~ s(phonlengthscale, k = 3) + s(AoAscale) + s(subtlexzipfscale) +
##     s(cat, bs = "re") + s(cat, phonlengthscale, bs = "re") +
##     s(cat, AoAscale, bs = "re") + s(cat, subtlexzipfscale, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.5126     0.2509  -6.029 1.65e-09 ***
## ---
## 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)      1.000012  1.000  1.901 0.167947
## s(AoAscale)             2.309782  2.942  4.622 0.193400
## s(subtlexzipfscale)     1.000006  1.000 10.522 0.001180 **
## s(cat)                  2.797584  8.000 16.090 0.000264 ***
## s(cat,phonlengthscale)  0.000056  8.000  0.000 0.432340
## s(cat,AoAscale)         1.227448  8.000  2.597 0.133447
## s(cat,subtlexzipfscale) 0.674600  8.000  0.944 0.294141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0357   Deviance explained = 3.86%
## fREML = 2020.4  Scale est. = 1         n = 1422
```

```r
px = plot.gam(m0.french,select=3, xlab="Frequency", ylab="Log odds of borrowing",shade = T)
```



Significant effect of frequency.

```
f1 = rescaleGam(px, 1, french$phonlengthscale,"Word length") + ggtitle("French")
f2 = rescaleGam(px, 2, french$AoAscale,"Age of acquisition") + ggtitle("")
f3 = rescaleGam(px, 3, french$subtlexzipfscale,"Frequency") + ggtitle("")
grid.arrange(f1,f3,f2, nrow=1)
```



```r
pdf("../results/graphs/French_ModelResults.pdf",
    height =3,width = 8)
grid.arrange(f1,f3,f2, nrow=1)
dev.off()
```

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

Indonesian

Data from Sianipar, van Groenestijn and Dijkstra (2016). There were only matches for 406 words, so we count ratings of 1 (definately borrowed) and 2 (probably borrowed) as borrowed, and 4 (little evidence) and 5 (no evidence) as not borrowed. We also use a simpler linear mixed effects model.

```r
m0.indonesian = glmer(bor15.cat ~
      ALL_Frequency_Mean.scaled +
      ALL_Concreteness_Mean.scaled+
      ALL_Valence_Mean.scaled+
      ALL_Arousal_Mean.scaled+
      ALL_Dominance_Mean.scaled+
      ALLPredictability_Mean.scaled+
        (1|cat),
    data = indonesian,
      family="binomial")
summary(m0.indonesian)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## bor15.cat ~ ALL_Frequency_Mean.scaled + ALL_Concreteness_Mean.scaled +
##     ALL_Valence_Mean.scaled + ALL_Arousal_Mean.scaled + ALL_Dominance_Mean.scaled +
##     ALLPredictability_Mean.scaled + (1 | cat)
##    Data: indonesian
##
##      AIC      BIC   logLik deviance df.resid
##    491.7    523.8   -237.9    475.7      398
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9025 -0.6994 -0.4971  1.0343  2.7186
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  cat    (Intercept) 0.2967   0.5447
## Number of obs: 406, groups:  cat, 3
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.23363    0.36779  -3.354 0.000796 ***
## ALL_Frequency_Mean.scaled     -0.28162    0.12273  -2.295 0.021754 *
## ALL_Concreteness_Mean.scaled   0.03402    0.15084   0.226 0.821564
## ALL_Valence_Mean.scaled        0.20773    0.15259   1.361 0.173408
## ALL_Arousal_Mean.scaled        0.37260    0.13349   2.791 0.005250 **
## ALL_Dominance_Mean.scaled      0.32783    0.14796   2.216 0.026713 *
## ALLPredictability_Mean.scaled  0.10298    0.12364   0.833 0.404884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ALL_F_ ALL_C_ ALL_V_ ALL_A_ ALL_D_
## ALL_Frqn_M. -0.011
```

```
## ALL_Cncr_M. -0.193  0.052
## ALL_Vlnc_M. -0.022 -0.122  0.050
## ALL_Arsl_M. -0.042 -0.134 -0.338  0.260
## ALL_Dmnn_M. -0.015 -0.255 -0.160 -0.482 -0.017
## ALLPrdct_M. -0.025 -0.081 -0.040 -0.162  0.210 -0.106
```

The model above contains a number of affective parameters. Taking these out removes the effect for frequency:

```r
m1.indonesian = glmer(bor15.cat ~
      ALL_Frequency_Mean.scaled +
      ALL_Concreteness_Mean.scaled +
        (1|cat),
    data = indonesian,
      family="binomial")
summary(m1.indonesian)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## bor15.cat ~ ALL_Frequency_Mean.scaled + ALL_Concreteness_Mean.scaled +
##     (1 | cat)
##    Data: indonesian
##
##      AIC      BIC   logLik deviance df.resid
##    503.6    519.6   -247.8    495.6      402
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0634 -0.7346 -0.5184  1.2535  2.2795
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  cat    (Intercept) 0.3441   0.5866
## Number of obs: 406, groups:  cat, 3
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.22302    0.38427  -3.183  0.00146 **
## ALL_Frequency_Mean.scaled    -0.08345    0.10804  -0.772  0.43987
## ALL_Concreteness_Mean.scaled  0.20620    0.13413   1.537  0.12422
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ALL_F_
## ALL_Frqn_M. -0.027
## ALL_Cncr_M. -0.203 -0.007
```

Japanese

Note that the japanese data only has 193 datapoints, so we'll use model comparison with a standard liner model to test for effects.

```
m0.japanese = glm(bor15.cat ~
      1,
    data = jap,
    family='binomial')

m1.japanese = glm(bor15.cat ~
      phonlengthscale,
    data = jap,
    family='binomial')
anova(m0.japanese, m1.japanese)
```

```
## Analysis of Deviance Table
##
## Model 1: bor15.cat ~ 1
## Model 2: bor15.cat ~ phonlengthscale
##   Resid. Df Resid. Dev Df Deviance
## 1       192     266.39
## 2       191     266.19  1  0.20221
```

No improvement for length.

```
m2.japanese = glm(bor15.cat ~
      1 + subtlexzipfscale,
    data = jap,
    family='binomial')
lrtest(m0.japanese, m2.japanese)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ 1
## Model 2: bor15.cat ~ 1 + subtlexzipfscale
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   1 -133.19
## 2   2 -132.49  1 1.4014     0.2365
```

No improvement for frequency.

```
m3.japanese = glm(bor15.cat ~
      1 + AoAscale,
    data = jap,
    family='binomial')
lrtest(m0.japanese, m3.japanese)
```

```
## Likelihood ratio test
##
## Model 1: bor15.cat ~ 1
## Model 2: bor15.cat ~ 1 + AoAscale
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   1 -133.19
## 2   2 -132.19  1 2.0028      0.157
```

No improvement for age of acquisition.

```r
summary(m3.japanese)
```

```
##
## Call:
## glm(formula = bor15.cat ~ 1 + AoAscale, family = "binomial",
##     data = jap)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.484  -1.224   1.012   1.116   1.289
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.1573     0.1452   1.084    0.278
## AoAscale      0.2060     0.1464   1.407    0.159
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 266.39  on 192  degrees of freedom
## Residual deviance: 264.39  on 191  degrees of freedom
## AIC: 268.39
##
## Number of Fisher Scoring iterations: 4
```