Language in economics and accounting research: controlling for linguistic history

Giorgio Gotti, Seán G. Roberts, Marco Fasan & Cole Robertson

Contents

Introduction 1
- Variables . . . 2

Load libraries 3

Load data 3

Modelling AAM with a gamma distribution 6

Mixed effects modelling 11

Summary 18

Other effects 19

Alternative tests 21
- Gaussian distribution model . . . 22
- Model with average FTR per family . . . 30
- Decision tree . . . 33
- Random slopes . . . 35
- Phylogenetic test . . . 39
- OLS with clustered errors . . . 42

Introduction

This document shows the statistical procedure and R code for testing the relationship between strong/weak FTR and accrual based earnings management (AAM), with and without controls for language family. The code and data are available on github: https://github.com/seannyD/FTRAccountingStudy

We start by describing the variables, then showing how the data was loaded and linked to the language family data. We then demonstrate that the AAM is best modelled with a gamma distribution (see later in the document for the same test with gaussian distributions).

The mixed effecst modelling section runs the main statistical models with and without controls for language family.

The next sections demonstrate a series of alterantive tests, including:

- Assuming a gaussian distribution
- A decision tree analysis that takes into account non-linear effects and interactions.
- A visualisation of differences between language families
- A test that uses continuous historical distances from a phylogenetic tree

- An OLS regression with cluster robust standard errors

Variables

Each observation in the data is a single company within a particular country.

- `AAM`: accrual-based earnings management, following Kothari et al. (2005).
- `strongftr`: Whether the main language of the country has a 'strong' Future Tense Reference system, according to Chen (2013).
- `mainLanguageFamily` (constructed below): The language family of the main language(s) in the company's country.

Country-level economic predictors:

- `invpro`: Investor protection score, based on the anti-director index from Djankov et al. (2008)
- `ggr`: Country GDP growth rate

Country-level cultural predictors:

- `pd`: Power distance index, based on Hofstede (2001)
- `indiv`: Individualism/collectivism score, based on Hofstede (2001)
- `mas`: Masculinity/femininity score, based on Hofstede (2001)
- `ua`: Uncertainty avoidance score, based on Hofstede (2001)
- `lto`: Long-/short-term orientation score, based on Hofstede (2001);
- `indul`: Indulgence, based on Hofstede (2001);

Company-level economic predictors:

- `SIZE`: Company size, measured as the natural logarithm of total assets adjusted for inflation rate
- `BTM`: Company book value of common equity divided by common value of equity;
- `LEV`: Company leverage, measured as short- and long- term debt divided by total assets
- `ROA`: Company return on assets, measured as income before extraordinary items divided by total assets
- `MEET`: Dummy variable that takes one for firm-year observations with actual annual EPS greater than or equal to consensus analyst earnings forecast, zero otherwise.
- `LOSS`: Dummy variable that takes one for firm-year observations with negative income before extraordinary items, zero otherwise.

Load libraries

```
library(lme4)
library(sjPlot)
library(REEMtree)
library(rpart)
library(rpart.plot)
library(MASS)
library(ggplot2)
library(RColorBrewer)
library(MCMCglmm)
library(ape)
library(caper)
library(stargazer)
library(dplyr)
library(lattice)
```

Load data

```
d = read.csv("../data/clean/data.csv",
             fileEncoding = "utf-8",
             encoding = 'utf-8')
```

Match each country to its main language and language family:

```
countryMainLanguageFamily =
  read.csv("../data/raw/CountryMainLanguageToLanguageFamily.csv",
    stringsAsFactors = F)

d$mainLanguageFamily =
  countryMainLanguageFamily[
    match(as.character(d$loc),
          countryMainLanguageFamily$Country.Code),
    ]$Family
```

Remove countries with many main language families:

```
d$CountryHasManyMainLanguages = countryMainLanguageFamily[
  match(as.character(d$loc),
        countryMainLanguageFamily$Country.Code),
  ]$ManyLanguages=="Y"
d2 = d[!d$CountryHasManyMainLanguages,]

d2 = d2[!is.na(d2$AAM),]
```

Remove cases with missing data:

```
keyVar = c("invpro","pd","indiv","mas",
  "ua","lto","indul","ggr","SIZE",
  "BTM","LEV","ROA","MEET","LOSS")
d2 = d2[complete.cases(d2[,keyVar,]),]
```

Table of languages:

```r
data.frame(
  tapply(d2$strongftr,as.character(d2$loc),head,n=1)
)
```

```
##     tapply.d2.strongftr..as.character.d2.loc...head..n...1.
## AUS                                                       1
## AUT                                                       0
## BEL                                                       0
## BGR                                                       1
## BRA                                                       0
## CAN                                                       1
## CHE                                                       0
## CHL                                                       1
## CHN                                                       0
## COL                                                       1
## CZE                                                       1
## DEU                                                       0
## DNK                                                       0
## EGY                                                       1
## ESP                                                       1
## FIN                                                       0
## FRA                                                       1
## GBR                                                       1
## GRC                                                       1
## HKG                                                       0
## HUN                                                       1
## IDN                                                       0
## IND                                                       1
## IRL                                                       1
## ITA                                                       1
## JOR                                                       1
## JPN                                                       0
## KOR                                                       1
## LTU                                                       1
## LUX                                                       0
## LVA                                                       1
## MAR                                                       1
## MEX                                                       1
## MYS                                                       0
## NLD                                                       0
## NOR                                                       0
## NZL                                                       1
## PAK                                                       1
## PER                                                       1
## PHL                                                       1
## POL                                                       1
## PRT                                                       1
## ROU                                                       1
## RUS                                                       1
## SGP                                                       1
## SWE                                                       0
## THA                                                       1
## TUR                                                       1
## TWN                                                       0
```

```
## USA                                                            1
```

Convert to factors:

```
d2$mainLanguageFamily = factor(d2$mainLanguageFamily)
d2$MEET = factor(d2$MEET)
d2$LOSS = factor(d2$LOSS)
d2$strongftr = factor(d2$strongftr)
```

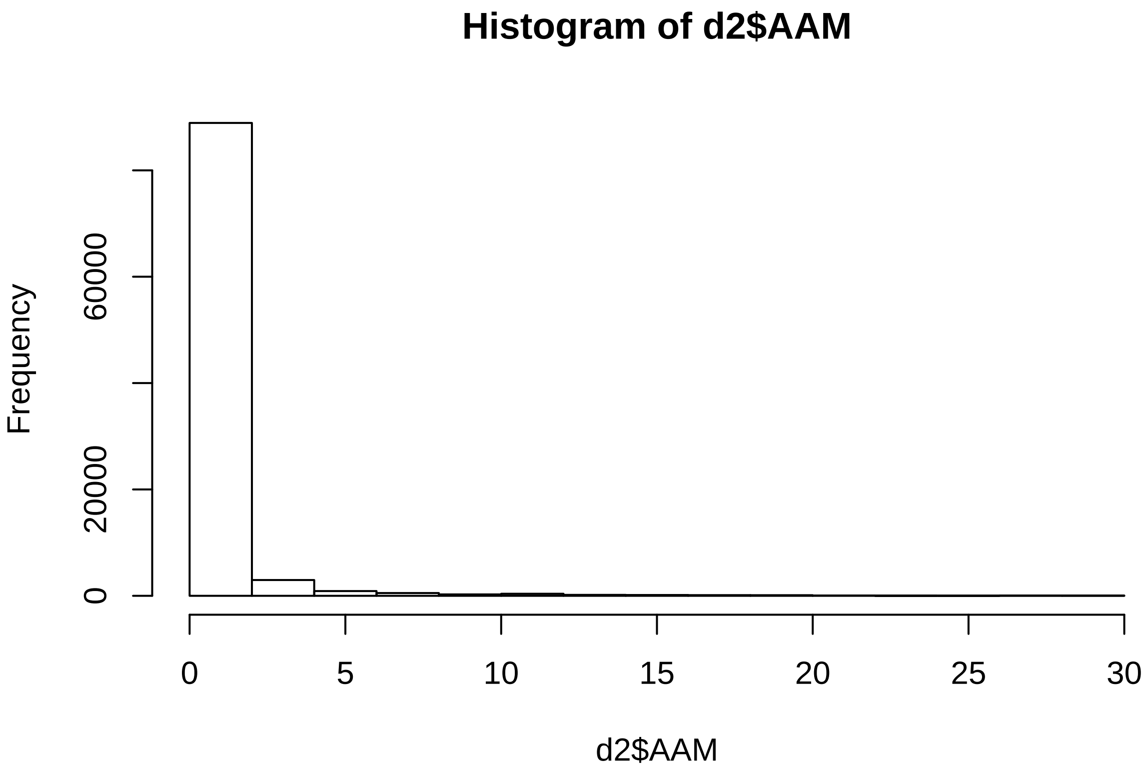
Scale and center varaibles:

```
d2Orig = d2
# Take log of AAM
d2$logAAM = log(1+d2$AAM)
# Scale and center continuous variables
for(v in c("pd",'indiv','mas',
           'ua','lto','indul','ggr',
           'SIZE',"BTM","LEV","ROA")){
  d2[,v] = scale(d2[,v])
}
d2$AAM.scaled = scale(d2$AAM)
```

Modelling AAM with a gamma distribution

The distribution of the AAM variable is highly skewed and values below zero are not permitted:

```
hist(d2$AAM)
```



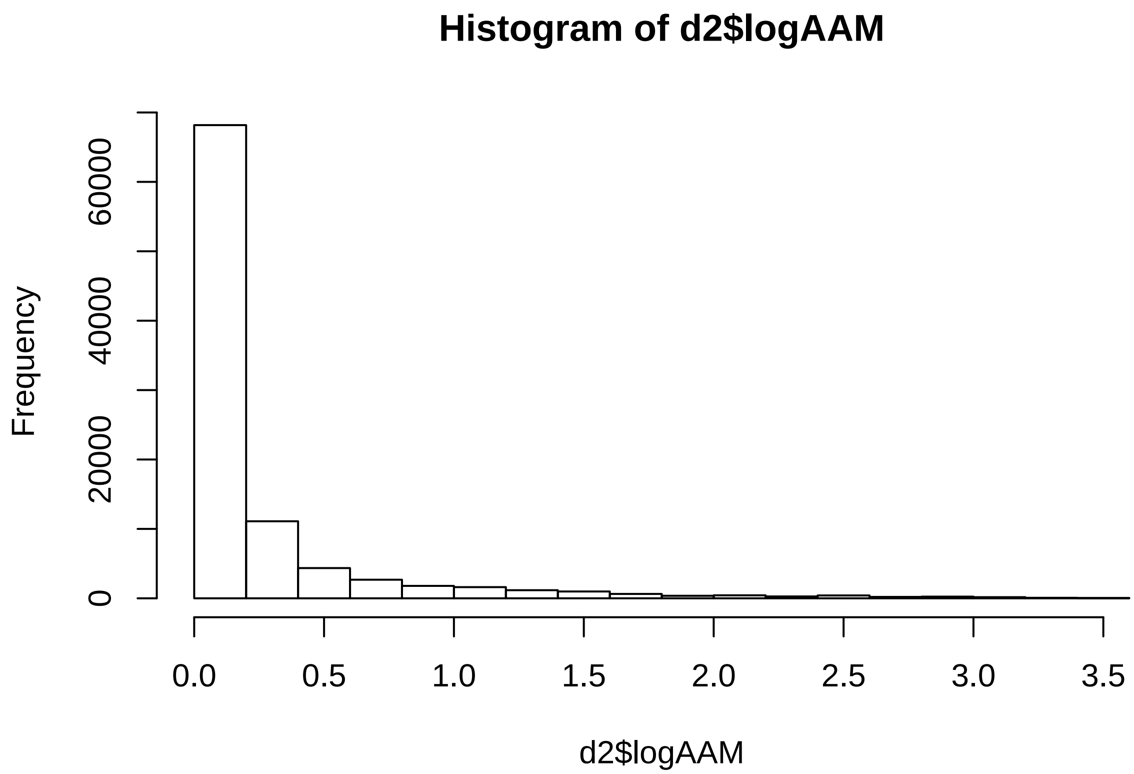

```
normalDist = rnorm(n=length(d2$AAM),
                   mean = mean(d2$AAM),
                   sd = sd(d2$AAM))
ks.test(d2$AAM,normalDist)
```

```
## Warning in ks.test(d2$AAM, normalDist): p-value will be approximate in the
## presence of ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  d2$AAM and normalDist
## D = 0.38823, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

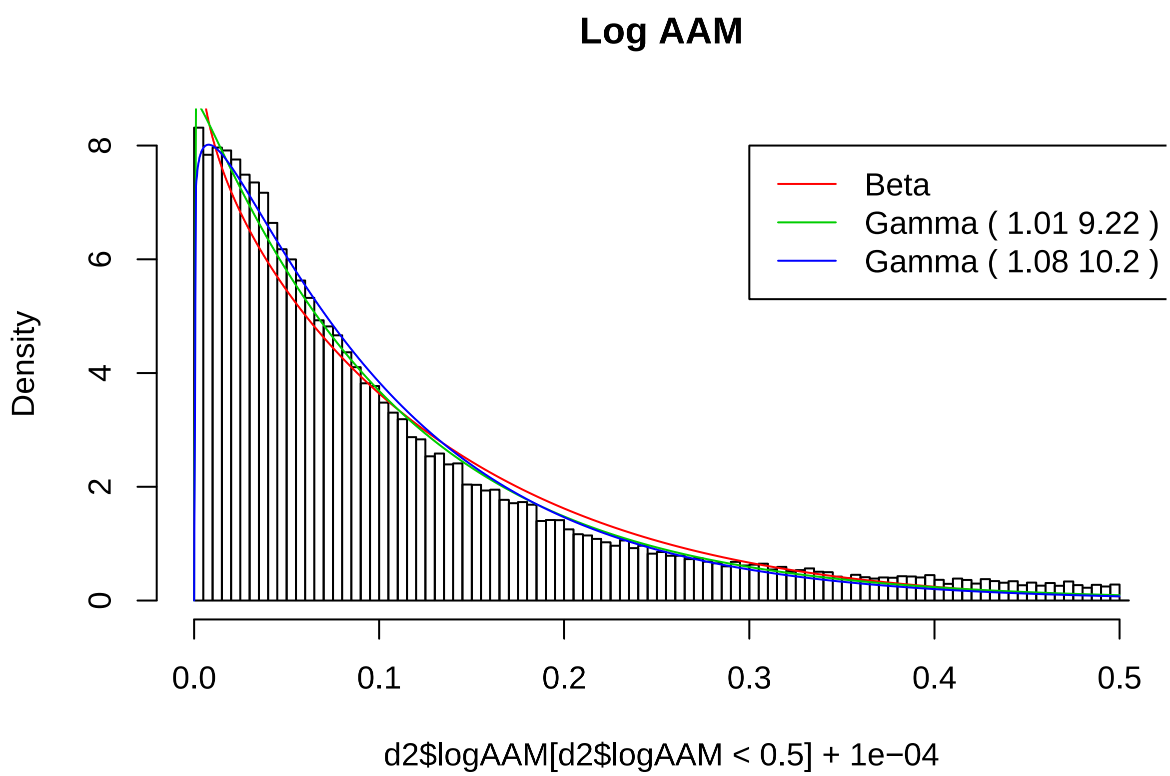
Even the log-transformed variable is skewed:

```
hist(d2$logAAM)
```



If we assume a Gaussian distribution, the statistical models below produce very poor fits, as can be seen in this QQ plot below:

We can compare how Beta and Gamma distributions fit the log data:



The Gamma distribution seems to fit best. We can compare the QQ plot of the model (fit below):

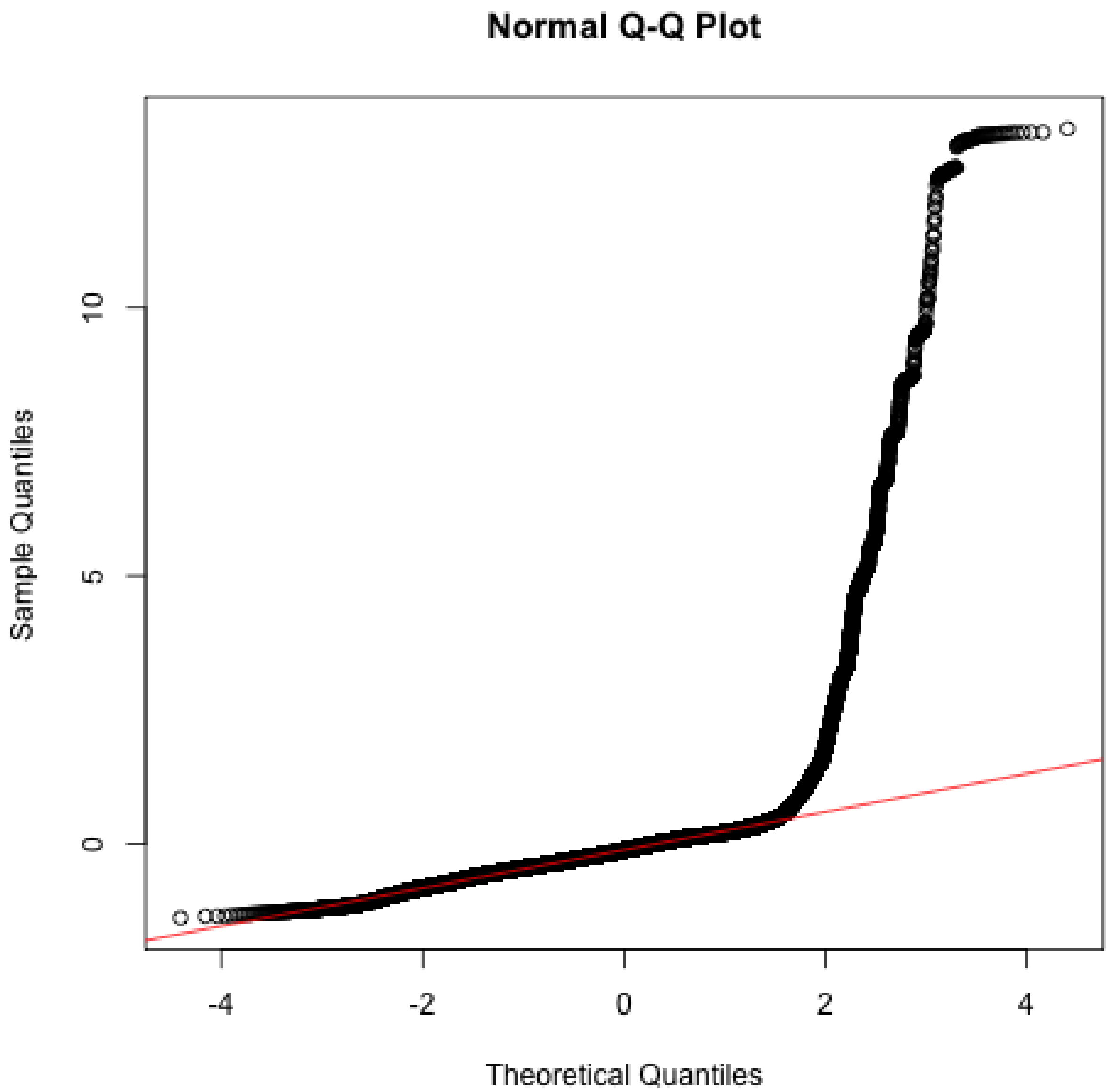


Figure 1: QQ plot for a model with a Gaussian distribution

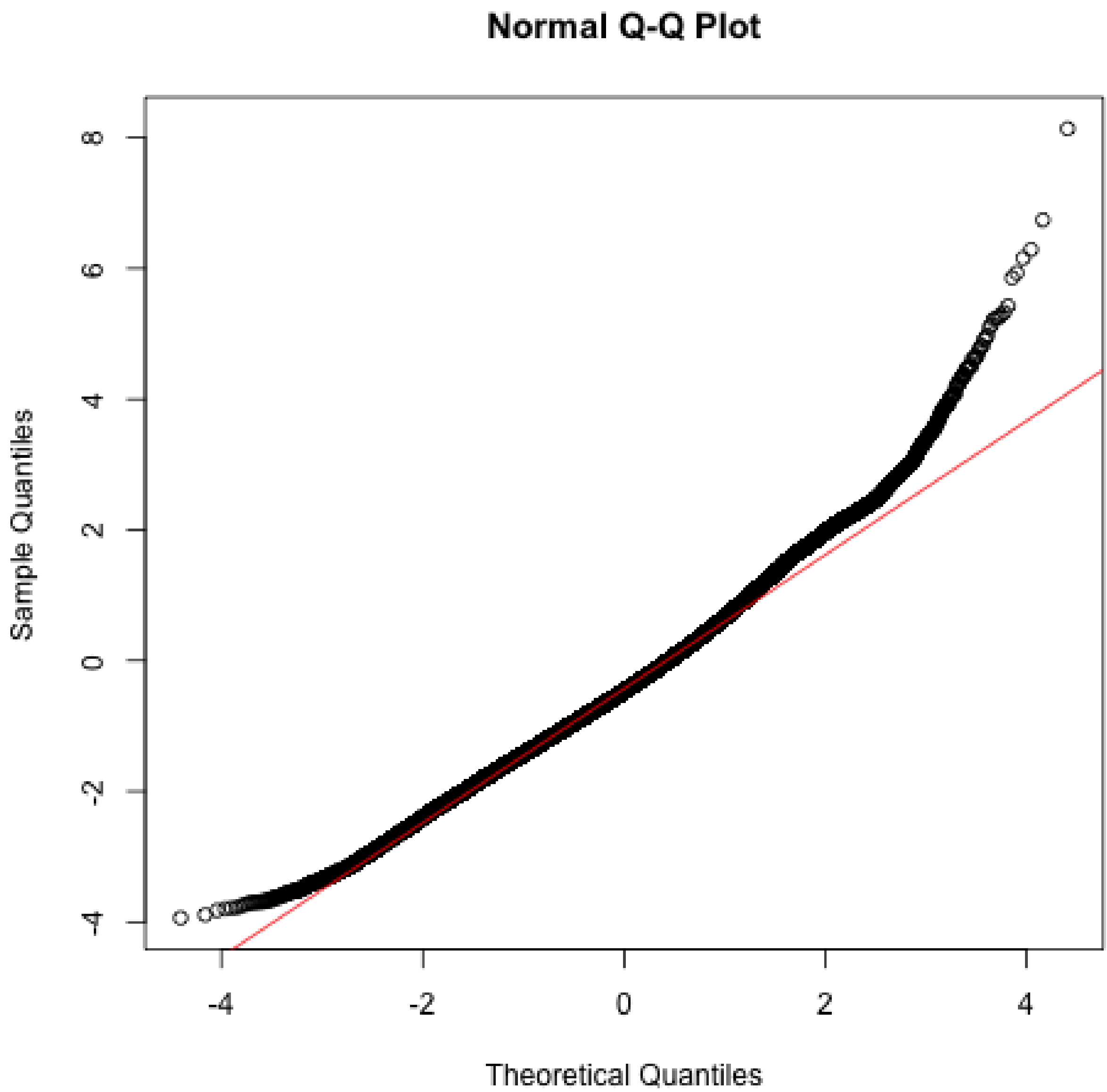


Figure 2: QQ plot for the model with a Gamma distribution

Still not perfect at higher levels, but much better than the Gaussian models.

Mixed effects modelling

Fit a model with a Gamma distribution (without language family controls, with a random intercept by family and with a random intercept and slope):

```
mA0Gamma = glmer(logAAM+0.0001 ~ 1 +
             pd + indiv + mas + ua + lto + indul +
             ggr + invpro +
             SIZE + BTM + LEV + ROA + MEET + LOSS +
             (1 | fyear) +
             (1 | indus),
           data = d2,
           family=Gamma(link="log"))

mA1Gamma = update(mA0Gamma,~.+strongftr)
```

Now we add a random intercept for each language family:

```
mB0Gamma = glmer(logAAM+0.0001 ~ 1 +
             pd + indiv + mas + ua + lto + indul +
             ggr + invpro +
             SIZE + BTM + LEV + ROA + MEET + LOSS +
             (1 | fyear) +
             (1 | indus) +
             (1 | mainLanguageFamily),
           data = d2,
           family=Gamma(link="log"))
mB1Gamma = update(mB0Gamma, ~.+strongftr)
```

We also add a random slope for FTR by language family:

```
mB2Gamma= glmer(logAAM+0.0001 ~ 1 +
             pd + indiv + mas + ua + lto + indul +
             ggr + invpro +
             SIZE + BTM + LEV + ROA + MEET + LOSS +
             strongftr +
             (1 | fyear) +
             (1 | indus) +
             (1 + I(as.numeric(strongftr)) | mainLanguageFamily),
           data = d2,
           family=Gamma(link="log"))
```

Use model comparison to see if the random effects are explaining variance in the model:

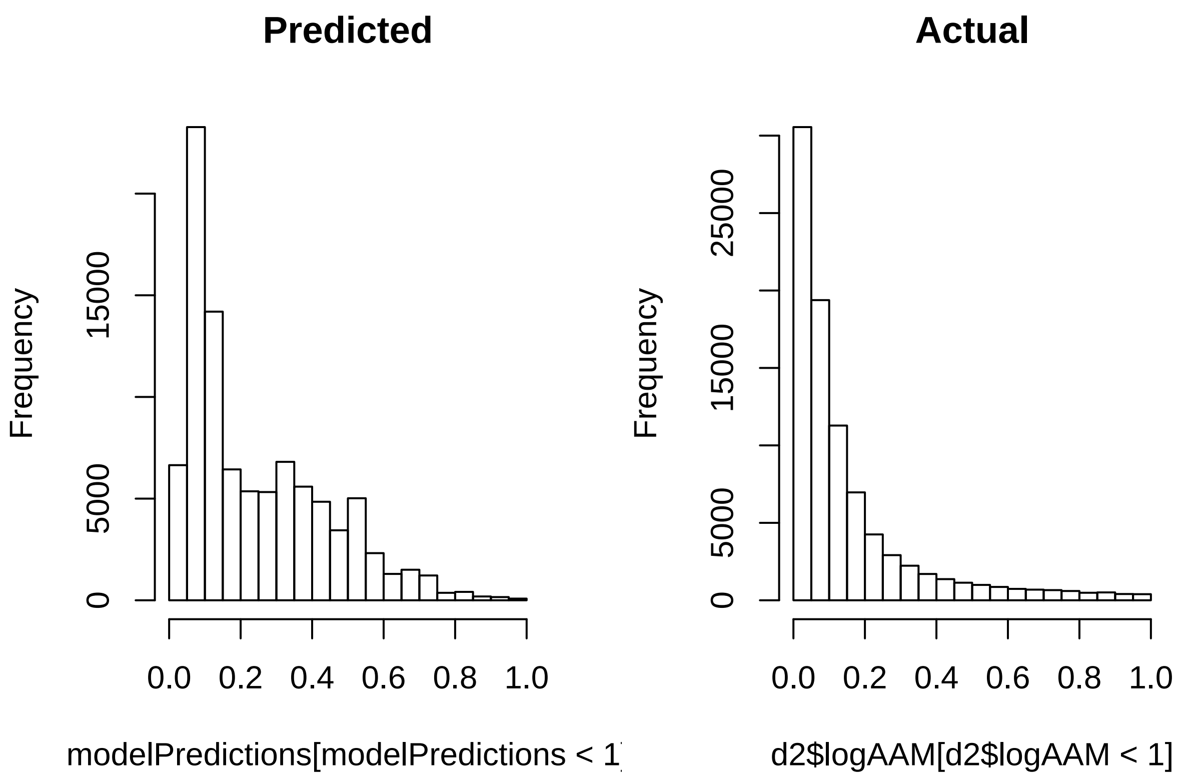
```
anova(mA1Gamma,mB1Gamma,mB2Gamma)
```

```
## Data: d2
## Models:
## mA1Gamma: logAAM + 1e-04 ~ pd + indiv + mas + ua + lto + indul + ggr +
## mA1Gamma:     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
## mA1Gamma:     (1 | indus) + strongftr
## mB1Gamma: logAAM + 1e-04 ~ pd + indiv + mas + ua + lto + indul + ggr +
## mB1Gamma:     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
## mB1Gamma:     (1 | indus) + (1 | mainLanguageFamily) + strongftr
## mB2Gamma: logAAM + 1e-04 ~ 1 + pd + indiv + mas + ua + lto + indul + ggr +
## mB2Gamma:     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
## mB2Gamma:     (1 | fyear) + (1 | indus) + (1 + I(as.numeric(strongftr)) |
```

```
## mB2Gamma:     mainLanguageFamily)
##          Df     AIC     BIC logLik deviance   Chisq Chi Df Pr(>Chisq)
## mA1Gamma 19 -135645 -135465  67841  -135683
## mB1Gamma 20 -138253 -138064  69146  -138293 2609.94      1  < 2.2e-16 ***
## mB2Gamma 22 -138423 -138215  69233  -138467  173.94      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Check that the model is producing a sensible distribution:

```
modelPredictions = exp(predict(mB1Gamma))-0.0001
par(mfrow=c(1,2))
hist(modelPredictions[modelPredictions<1],main="Predicted")
hist(d2$logAAM[d2$logAAM<1],main="Actual")
```




```
par(mfrow=c(1,1))
png("../results/misc/qqplot_Gamma.png")
qqnorm(resid(mB1Gamma))
qqline(resid(mB1Gamma),col=2)
dev.off()
```

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

Model results:

```
summary(mA1Gamma)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ pd + indiv + mas + ua + lto + indul + ggr +
```

```
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
##     (1 | indus) + strongftr
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
## -135644.9 -135465.2   67841.5 -135682.9     94688
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.821 -0.619 -0.338  0.225 34.398
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fyear    (Intercept) 0.2003   0.4476
##  indus    (Intercept) 0.1181   0.3436
##  Residual             1.4813   1.2171
## Number of obs: 94707, groups:  fyear, 20; indus, 9
##
## Fixed effects:
##              Estimate Std. Error t value Pr(>|z|)
## (Intercept) -1.628030   0.126587 -12.861  < 2e-16 ***
## pd          -0.086784   0.007523 -11.535  < 2e-16 ***
## indiv       -0.051336   0.012296  -4.175 2.98e-05 ***
## mas          0.107415   0.004742  22.651  < 2e-16 ***
## ua          -0.191481   0.005521 -34.685  < 2e-16 ***
## lto         -0.364160   0.009434 -38.600  < 2e-16 ***
## indul        0.071917   0.007524   9.559  < 2e-16 ***
## ggr         -0.083885   0.006459 -12.987  < 2e-16 ***
## invpro      -0.212485   0.004301 -49.407  < 2e-16 ***
## SIZE         0.008388   0.004261   1.968    0.049 *
## BTM         -0.044603   0.003439 -12.970  < 2e-16 ***
## LEV         -0.002114   0.003680  -0.575    0.566
## ROA          0.000377   0.003958   0.095    0.924
## MEET1        0.052116   0.007133   7.306 2.75e-13 ***
## LOSS1        0.308902   0.013138  23.512  < 2e-16 ***
## strongftr1   0.538942   0.011844  45.505  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```r
summary(mB1Gamma)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ pd + indiv + mas + ua + lto + indul + ggr +
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
##     (1 | indus) + (1 | mainLanguageFamily) + strongftr
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
```

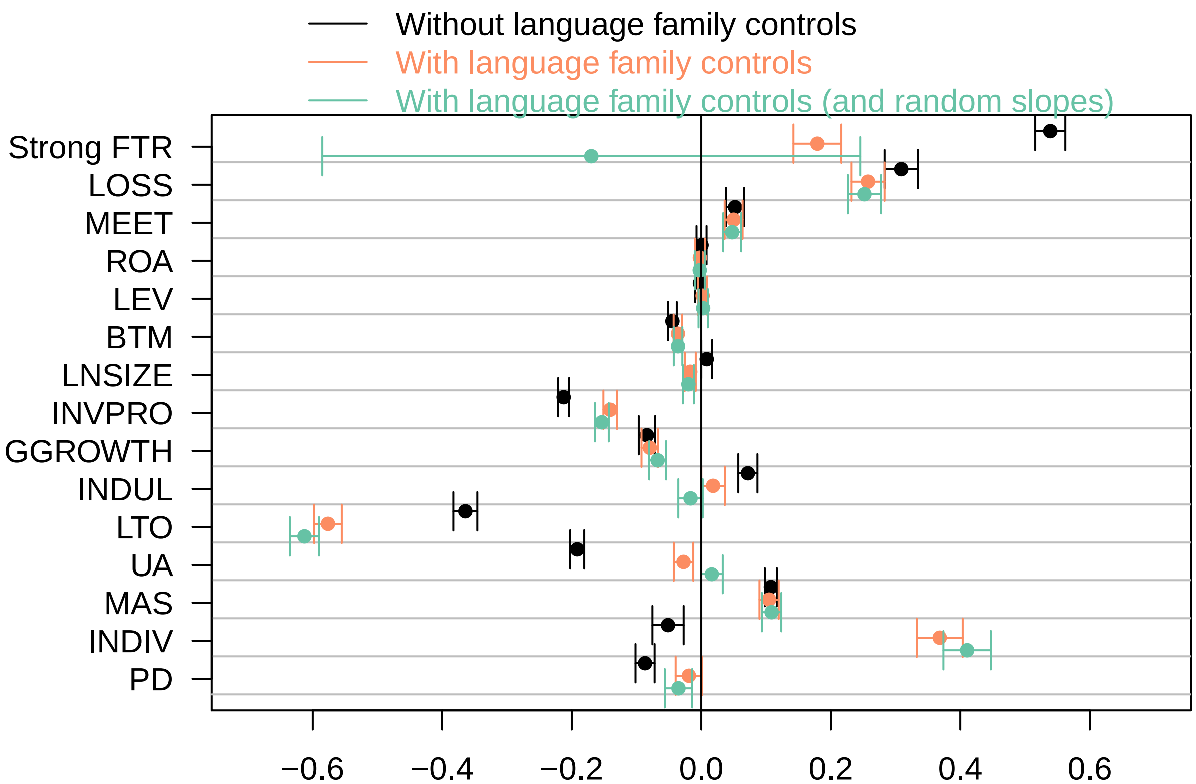
```
## -138252.8 -138063.7   69146.4 -138292.8     94687
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8377 -0.6265 -0.3331  0.2380 30.7849
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  fyear              (Intercept) 0.1944   0.4409
##  indus              (Intercept) 0.1172   0.3424
##  mainLanguageFamily (Intercept) 1.0054   1.0027
##  Residual                       1.4246   1.1936
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
##
## Fixed effects:
##              Estimate Std. Error t value Pr(>|z|)
## (Intercept) -1.810417   0.311399  -5.814 6.11e-09 ***
## pd          -0.019086   0.010443  -1.828 0.067599 .
## indiv        0.368239   0.018089  20.357  < 2e-16 ***
## mas          0.104654   0.007575  13.816  < 2e-16 ***
## ua          -0.027408   0.007689  -3.564 0.000365 ***
## lto         -0.576462   0.010866 -53.050  < 2e-16 ***
## indul        0.018231   0.009269   1.967 0.049193 *
## ggr         -0.079519   0.006515 -12.205  < 2e-16 ***
## invpro      -0.140606   0.005364 -26.213  < 2e-16 ***
## SIZE        -0.017009   0.004316  -3.941 8.11e-05 ***
## BTM         -0.036026   0.003404 -10.583  < 2e-16 ***
## LEV          0.002055   0.003653   0.563 0.573636
## ROA         -0.002173   0.003933  -0.553 0.580529
## MEET1        0.049932   0.007059   7.074 1.51e-12 ***
## LOSS1        0.257498   0.013082  19.683  < 2e-16 ***
## strongftr1   0.179187   0.018866   9.498  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
summary(mB2Gamma)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ 1 + pd + indiv + mas + ua + lto + indul + ggr +
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus) + (1 + I(as.numeric(strongftr)) |
##     mainLanguageFamily)
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
## -138422.8 -138214.7   69233.4 -138466.8     94685
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -0.8383 -0.6263 -0.3326  0.2380 31.1752 
## 
## Random effects:
##  Groups             Name                      Variance Std.Dev. Corr 
##  fyear              (Intercept)               0.2067   0.4547        
##  indus              (Intercept)               0.1191   0.3452        
##  mainLanguageFamily (Intercept)               0.3341   0.5780        
##                     I(as.numeric(strongftr)) 0.3950   0.6285   -0.14
##  Residual                                     1.4226   1.1927        
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
## 
## Fixed effects:
##              Estimate Std. Error t value Pr(>|z|)    
## (Intercept) -1.620150   0.268547  -6.033 1.61e-09 ***
## pd          -0.035222   0.010760  -3.273  0.00106 ** 
## indiv        0.410556   0.018691  21.966  < 2e-16 ***
## mas          0.108590   0.007622  14.248  < 2e-16 ***
## ua           0.016182   0.008635   1.874  0.06095 .  
## lto         -0.612745   0.011464 -53.448  < 2e-16 ***
## indul       -0.016606   0.009617  -1.727  0.08420 .  
## ggr         -0.067387   0.006620 -10.180  < 2e-16 ***
## invpro      -0.153494   0.005415 -28.347  < 2e-16 ***
## SIZE        -0.019870   0.004320  -4.599 4.24e-06 ***
## BTM         -0.035865   0.003402 -10.543  < 2e-16 ***
## LEV          0.002873   0.003653   0.787  0.43156    
## ROA         -0.002389   0.003930  -0.608  0.54317    
## MEET1        0.047748   0.007056   6.767 1.32e-11 ***
## LOSS1        0.252019   0.013080  19.268  < 2e-16 ***
## strongftr1  -0.169746   0.211944  -0.801  0.42319    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

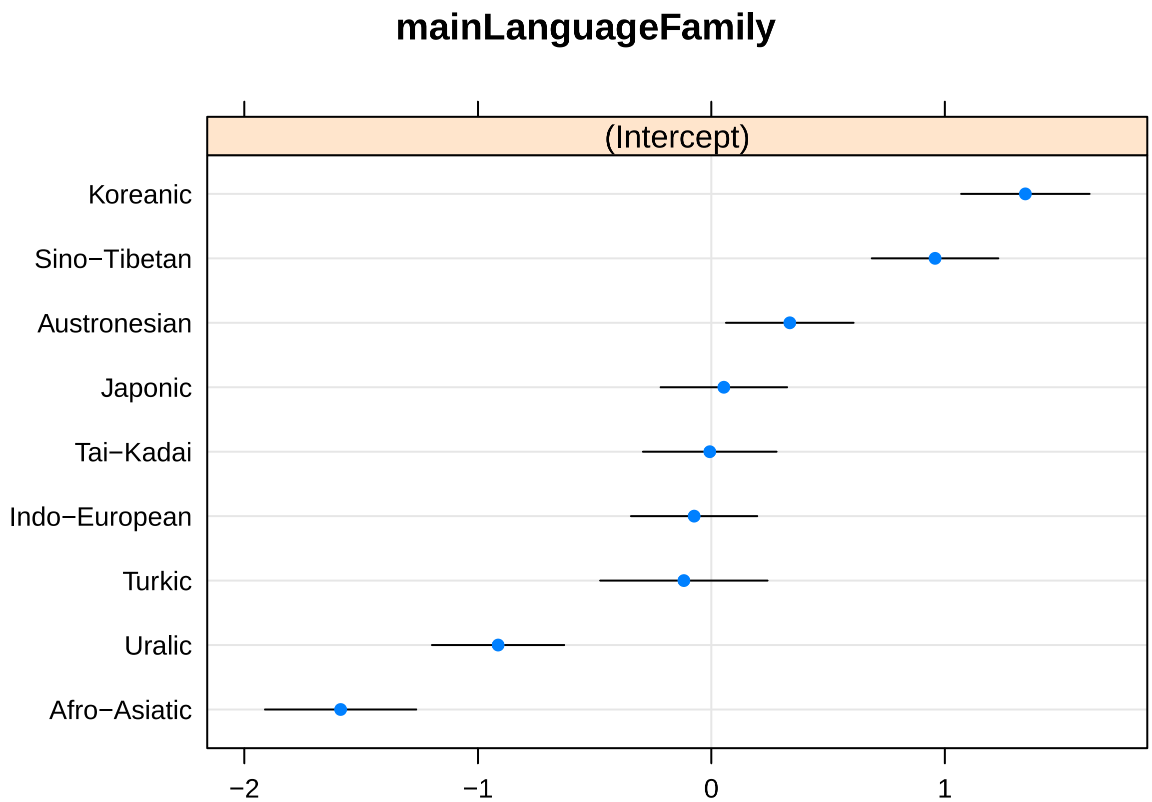
Plot fixed effects for all models (code hidden):



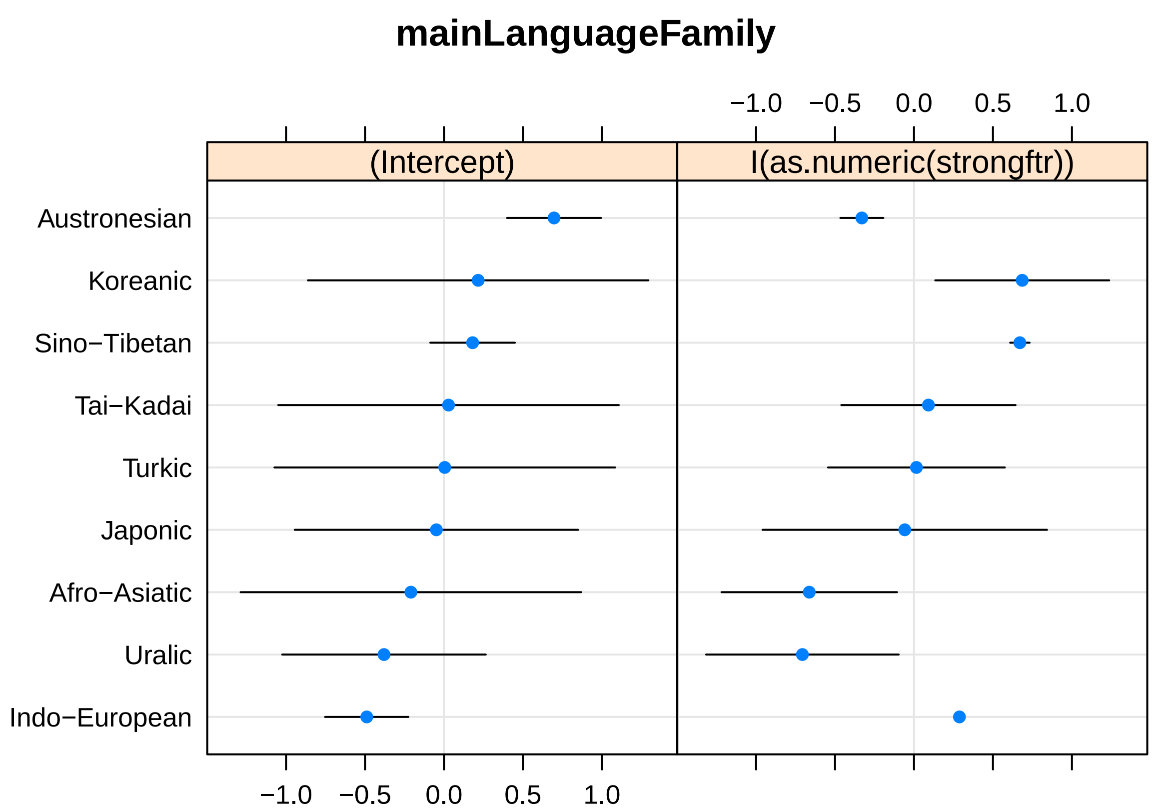

```
## pdf
##   2
```

View random effects for language family

```
dotplot(ranef(mB1Gamma))$mainLanguageFamily
```




```
dotplot(ranef(mB2Gamma))$mainLanguageFamily
```



Summary

Without a random intercept by main language family: There was a significant main effect of FTR (beta = 0.54 , log likelihood difference = 1000 , df = 1 , Chi Squared = 2036.03 , p = 0).

With a random intercept by main language family: There was a significant main effect of FTR (beta = 0.18 , log likelihood difference = 45 , df = 1 , Chi Squared = 89.4 , p = 3.2e-21).

With a random intercept by main language family and a random slope for FTR by main language family: There was a significant main effect of FTR (beta = -0.17 , log likelihood difference = 87 , df = 2 , Chi Squared = 173.94 , p = 1.7e-38).

Other effects

Below are some statistics for other effects, using the same method as above:

```r
resOther = data.frame(
  Label = NA,
  Beta = NA,
  loglikDiff = NA,
  df = NA,
  chisq.test = NA,
  p = NA, stringsAsFactors = F)
for(v in c("pd",'indiv','mas',
           'ua','lto','indul','ggr',
           'SIZE',"BTM","LEV","ROA")){
  mAOther0 = update(mA1Gamma, paste("~ . -",v))
  mAOtherAnova = anova(mAOther0,mA1Gamma)
  mBOther0 = update(mB2Gamma, paste("~ . -",v))
  mBOtherAnova = anova(mBOther0,mB2Gamma)
  mARes = getMEText(mAOtherAnova,"X",summary(mA1Gamma)$coef[v,],returnText = F)
  mBRes = getMEText(mBOtherAnova,"X",summary(mB2Gamma)$coef[v,],returnText = F)
  resOther = rbind(resOther, c(paste(v,": No controls"),mARes))
  resOther = rbind(resOther, c(paste(v,": With Controls"),mBRes))
}
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00113304
## (tol = 0.001, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00145232
## (tol = 0.001, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00135493
## (tol = 0.001, component 1)
```

```r
resOther = resOther[!is.na(resOther$Label),]
print(resOther)
```

```
##                    Label    Beta loglikDiff df chisq.test        p
## 2       pd : No controls  -0.087         65  1     130.74  2.8e-30
## 3     pd : With Controls  -0.035        5.4  1      10.75    0.001
## 4    indiv : No controls  -0.051        8.7  1      17.43    3e-05
## 5  indiv : With Controls    0.41        230  1     451.83 2.9e-100
## 6      mas : No controls    0.11        240  1      489.6 1.7e-108
## 7    mas : With Controls    0.11        100  1     200.99  1.3e-45
## 8       ua : No controls   -0.19        610  1     1212.6 1.1e-265
## 9     ua : With Controls   0.016        1.8  1       3.51    0.061
## 10     lto : No controls   -0.36        760  1     1522.8        0
## 11   lto : With Controls   -0.61       1500  1    2900.95        0
## 12   indul : No controls   0.072         45  1      90.36    2e-21
## 13 indul : With Controls  -0.017        1.5  1       2.98    0.084
## 14     ggr : No controls  -0.084         86  1     171.49  3.5e-39
## 15   ggr : With Controls  -0.067         53  1     105.06  1.2e-24
## 16    SIZE : No controls  0.0084        1.9  1       3.88    0.049
## 17  SIZE : With Controls   -0.02         11  1      21.11  4.3e-06
## 18     BTM : No controls  -0.045         80  1     160.06  1.1e-36
```

```
## 19   BTM : With Controls  -0.036        53  1     106.91  4.6e-25
## 20     LEV : No controls -0.0021       0.16  1       0.33     0.57
## 21   LEV : With Controls  0.0029       0.31  1       0.62     0.43
## 22     ROA : No controls 0.00038     0.0045  1       0.01     0.92
## 23   ROA : With Controls -0.0024       0.19  1       0.37     0.54
```

```
resOther2 = cbind(
  resOther[seq(1,nrow(resOther)-1,by=2),c("Label","Beta","p")],
  resOther[seq(2,nrow(resOther),by=2),c("Beta","p")])
write.csv(resOther2,"../results/BetaResults_OtherVariables.csv",row.names = F)
```

Alternative tests

Gaussian distribution model

Model A: no controls for language family

Model `mA0` is a baseline model and model `mA1` adds the effect for FTR.

```
mA0 = lmer(AAM.scaled ~ 1 +
             invpro +
             pd + indiv + mas + ua + lto + indul +
             ggr +
             SIZE + BTM + LEV + ROA +
             MEET + LOSS +
             (1 | fyear) +
             (1 | indus),
           data = d2)
mA1 = update(mA0,~.+strongftr)
```

Look at the estiamtes for variables within model `mA1`:

```
summary(mA1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
##     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
##     indus) + strongftr
##    Data: d2
##
## REML criterion at convergence: 256954.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4990 -0.3712 -0.1335  0.1310 14.2360
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fyear    (Intercept) 0.04171  0.2042
##  indus    (Intercept) 0.01485  0.1218
##  Residual             0.88034  0.9383
## Number of obs: 94707, groups:  fyear, 20; indus, 9
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.153654   0.063450   2.422
## invpro      -0.100554   0.003510 -28.644
## pd           0.012819   0.006140   2.088
## indiv        0.013771   0.009557   1.441
## mas          0.063738   0.003837  16.610
## ua          -0.062749   0.004456 -14.081
## lto         -0.124885   0.007243 -17.241
## indul        0.031991   0.006528   4.901
## ggr         -0.091870   0.005601 -16.401
## SIZE         0.036230   0.003744   9.677
## BTM         -0.010561   0.003274  -3.225
## LEV          0.006785   0.003353   2.023
## ROA          0.014690   0.003787   3.879
## MEET1        0.031684   0.006293   5.035
```

```
## LOSS1        0.167342   0.011696  14.308
## strongftr1   0.149591   0.010309  14.511
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Compare the fit of the two models to assess the effect of FTR:

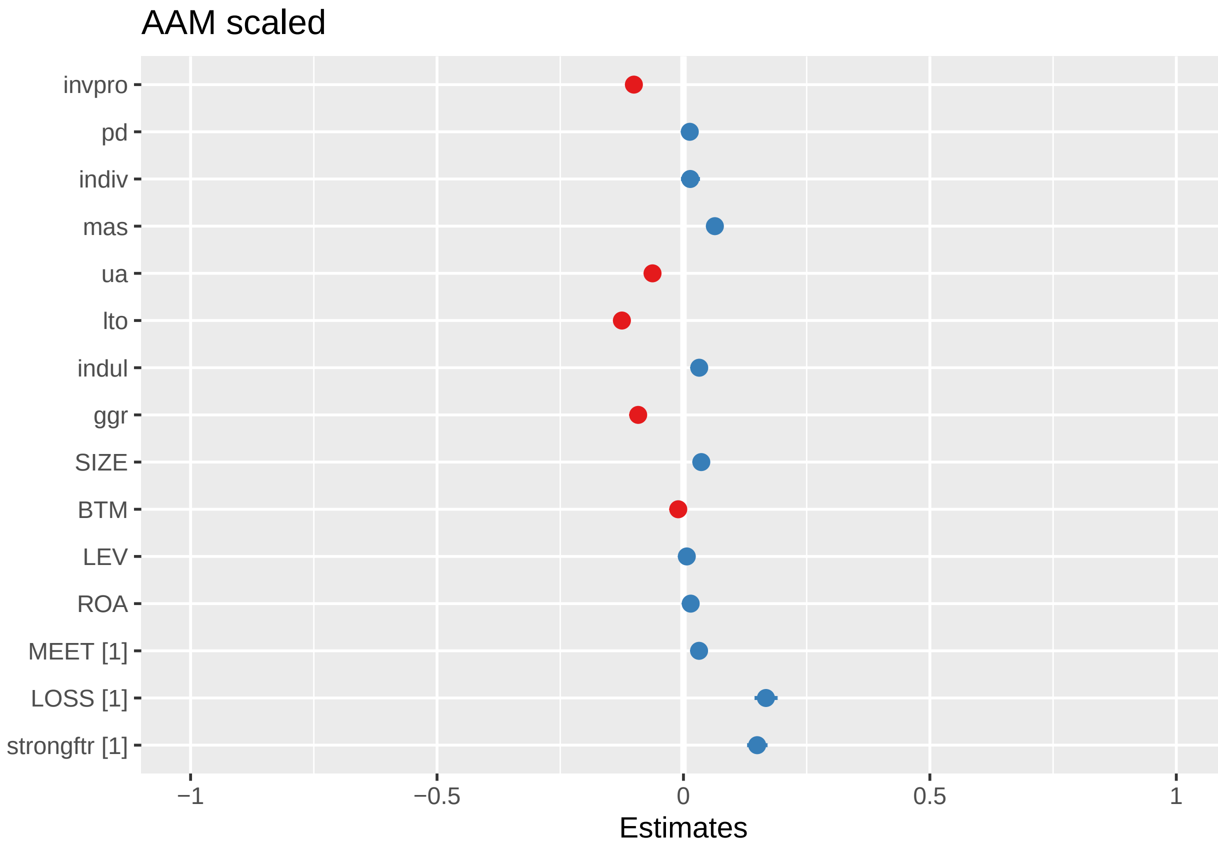
```r
anova(mA0,mA1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: d2
## Models:
## mA0: AAM.scaled ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
## mA0:     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
## mA0:     (1 | indus)
## mA1: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
## mA1:     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
## mA1:     indus) + strongftr
##     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mA0 18 257063 257234 -128514   257027
## mA1 19 256855 257035 -128409   256817 210.38      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot fixed effects:

```
plot_model(mA1,type="est",p.kr = F)
```



Model B: with controls for language family

Model `mB0` is the same as `mA0`, but with controls for language family. Model `mB1` adds the FTR variable to the model for comparison.

```
mB0= update(mA0,  ~.+(1 | mainLanguageFamily))
mB1= update(mB0,  ~.+strongftr)
```

Look at the estimates for `mB1`:

```
summary(mB1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
##     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
##     indus) + (1 | mainLanguageFamily) + strongftr
##    Data: d2
##
## REML criterion at convergence: 256343.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4757 -0.3765 -0.1269  0.1357 14.2536
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  fyear              (Intercept) 0.04165  0.2041
##  indus              (Intercept) 0.01531  0.1237
##  mainLanguageFamily (Intercept) 0.10018  0.3165
##  Residual                       0.87436  0.9351
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.246953   0.124566   1.983
## invpro      -0.099264   0.004529 -21.919
## pd           0.055776   0.009174   6.079
## indiv        0.162885   0.014282  11.405
## mas          0.039045   0.006110   6.390
## ua          -0.047815   0.006584  -7.262
## lto         -0.220395   0.009085 -24.259
## indul        0.016509   0.007498   2.202
## ggr         -0.102036   0.005758 -17.720
## SIZE         0.020039   0.003848   5.207
## BTM         -0.010632   0.003271  -3.251
## LEV          0.009026   0.003355   2.690
## ROA          0.015726   0.003775   4.165
## MEET1        0.030613   0.006273   4.880
## LOSS1        0.140498   0.011720  11.988
## strongftr1   0.021657   0.016910   1.281

##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Compare the two models to assess the significance of the FTR variable:

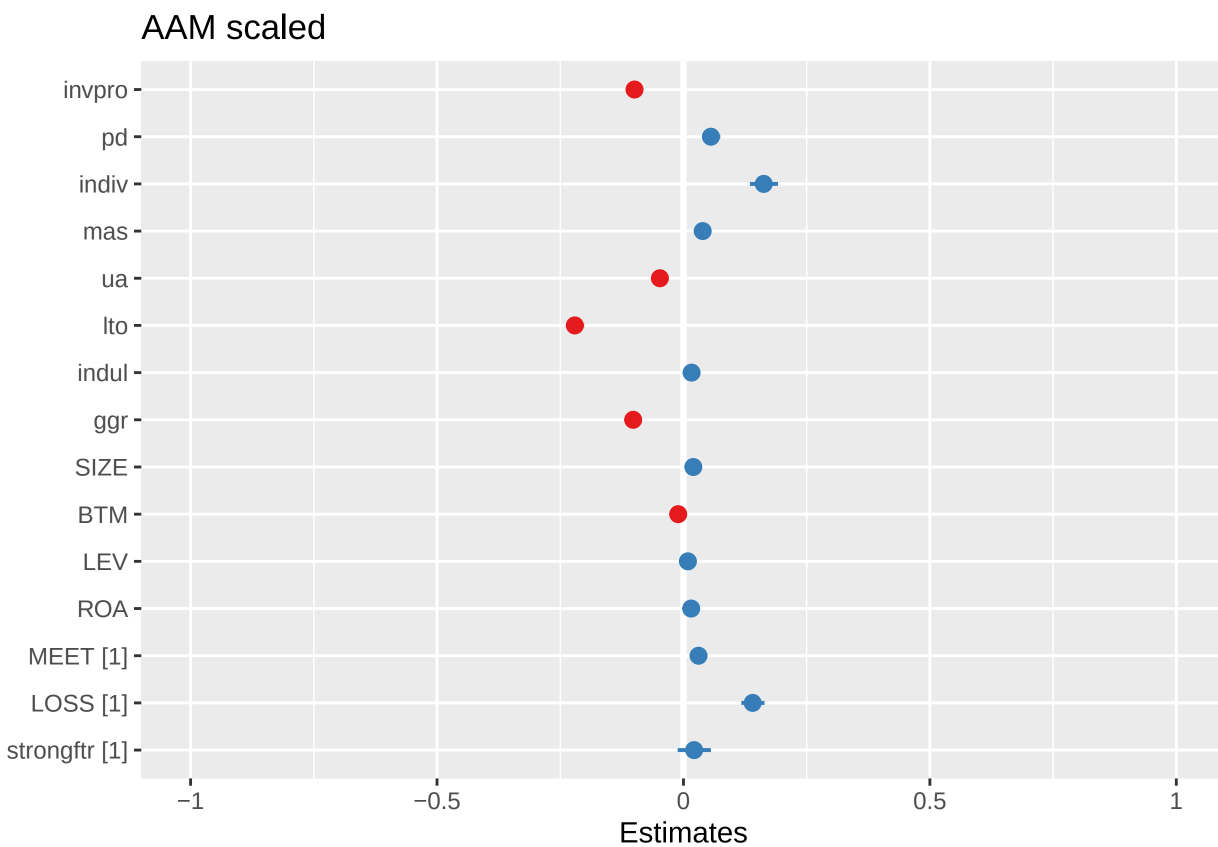
```r
anova(mB0,mB1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: d2
## Models:
## mB0: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
## mB0:     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
## mB0:     indus) + (1 | mainLanguageFamily)
## mB1: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
## mB1:     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
## mB1:     indus) + (1 | mainLanguageFamily) + strongftr
##     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mB0 19 256254 256434 -128108   256216
## mB1 20 256254 256443 -128107   256214 1.6614      1     0.1974
```

Plot fixed effects with controls for language family:

```r
plot_model(mB1,type="est",p.kr = F)
```



Random slopes for FTR

Test if adding a random slope for FTR by language family significantly improves the fit of the model:

```r
mB2 = lmer(AAM.scaled ~ 1 +
             invpro +
             pd + indiv + mas + ua + lto + indul +
             ggr +
             SIZE + BTM + LEV + ROA +
```

```r
             MEET + LOSS +
             strongftr +
             (1 | fyear) +
             (1 | indus) +
             (1 + strongftr | mainLanguageFamily),
           data = d2)
```

```r
anova(mB1,mB2)
```

```
## refitting model(s) with ML (instead of REML)

## Data: d2
## Models:
## mB1: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
## mB1:     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
## mB1:     indus) + (1 | mainLanguageFamily) + strongftr
## mB2: AAM.scaled ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
## mB2:     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
## mB2:     (1 | fyear) + (1 | indus) + (1 + strongftr | mainLanguageFamily)
##     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mB1 20 256254 256443 -128107   256214
## mB2 22 256180 256388 -128068   256136 78.319      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, model fit significantly improves. The effect of FTR is even weaker:

```r
summary(mB2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: AAM.scaled ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
##     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus) + (1 + strongftr | mainLanguageFamily)
##    Data: d2
##
## REML criterion at convergence: 256260.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4711 -0.3755 -0.1230  0.1344 14.2610
##
## Random effects:
##  Groups             Name        Variance Std.Dev. Corr
##  fyear              (Intercept) 0.04225  0.2055
##  indus              (Intercept) 0.01569  0.1253
##  mainLanguageFamily (Intercept) 0.04333  0.2082
##                     strongftr1  0.07239  0.2691   0.60
##  Residual                       0.87352  0.9346
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)  0.2949707  0.1012130   2.914
## invpro      -0.1061720  0.0045831 -23.166
## pd           0.0499991  0.0093405   5.353
## indiv        0.1890539  0.0147841  12.788
```

```
## mas           0.0464436  0.0062111   7.478
## ua           -0.0248139  0.0071357  -3.477
## lto          -0.2406902  0.0094705 -25.415
## indul         0.0007689  0.0076796   0.100
## ggr          -0.0950665  0.0058169 -16.343
## SIZE          0.0181445  0.0038528   4.709
## BTM          -0.0105549  0.0032700  -3.228
## LEV           0.0094880  0.0033542   2.829
## ROA           0.0161278  0.0037749   4.272
## MEET1         0.0295640  0.0062706   4.715
## LOSS1         0.1391878  0.0117246  11.871
## strongftr1   -0.0366099  0.1089020  -0.336
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Calculate p-value for effect of FTR:

```
mB2_noFTR = update(mB2, ~. - strongftr)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00586163 (tol = 0.002, component 1)
```

```
anova(mB2,mB2_noFTR)
```

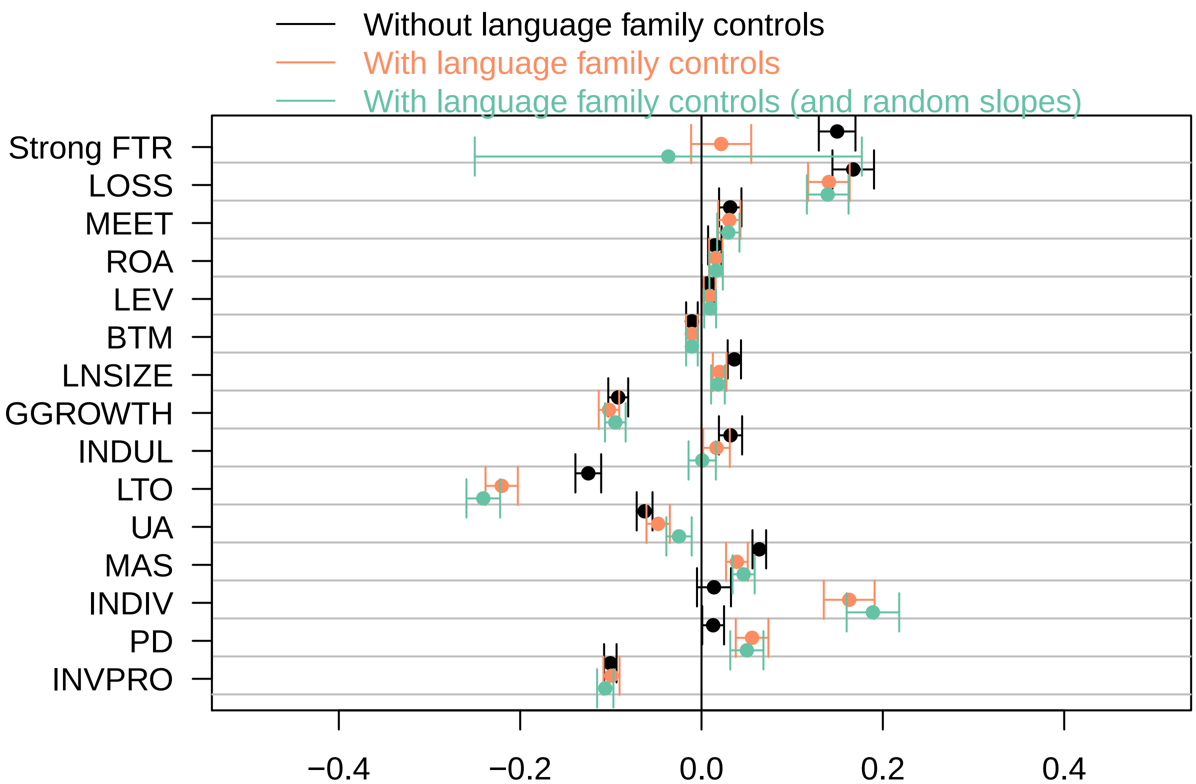
```
## refitting model(s) with ML (instead of REML)
```

```
## Data: d2
## Models:
## mB2_noFTR: AAM.scaled ~ invpro + pd + indiv + mas + ua + lto + indul + ggr +
## mB2_noFTR:     SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) + (1 |
## mB2_noFTR:     indus) + (1 + strongftr | mainLanguageFamily)
## mB2: AAM.scaled ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
## mB2:     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
## mB2:     (1 | fyear) + (1 | indus) + (1 + strongftr | mainLanguageFamily)
##           Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mB2_noFTR 21 256178 256377 -128068   256136
## mB2       22 256180 256388 -128068   256136 0.1409      1     0.7073
```

Plot fixed effects:

```
plotA = get_model_data(mA1,type="est", transform = NULL)
plotB = get_model_data(mB1,type="est", transform = NULL)
plotB2 = get_model_data(mB2,type="est", transform = NULL)

plotBoth(plotA,plotB,plotB2)
```




```
pdf("../results/singleMembershipModel.pdf",
    width=6,height=5)
  plotBoth(plotA,plotB,plotB2)
dev.off()
```

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

Summary of Gaussian model

Without a random intercept by main language family: There was a significant main effect of FTR (beta = 0.15 , log likelihood difference = 110 , df = 1 , Chi Squared = 210.38 , p = 1.1e-47).

With a random intercept by main language family: There was no significant main effect of FTR (beta = 0.022 , log likelihood difference = 0.83 , df = 1 , Chi Squared = 1.66 , p = 0.2).

Model with average FTR per family

A reviewer asked us to include the mean FTR for the language family as a fixed effect.

```
meanFTRPerLangFam = tapply(as.numeric(d2$strongftr)-1,
                    d2$mainLanguageFamily,mean)
d2$meanFTR = meanFTRPerLangFam[d2$mainLanguageFamily]
mA1MeanFTR = glmer(logAAM+0.0001 ~ 1 +
             pd + indiv + mas + ua + lto + indul +
             ggr + invpro +
             SIZE + BTM + LEV + ROA + MEET + LOSS +
             meanFTR +
             strongftr +
             (1 | fyear) +
             (1 | indus),
           data = d2,
           family=Gamma(link="log"))
```

The results are qualitatively the same: FTR is a significant predictor. Note that family mean FTR is not a significant predictor when strong FTR is included in the model:

```
summary(mA1MeanFTR)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ 1 + pd + indiv + mas + ua + lto + indul + ggr +
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + meanFTR +
##     strongftr + (1 | fyear) + (1 | indus)
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
## -135644.4 -135455.2   67842.2 -135684.4     94687
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.821 -0.619 -0.338  0.224 34.573
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fyear    (Intercept) 0.2004   0.4476
##  indus    (Intercept) 0.1184   0.3440
##  Residual             1.4818   1.2173
## Number of obs: 94707, groups:  fyear, 20; indus, 9
##
## Fixed effects:
##               Estimate Std. Error t value Pr(>|z|)
## (Intercept) -1.6426041  0.1274645 -12.887  < 2e-16 ***
## pd          -0.0853228  0.0076216 -11.195  < 2e-16 ***
## indiv       -0.0606177  0.0145050  -4.179 2.93e-05 ***
## mas          0.1133563  0.0068184  16.625  < 2e-16 ***
## ua          -0.1937446  0.0058262 -33.254  < 2e-16 ***
## lto         -0.3650125  0.0094663 -38.559  < 2e-16 ***
## indul        0.0754436  0.0080694   9.349  < 2e-16 ***
## ggr         -0.0849890  0.0065279 -13.019  < 2e-16 ***
## invpro      -0.2124214  0.0043036 -49.359  < 2e-16 ***
```

```
## SIZE         0.0090862  0.0043004   2.113   0.0346 *
## BTM         -0.0446168  0.0034387 -12.975  < 2e-16 ***
## LEV         -0.0024430  0.0036894  -0.662   0.5079
## ROA          0.0004548  0.0039584   0.115   0.9085
## MEET1        0.0521603  0.0071331   7.312 2.62e-13 ***
## LOSS1        0.3097837  0.0131594  23.541  < 2e-16 ***
## meanFTR      0.0327991  0.0271097   1.210   0.2263
## strongftr1   0.5285639  0.0146362  36.113  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 17 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Check model with random intercepts and random slopes:

```
mB2MeanFTR= glmer(logAAM+0.0001 ~ 1 +
             pd + indiv + mas + ua + lto + indul +
             ggr + invpro +
             SIZE + BTM + LEV + ROA + MEET + LOSS +
             meanFTR +
             strongftr +
             (1 | fyear) +
             (1 | indus) +
             (1 + I(as.numeric(strongftr)) | mainLanguageFamily),
           data = d2,
           family=Gamma(link="log"))
```

The results are qualitatively the same. Note that the mean FTR is not a significant predictor when strong FTR is included in the model.

```
summary(mB2MeanFTR)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ 1 + pd + indiv + mas + ua + lto + indul + ggr +
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + meanFTR +
##     strongftr + (1 | fyear) + (1 | indus) + (1 + I(as.numeric(strongftr)) |
##     mainLanguageFamily)
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
## -138421.3 -138203.8   69233.7 -138467.3     94684
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8383 -0.6263 -0.3325  0.2381 31.1779
##
## Random effects:
##  Groups             Name                      Variance Std.Dev. Corr
##  fyear              (Intercept)               0.2064   0.4543
##  indus              (Intercept)               0.1187   0.3445
##  mainLanguageFamily (Intercept)               0.2277   0.4772
```

```
##                       I(as.numeric(strongftr)) 0.4240   0.6511   0.01
##  Residual                                      1.4227   1.1928
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
##
## Fixed effects:
##              Estimate Std. Error t value Pr(>|z|)
## (Intercept) -1.409887   0.393672  -3.581 0.000342 ***
## pd          -0.034912   0.010788  -3.236 0.001211 **
## indiv        0.410555   0.018691  21.966  < 2e-16 ***
## mas          0.108657   0.007626  14.248  < 2e-16 ***
## ua           0.016116   0.008638   1.866 0.062083 .
## lto         -0.612761   0.011461 -53.464  < 2e-16 ***
## indul       -0.016427   0.009631  -1.706 0.088084 .
## ggr         -0.067476   0.006623 -10.188  < 2e-16 ***
## invpro      -0.153434   0.005417 -28.324  < 2e-16 ***
## SIZE        -0.019892   0.004320  -4.604 4.14e-06 ***
## BTM         -0.035875   0.003402 -10.546  < 2e-16 ***
## LEV          0.002862   0.003653   0.784 0.433327
## ROA         -0.002384   0.003930  -0.607 0.544117
## MEET1        0.047753   0.007056   6.768 1.31e-11 ***
## LOSS1        0.252027   0.013080  19.268  < 2e-16 ***
## meanFTR     -0.483900   0.648341  -0.746 0.455446
## strongftr1  -0.128232   0.216230  -0.593 0.553158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 17 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Decision tree

A decision tree is a machine learning technique that tries to find patterns in data. It finds a series of yes/no questions which divide datapoints into partitions that look similar. 'Variable importance' is a measure of how influential each variable is in making decisions in the tree. This is a useful way of spotting patterns in the data that linear models might miss. In this case, if FTR is a good predictor, we would expect it to appear on the tree and have relatively high variable importance.

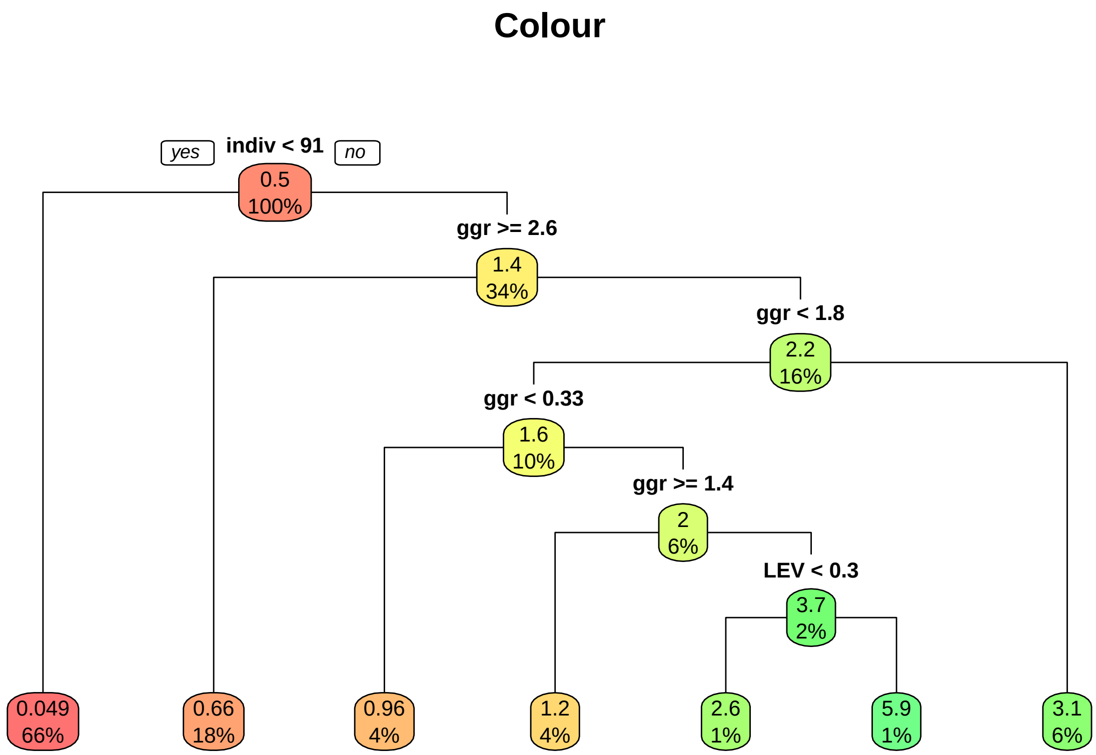
The package `REEMtree` allows the inclusion of random effects for year, industry type and main language family.

The tree below shows the yes/no questions at each branch in the tree. Coloured boxes show the mean AAM value and proportion of the data in that node. As it turns out, FTR does not appear on the tree. The most important factors are `ggr` and `indiv`.

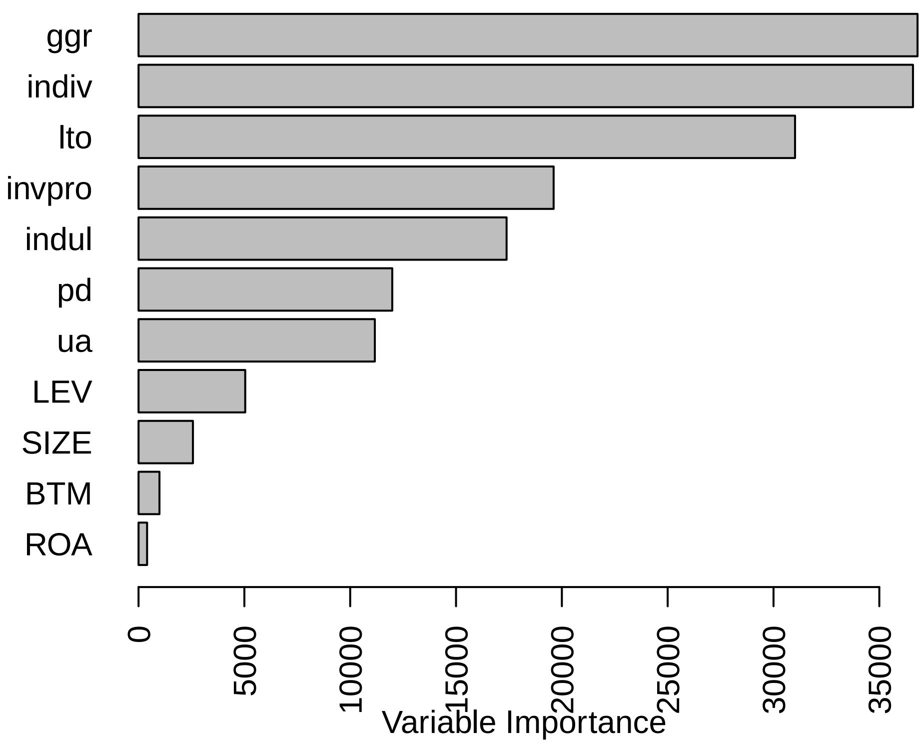
```
set.seed(1111) # set random seed for reproducability
rt = REEMtree(AAM ~
            strongftr +
            invpro +
            pd + indiv + mas + ua + lto + indul +
            ggr +
            SIZE + BTM + LEV + ROA +
            MEET + LOSS,
         data = d2Orig,
         random = ~1|mainLanguageFamily
                  ~1|fyear
                  ~1|indus)

rpart.plot(tree(rt), type=1,extra=100, branch.lty=1, box.palette="RdYlGn", main="Colour")
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
varimp = rt$Tree$variable.importance
par(mar=c(5,10,2,2))
barplot(sort(varimp), horiz=T, las=2,
        xlab="Variable Importance")
```

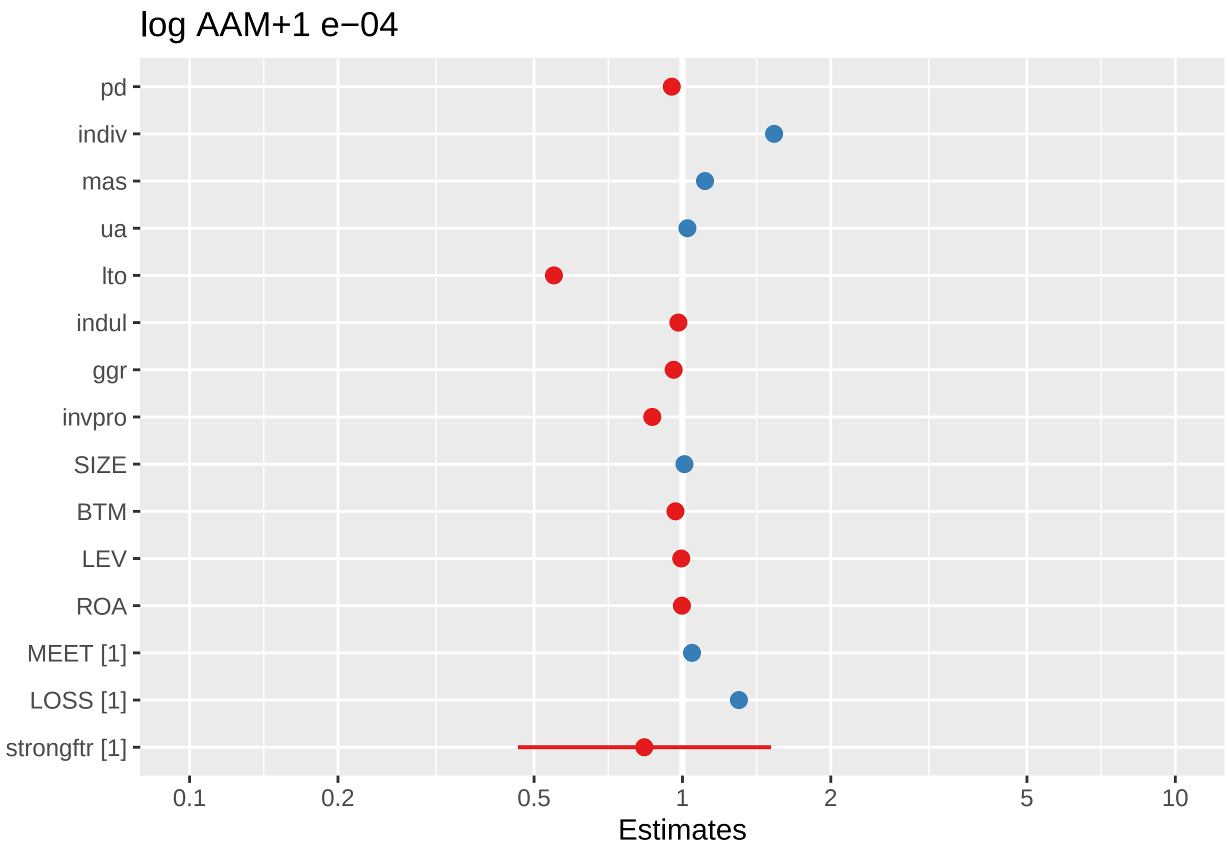



```
par(mar=c(5, 4, 4, 2) + 0.1)
```

Random slopes

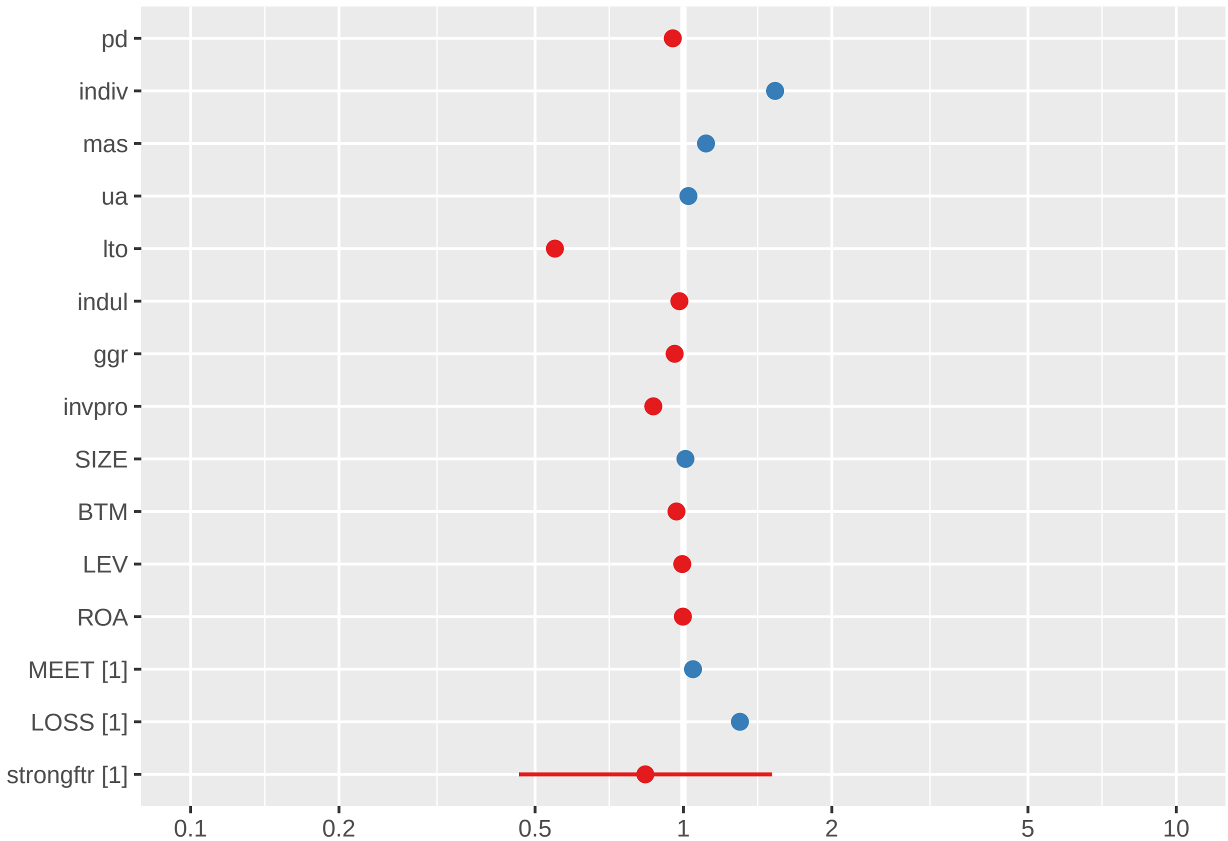
We can take a colser look at the random slopes for each language family:

```
d3 = d2[d2$mainLanguageFamily %in%
          c("Austronesian","Indo-European",
            "Sino-Tibetan","Uralic"),]
mB2GammaFamily= update(mB2Gamma, data = d3)
plot_model(mB2GammaFamily,type="est",p.kr = F)
```




```
plot_model(mB2GammaFamily,type="slope",vars="strongftr",show.legend = T)
```

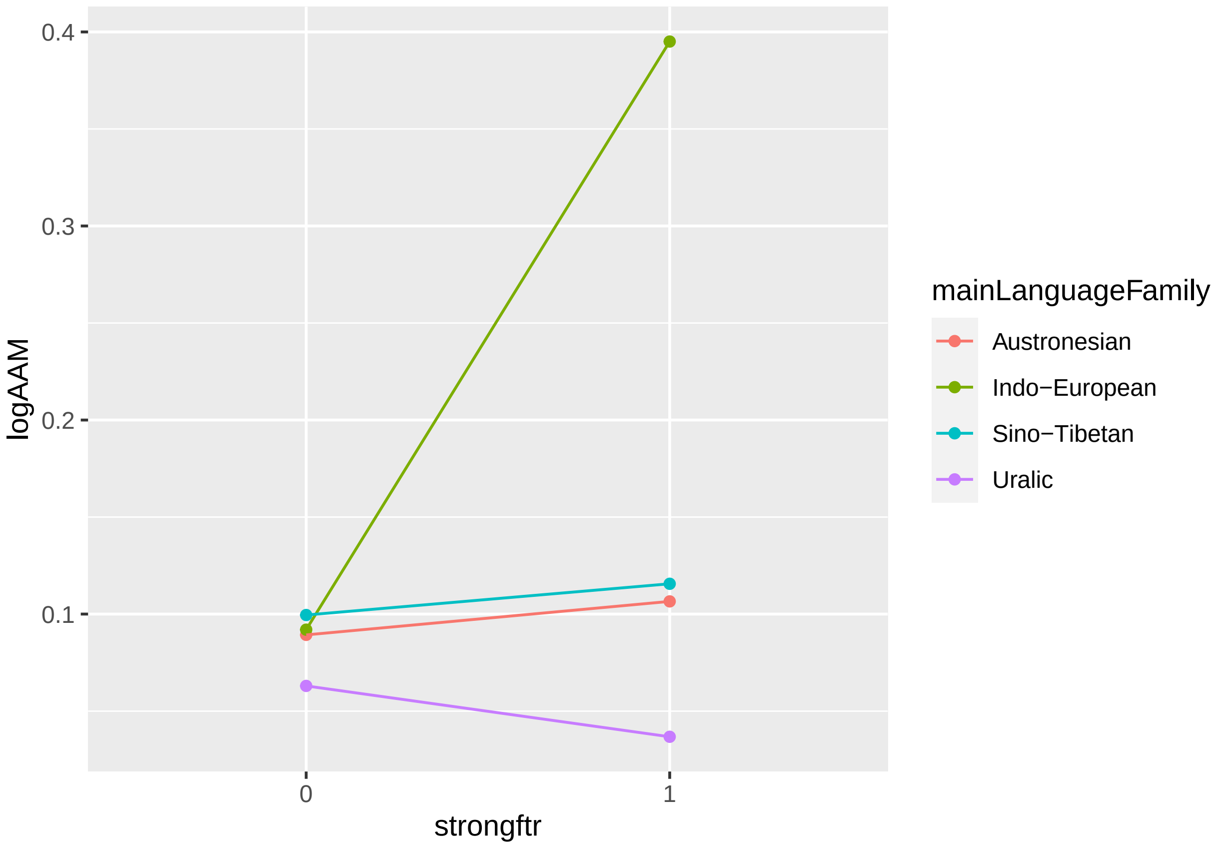
```
## Plot-type "slope" only available for linear models. Using `type = "est"` now.
```




```r
x = d2[d2$mainLanguageFamily %in%
          c("Austronesian","Indo-European",
            "Sino-Tibetan","Uralic"),] %>%
  group_by(mainLanguageFamily,strongftr) %>%
  summarise(logAAM=mean(logAAM))
```

```
## `summarise()` regrouping output by 'mainLanguageFamily' (override with `.groups` argument)
```

```r
ggplot(x,aes(x=strongftr,y=logAAM,color=mainLanguageFamily)) +
  geom_point() +
  geom_line(aes(group=mainLanguageFamily))
```



Model just for Indo-European languages:

```
mB1GammaIE = update(mA1Gamma,data=d2[d2$mainLanguageFamily=="Indo-European",])
summary(mB1GammaIE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ pd + indiv + mas + ua + lto + indul + ggr +
##     invpro + SIZE + BTM + LEV + ROA + MEET + LOSS + (1 | fyear) +
##     (1 | indus) + strongftr
##    Data: d2[d2$mainLanguageFamily == "Indo-European", ]
##
##      AIC      BIC   logLik deviance df.resid
## -39726.8 -39555.3  19882.4 -39764.8    61616
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.828 -0.632 -0.348  0.239 32.297
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fyear    (Intercept) 0.2689   0.5186
##  indus    (Intercept) 0.2729   0.5224
##  Residual             1.4590   1.2079
## Number of obs: 61635, groups:  fyear, 20; indus, 9
##
## Fixed effects:
```

```
##              Estimate Std. Error t value Pr(>|z|)
## (Intercept) -0.326899   0.179284  -1.823 0.068249 .
## pd           0.369494   0.013969  26.452  < 2e-16 ***
## indiv        0.222795   0.021595  10.317  < 2e-16 ***
## mas          0.244858   0.008143  30.071  < 2e-16 ***
## ua          -0.534176   0.013590 -39.305  < 2e-16 ***
## lto         -0.483764   0.016041 -30.158  < 2e-16 ***
## indul       -0.002516   0.013145  -0.191 0.848203
## ggr         -0.184345   0.013150 -14.019  < 2e-16 ***
## invpro      -0.460481   0.010503 -43.841  < 2e-16 ***
## SIZE        -0.006982   0.005361  -1.302 0.192786
## BTM         -0.034559   0.004684  -7.378 1.61e-13 ***
## LEV         -0.014918   0.004513  -3.305 0.000948 ***
## ROA         -0.020072   0.004390  -4.573 4.82e-06 ***
## MEET1        0.015266   0.008982   1.700 0.089207 .
## LOSS1        0.176988   0.013648  12.968  < 2e-16 ***
## strongftr1  -0.105764   0.026846  -3.940 8.16e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Phylogenetic test

Much of the data is linked to the Indo-European language family. We can use a phylogenetic tree (Bourckaert et al., 2012) to investigate the relationship between AAM and FTR when taking more fine-grained distinctions in linguistic history.

Subset of variables for the indo-european language family:

```
dIE = d[d$mainLanguageFamily=="Indo-European",]
dIE$DPlaceLang =
  countryMainLanguageFamily[
    match(as.character(dIE$loc),
          countryMainLanguageFamily$Country.Code),
    ]$DPlaceLang
```

Load tree and drop languages that are not in the dataset:

```
tree = read.nexus(file = "../data/raw/trees/bouckaert_et_al2012-d-place_2.NEXUS")
dplaceLangs = countryMainLanguageFamily$DPlaceLang[countryMainLanguageFamily$DPlaceLang!=""]
tree = drop.tip(tree,tree$tip.label[!tree$tip.label %in% dplaceLangs])
lx = read.csv("../data/raw/langftr.csv",stringsAsFactors = F)
countryMainLanguageFamily[countryMainLanguageFamily$FTR=="",]$FTR =
  c("Weak","Strong")[lx[match(countryMainLanguageFamily[countryMainLanguageFamily$FTR=="",]$Country.Code
           lx$loc),]$strongftr+1]
```

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

Collapse AAM and FTR within languages, and scale and center the AAM variable.

```
DP.FTR = factor(tapply(dIE$strongftr,dIE$DPlaceLang,head,n=1))
DP.LTO = scale(tapply(dIE$lto,dIE$DPlaceLang,mean,na.rm=T))
DP.AAM = scale(tapply(dIE$AAM,dIE$DPlaceLang,mean,na.rm=T))

cdata = data.frame(
  FTR = DP.FTR,
  AAM = DP.AAM,
  LTO = DP.LTO,
  lang = names(DP.FTR)
)
cdata = cdata[cdata$lang!="",]
```

Run a regression using the phylogenetic tree as a variance-covariance matrix.

```
# Priors
prior.PN<-list(
  G=list(
    G1=list(V=1,nu=0.002)),
  R=list(V=1,nu=0.002))
# Chain length
burnin = 100000
postBurnin =100000
thin = 10
# Run the model
set.seed(1289)
phyloModel0<-MCMCglmm(
  AAM ~ FTR,
  random=~lang,
```

```
  ginverse=list(
    lang=inverseA(tree)$Ainv),
  prior = prior.PN,
  verbose=FALSE,
  family="gaussian",
  data = cdata,
  nitt=burnin+postBurnin,
  thin=thin,
  burnin=burnin)
```

Results:

```
summary(phyloModel0)
```

```
##
##  Iterations = 100001:199991
##  Thinning interval  = 10
##  Sample size  = 10000
##
##  DIC: 24.39859
##
##  G-structure:  ~lang
##
##      post.mean  l-95% CI u-95% CI eff.samp
## lang     1.395 0.0002236    4.025    578.1
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units     0.597 0.0001479    1.568    494.9
##
##  Location effects: AAM ~ FTR
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   -0.7944  -2.4719   0.6958   1485.2 0.310
## FTR1           0.9332  -0.4488   2.3000    905.9 0.188
```

There is no significant relationship between AAM and FTR.

Do the same test for Long-Term Orientation:

```
set.seed(12829)
phyloModelLTO<-MCMCglmm(
  AAM ~ LTO,
  random=~lang,
  ginverse=list(
    lang=inverseA(tree)$Ainv),
  prior = prior.PN,
  verbose=FALSE,
  family="gaussian",
  data = cdata,
  nitt=burnin+postBurnin,
  thin=thin,
  burnin=burnin)
summary(phyloModelLTO)
```

```
##
##  Iterations = 100001:199991
##  Thinning interval  = 10
##  Sample size  = 10000
##
##  DIC: 49.44665
##
##  G-structure:  ~lang
##
##      post.mean  l-95% CI u-95% CI eff.samp
## lang    0.4891 0.0001196    2.849    446.1
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units    0.8295 0.0003182    1.571     1090
##
##  Location effects: AAM ~ LTO
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)  -0.02423 -0.78404  0.72067     9630 0.9426
## LTO          -0.44768 -0.89869 -0.01900     8875 0.0476 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

OLS with clustered errors

Run a robust OLS regression, then print the results when clustering standard errors by language family. The results below were run using STATA code:

```
# No clustering
. reg AAM_scaled strongftr1 invpro pd
    indiv mas ua lto indul ggr size btm
    lev roa meet1 loss1,
    robust

Linear regression
    Number of obs     =     94,707
    F(15, 94691)      =     451.86
    Prob > F          =     0.0000
    R-squared         =     0.0731
    Root MSE          =     .96283

----------------------------------------------------------------------
            |           Robust
 AAM_scaled |    Coef.  Std. Err.    t   P>|t|   [95% Conf. Interval]
------------+---------------------------------------------------------
 strongftr1 | .1317505  .0033255  39.62  0.000   .1252326    .1382685

# With clustering by language family
. reg AAM_scaled strongftr1 invpro pd
    indiv mas ua lto indul ggr size btm
    lev roa meet1 loss1,
    robust cluster(mainLanguageFamily)

Linear regression
    Number of obs     =     94,707
    F(7, 8)           =          .
    Prob > F          =          .
    R-squared         =     0.0731
    Root MSE          =     .96283

(Std. Err. adjusted for 9 clusters in mainLanguageFamily)
----------------------------------------------------------------------
             |           Robust
  AAM_scaled |    Coef.  Std. Err.    t   P>|t|  [95% Conf. Interval]
-------------+--------------------------------------------------------
  strongftr1 | .1317505   .079045   1.67  0.134  -.0505275    .3140286
```