Language and earings management: controlling for linguistic history

Contents

Introduction

Test the relationship between strong/weak FTR and AAM, with and without controls for language family.

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)
```

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:

```
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 varaibles:

```
d2Orig = d2
# Take log of AAM
d2$logAAM = log(1+d2$AAM)
#d2$logAAM = d2$logAAM - median(d2$logAAM,na.rm = T)
# 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)
```

Mixed effects modelling

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) 0.01484  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   0.063448   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:

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

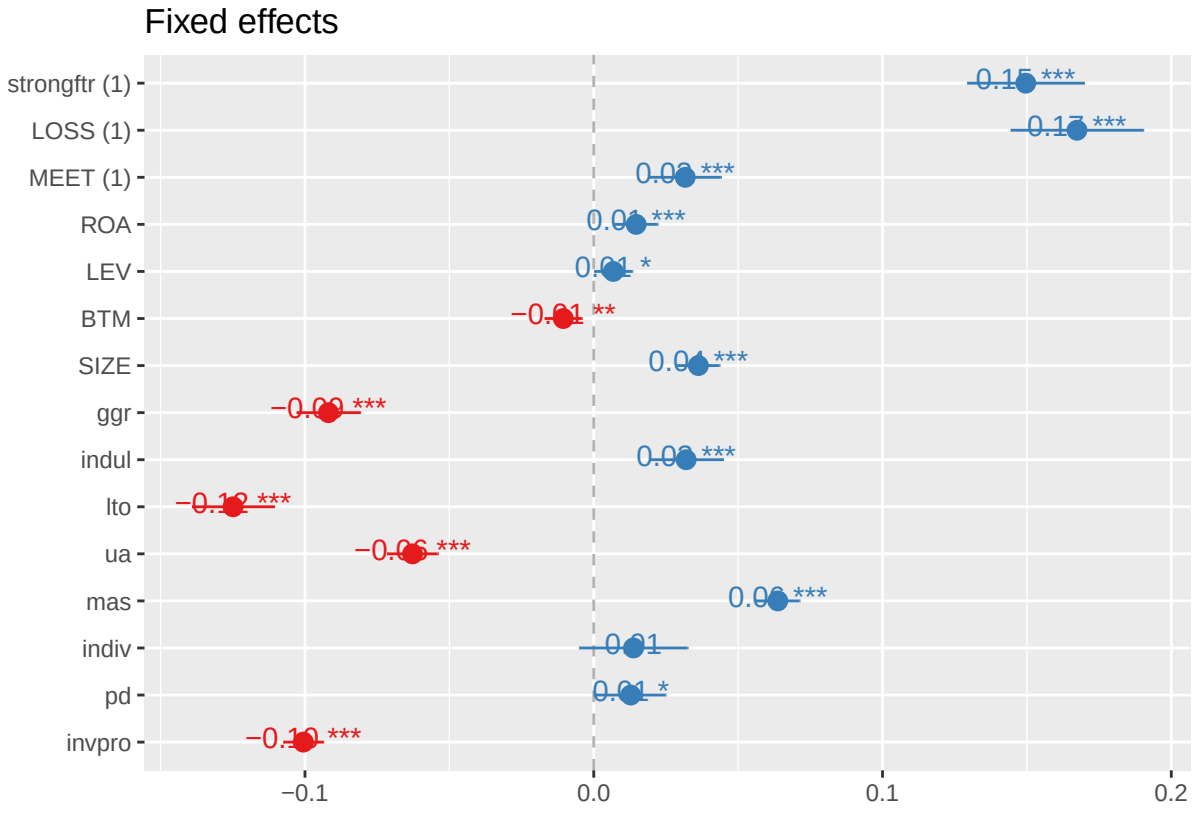
```
## refitting model(s) with ML (instead of REML)
```

```
## 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:

```
sjp.lmer(mA1,type="fe",p.kr = F)
```

```
## Computing p-values via Wald-statistics approximation (treating t as Wald z).
```



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) 0.10016  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   0.124554   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         -0.220394   0.009085 -24.259
## indul        0.016509   0.007498   2.202
## ggr         -0.102036   0.005758 -17.720
## SIZE         0.020040   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:

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

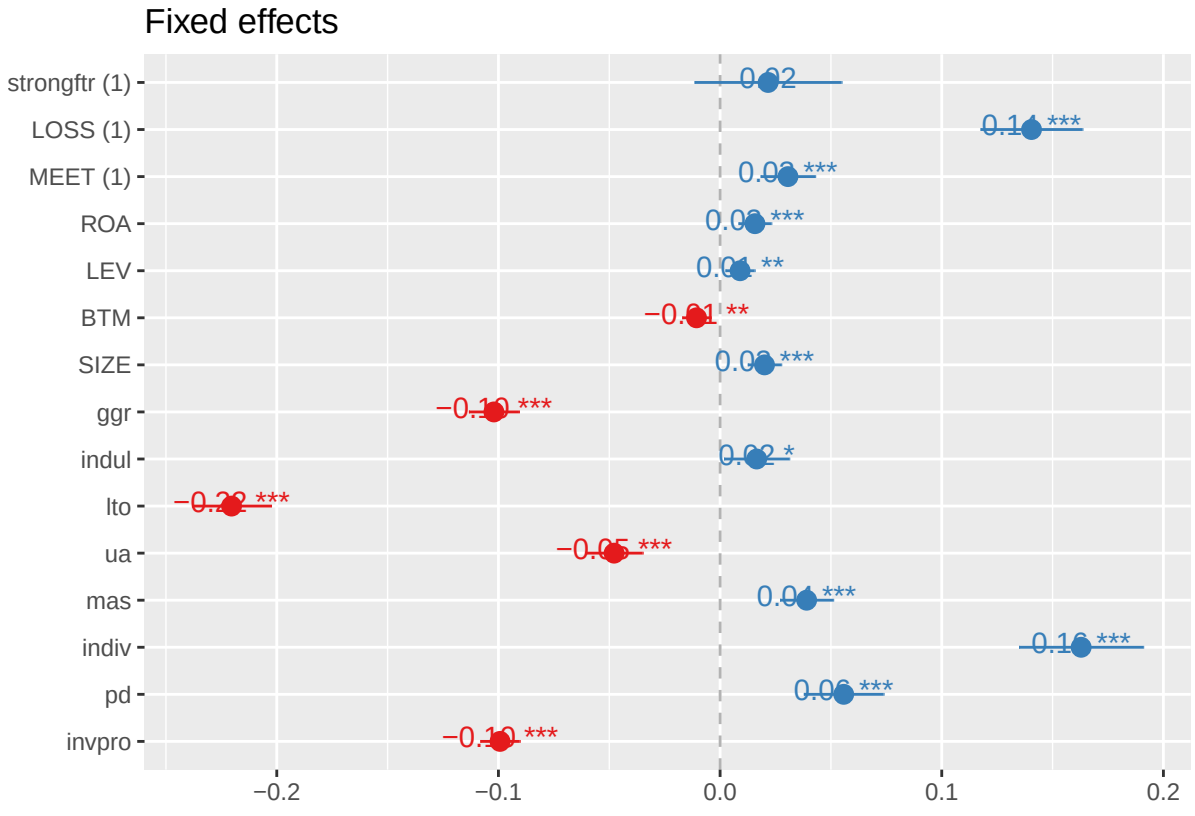
```
## refitting model(s) with ML (instead of REML)
```

```
## 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:

```
sjp.lmer(mB1,type="fe",p.kr = F)
```

```
## Computing p-values via Wald-statistics approximation (treating t as Wald z).
```



Random slopes for FTR

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

```
mB2 = lmer(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)
```

```
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:

```
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) 0.04226  0.2056
##  indus              (Intercept) 0.01570  0.1253
##  mainLanguageFamily (Intercept) 0.04331  0.2081
##                     strongftr1  0.07240  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)  0.2949671  0.1012065   2.915
```

```
## invpro      -0.1061720  0.0045830 -23.166
## pd           0.0499995  0.0093404   5.353
## indiv        0.1890531  0.0147841  12.788
## mas          0.0464442  0.0062111   7.478
## ua          -0.0248142  0.0071356  -3.477
## lto         -0.2406900  0.0094705 -25.415
## indul        0.0007693  0.0076795   0.100
## ggr         -0.0950665  0.0058169 -16.343
## SIZE         0.0181446  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  -0.0366087  0.1089072  -0.336
```

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

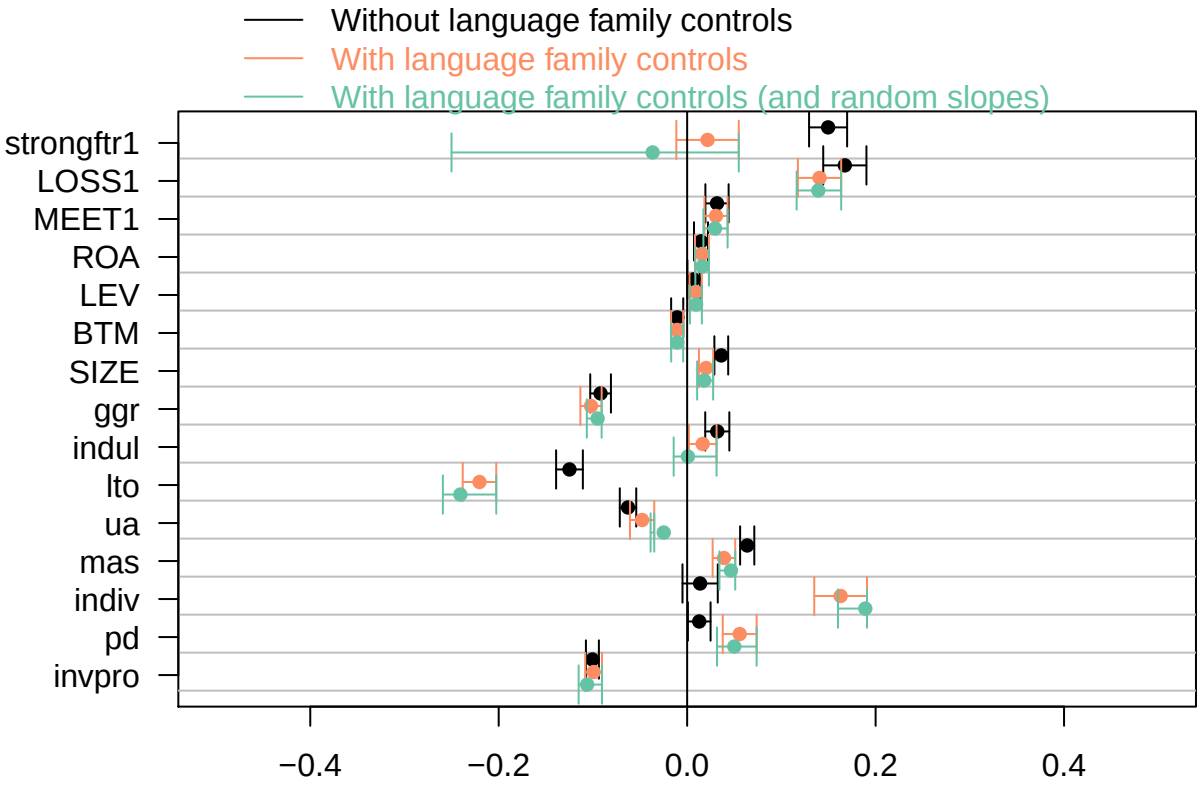
Calculate p-value for effect of FTR:

```
mB2_noFTR = update(mB2, ~. - strongftr)
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 both models (code hidden):




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

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

Summary

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).

Other effects

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

```
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(mA1, paste("~ . -",v))
  mAOtherAnova = anova(mAOther0,mA1)
  mBOther0 = update(mB1, paste("~ . -",v))
  mBOtherAnova = anova(mBOther0,mB1)
  mARes = getMEText(mAOtherAnova,"X",summary(mA1)$coef[v,],returnText = F)
  mBRes = getMEText(mBOtherAnova,"X",summary(mB1)$coef[v,],returnText = F)
  resOther = rbind(resOther, c(paste(v,": No controls"),mARes))
  resOther = rbind(resOther, c(paste(v,": With Controls"),mBRes))
}
```

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

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

```
##                   Label  Beta loglikDiff df chisq.test       p
## 2      pd : No controls 0.013        2.2  1       4.36   0.037
## 3    pd : With Controls 0.056         18  1      36.94 1.2e-09
## 4   indiv : No controls 0.014          1  1       2.08    0.15
## 5 indiv : With Controls  0.16         65  1     129.88 4.3e-30
```

```
## 6      mas : No controls  0.064        140  1     275.51  7.1e-62
## 7    mas : With Controls  0.039         20  1      40.87  1.6e-10
## 8       ua : No controls -0.063         99  1     198.09  5.4e-45
## 9     ua : With Controls -0.048         26  1      52.77  3.7e-13
## 10     lto : No controls  -0.12        150  1     296.83  1.6e-66
## 11   lto : With Controls  -0.22        290  1     581.64 1.6e-128
## 12   indul : No controls  0.032         12  1      24.01  9.6e-07
## 13 indul : With Controls  0.017        2.4  1       4.87    0.027
## 14     ggr : No controls -0.092        130  1     268.69  2.2e-60
## 15   ggr : With Controls   -0.1        160  1     313.39    4e-70
## 16    SIZE : No controls  0.036         47  1      93.62  3.8e-22
## 17  SIZE : With Controls   0.02         14  1      27.15  1.9e-07
## 18     BTM : No controls -0.011        5.2  1      10.41   0.0013
## 19   BTM : With Controls -0.011        5.3  1      10.57   0.0012
## 20     LEV : No controls 0.0068          2  1       4.09    0.043
## 21   LEV : With Controls  0.009        3.6  1       7.23   0.0072
## 22     ROA : No controls  0.015        7.5  1      15.04  0.00011
## 23   ROA : With Controls  0.016        8.7  1      17.35  3.1e-05
```

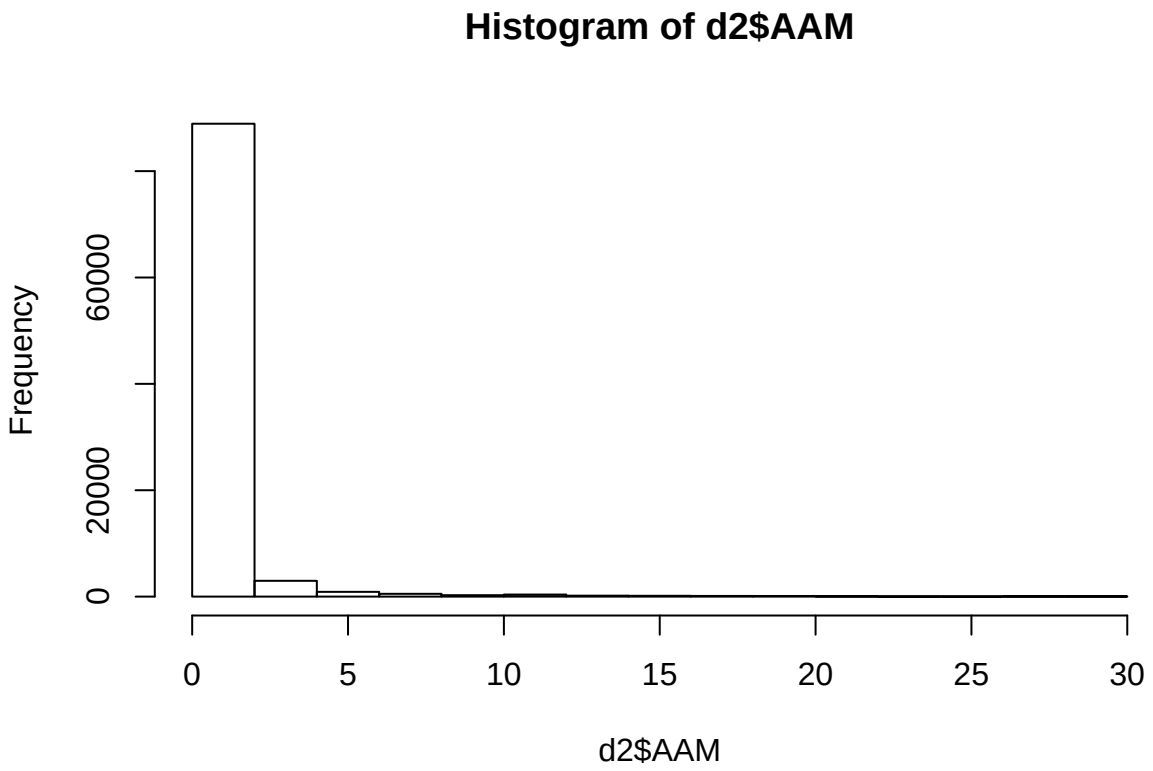
```
resOther2 = cbind(
  resOther[seq(1,nrow(resOther)-1,by=2),c("Label","Beta","p")],
  resOther[seq(2,nrow(resOther),by=2),c("Beta","p")])
write.csv(resOther2,"../results/BetaResults_OtherVariables.csv",row.names = F)
```

Alternative tests

Gamma distribution model

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

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



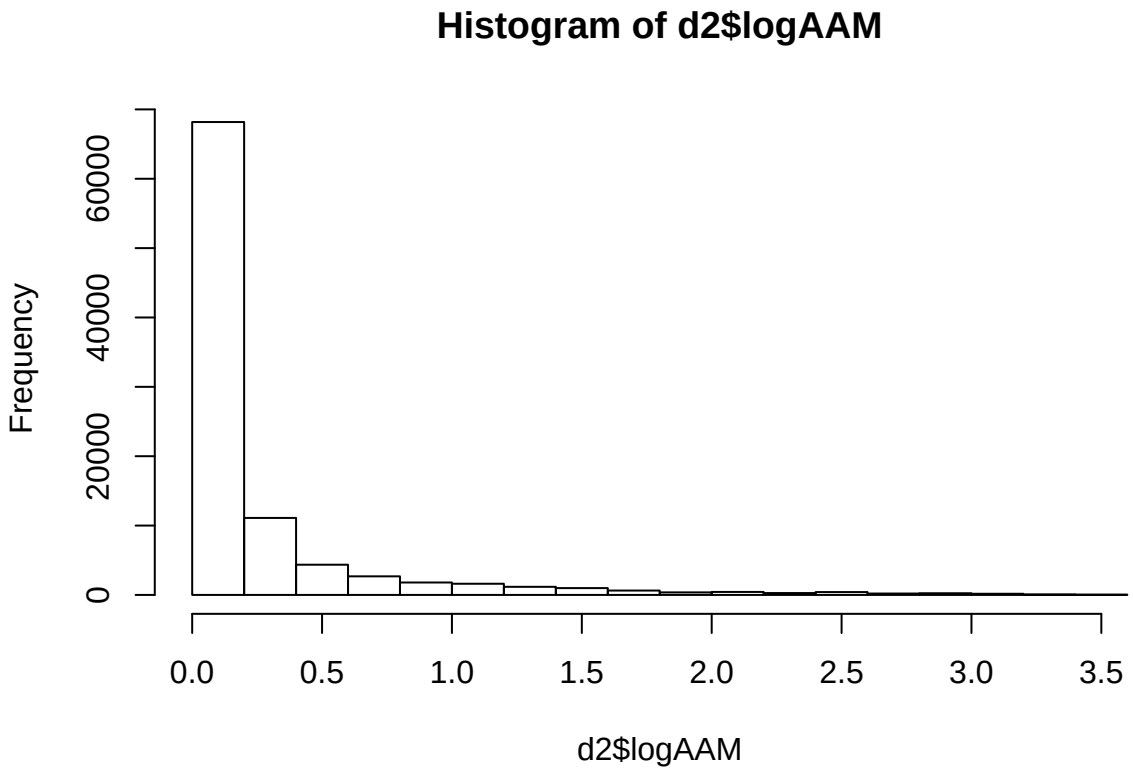

```
normalDist = rnorm(n=length(d2$AAM),
                   mean = mean(d2$AAM),
                   sd = sd(d2$AAM))
ks.test(d2$AAM,normalDist)
```

```
## 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 = 0.3916, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Even the log-transformed variable is skewed:

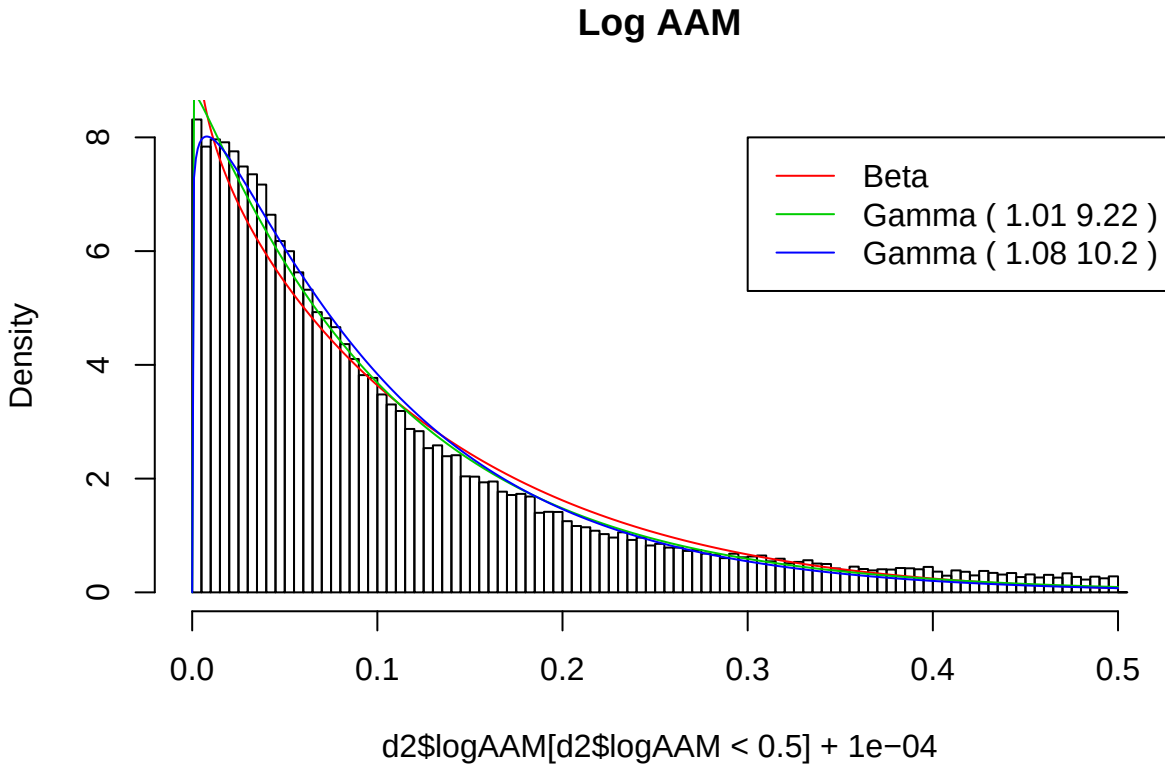
```
hist(d2$logAAM)
```



This leads to the models above (which use Gaussian distributions) producing very poor fits:

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

Here we compare how Beta and Gamma distributions fit the log data:



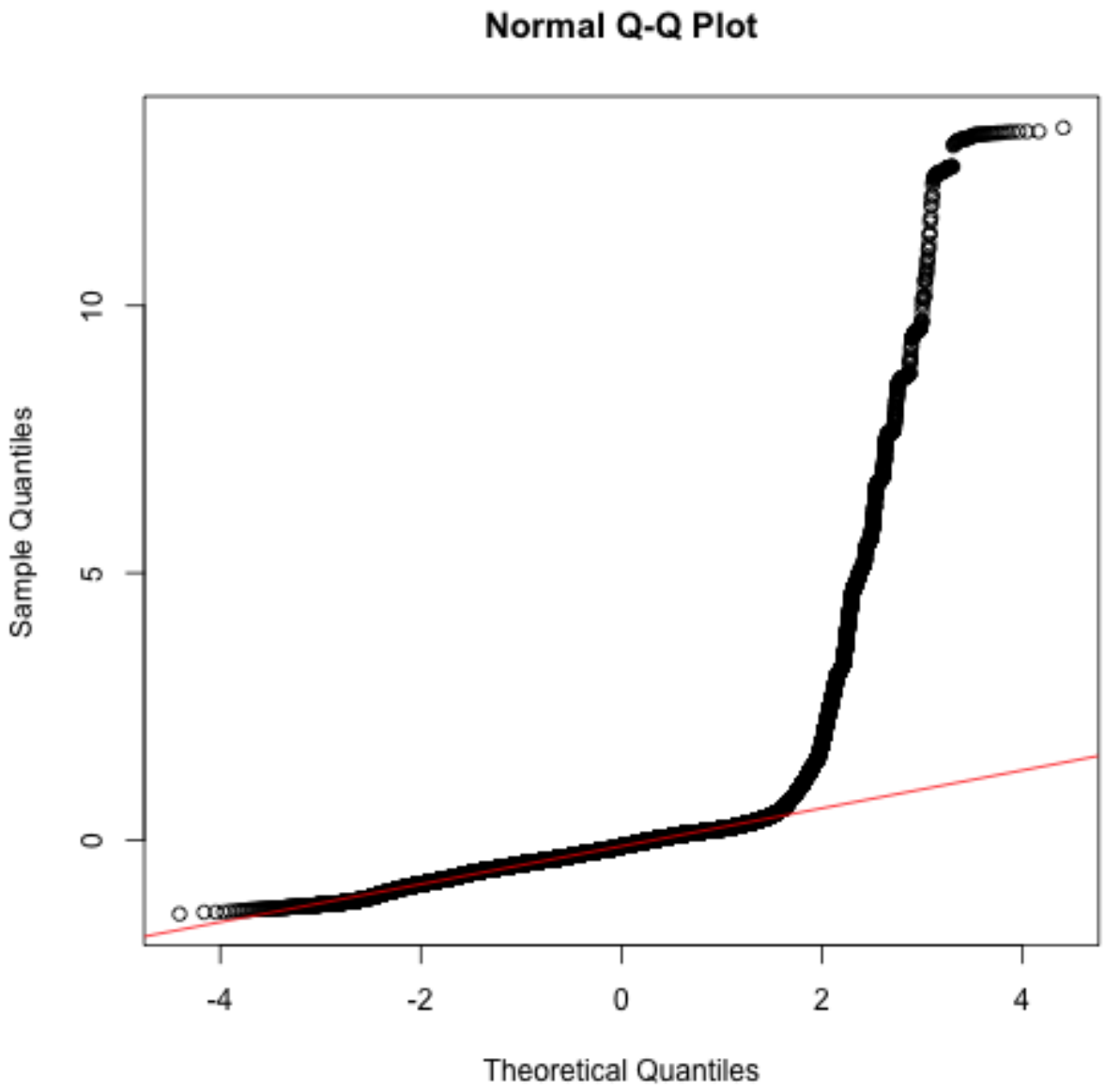

Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
0
5
10
-4
-2
0
2
4

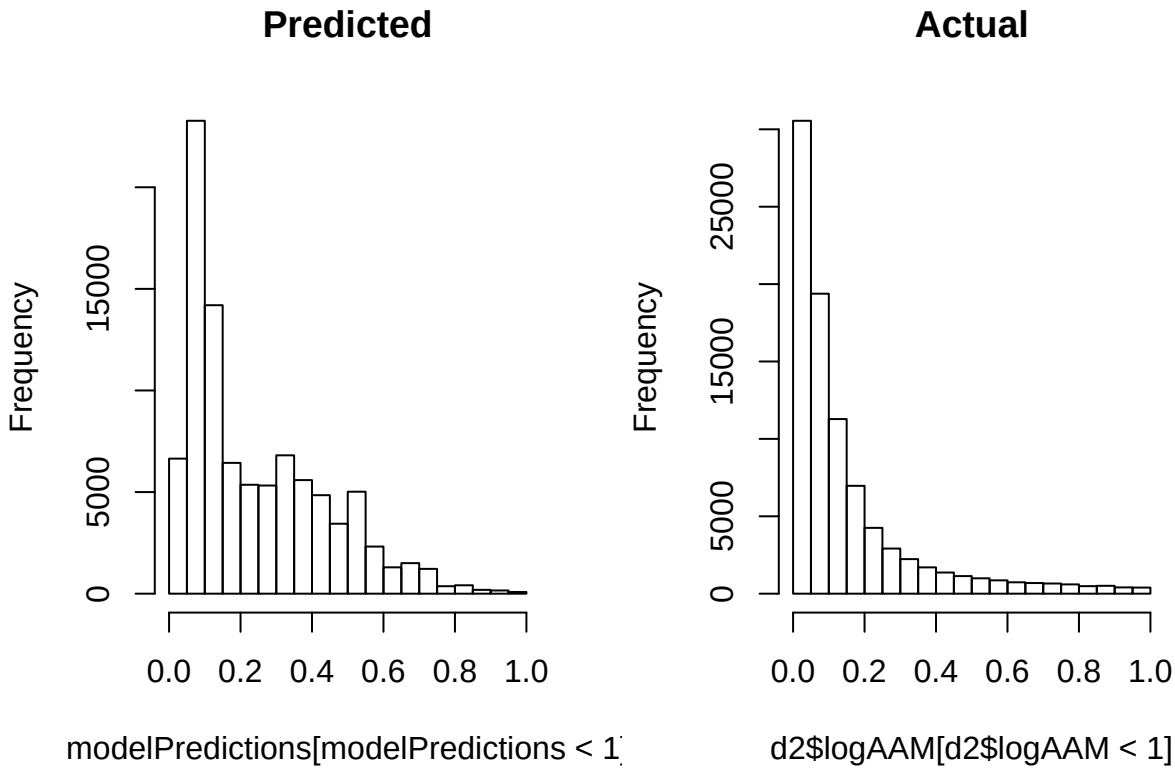
Figure 1:

The Gamma distribution seems to fit best. Fit a model with a Gamma distribution (without language family controls, with a random intercept by family and with a random intercept and slope):

```
mA1Gamma = glmer(logAAM+0.0001 ~ 1 +
              invpro +
              pd + indiv + mas + ua + lto + indul +
              ggr +
              SIZE + BTM + LEV + ROA +
              MEET + LOSS +
              strongftr +
              (1 | fyear) +
              (1 | indus),
            data = d2,
            family=Gamma(link="log"))
mB1Gamma = update(mA1Gamma, ~.+(1 | mainLanguageFamily))
mB2Gamma= update(mB1Gamma,
          ~.+(0+strongftr|mainLanguageFamily))
```

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
```

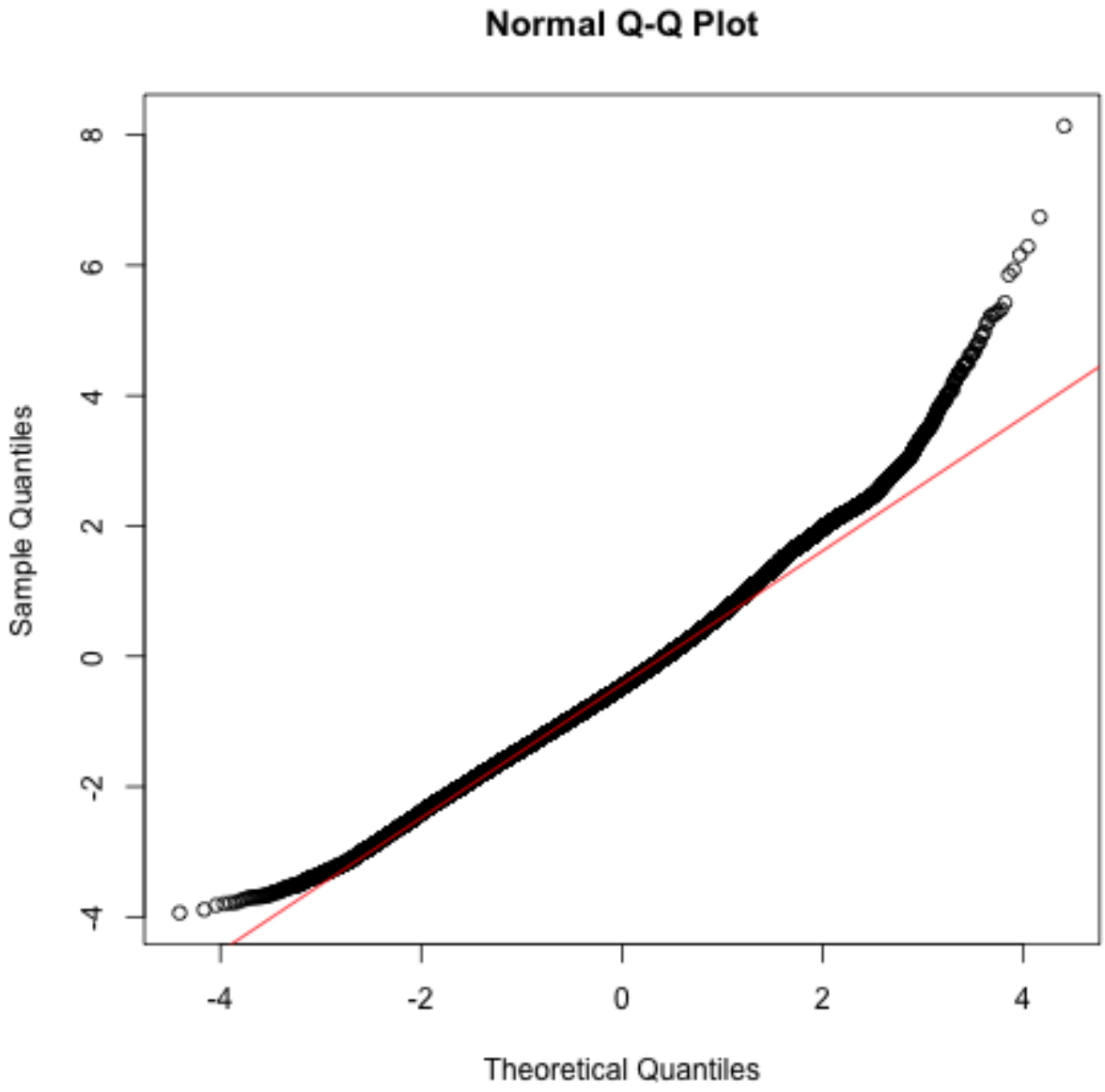


Figure 2:

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

Model results:

```
summary(mA1Gamma)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula:
```

```
## logAAM + 1e-04 ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
##     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus)
##    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) -1.6280301  0.1266208  -12.86  < 2e-16 ***
## invpro      -0.2124855  0.0043007  -49.41  < 2e-16 ***
## pd          -0.0867836  0.0075234  -11.54  < 2e-16 ***
## indiv       -0.0513334  0.0122957   -4.17 2.98e-05 ***
## mas          0.1074145  0.0047423   22.65  < 2e-16 ***
## ua          -0.1914821  0.0055205  -34.69  < 2e-16 ***
## lto         -0.3641583  0.0094341  -38.60  < 2e-16 ***
## indul        0.0719161  0.0075237    9.56  < 2e-16 ***
## ggr         -0.0838851  0.0064592  -12.99  < 2e-16 ***
## SIZE         0.0083879  0.0042610    1.97    0.049 *
## BTM         -0.0446026  0.0034390  -12.97  < 2e-16 ***
## LEV         -0.0021147  0.0036797   -0.57    0.565
## ROA          0.0003773  0.0039579    0.10    0.924
## MEET1        0.0521151  0.0071332    7.31 2.75e-13 ***
## LOSS1        0.3089023  0.0131382   23.51  < 2e-16 ***
## strongftr1   0.5389408  0.0118434   45.51  < 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(mB1Gamma)
```

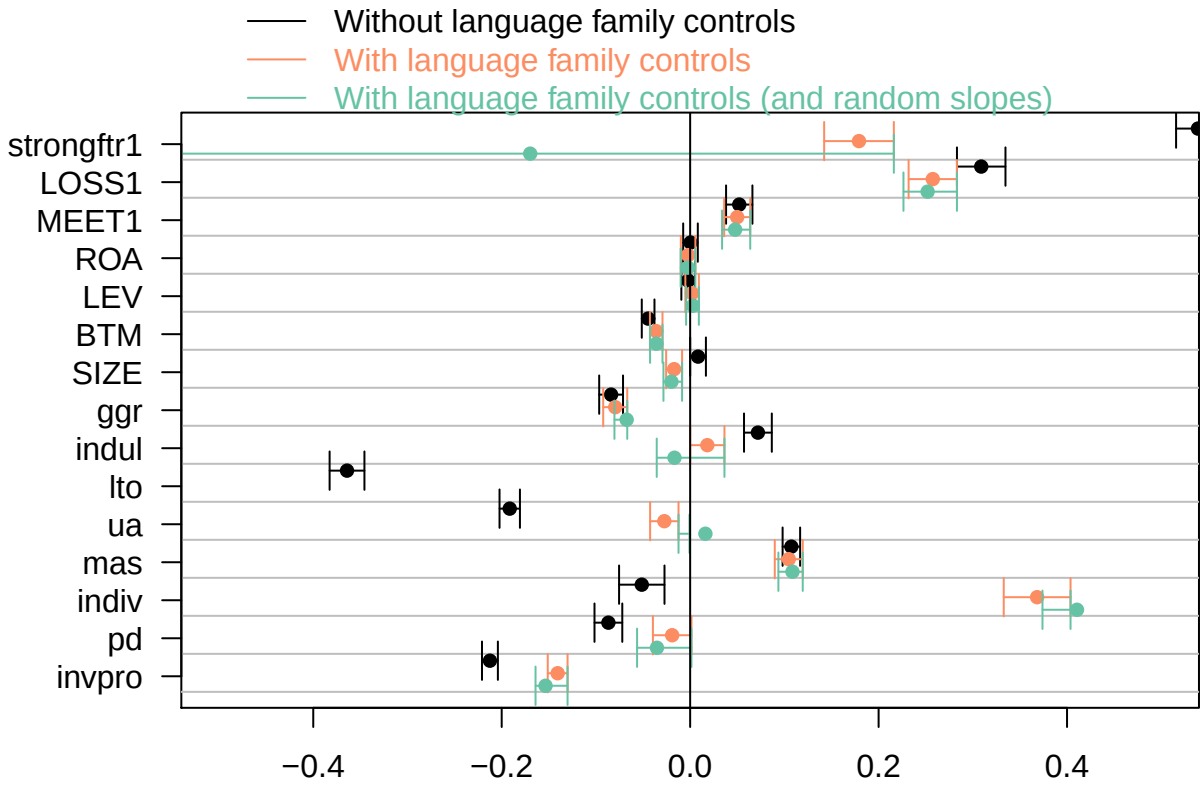
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: logAAM + 1e-04 ~ invpro + pd + indiv + mas + ua + lto + indul +
##     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus) + (1 | mainLanguageFamily)
##    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 30.7850
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  fyear              (Intercept) 0.1944   0.4409
##  indus              (Intercept) 0.1172   0.3424
##  mainLanguageFamily (Intercept) 1.0053   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) -1.810443   0.306027   -5.92 3.30e-09 ***
## invpro      -0.140605   0.005364  -26.21  < 2e-16 ***
## pd          -0.019084   0.010442   -1.83 0.067622 .
## indiv        0.368243   0.018088   20.36  < 2e-16 ***
## mas          0.104654   0.007575   13.82  < 2e-16 ***
## ua          -0.027409   0.007689   -3.56 0.000364 ***
## lto         -0.576459   0.010866  -53.05  < 2e-16 ***
## indul        0.018232   0.009269    1.97 0.049184 *
## ggr         -0.079520   0.006515  -12.21  < 2e-16 ***
## SIZE        -0.017009   0.004316   -3.94 8.11e-05 ***
## BTM         -0.036027   0.003404  -10.58  < 2e-16 ***
## LEV          0.002056   0.003653    0.56 0.573598
## ROA         -0.002173   0.003933   -0.55 0.580592
## MEET1        0.049931   0.007059    7.07 1.51e-12 ***
## LOSS1        0.257498   0.013082   19.68  < 2e-16 ***
## strongftr1   0.179186   0.018865    9.50  < 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 ~ invpro + pd + indiv + mas + ua + lto + indul +
##     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus) + (1 | mainLanguageFamily) + (0 +
##     strongftr | mainLanguageFamily)
##    Data: d2
##
##       AIC       BIC    logLik  deviance  df.resid
## -138420.8 -138203.2   69233.4 -138466.8     94684
##
```

```
## 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) 0.2067261 0.45467
##  indus                (Intercept) 0.1191341 0.34516
##  mainLanguageFamily   (Intercept) 0.0003559 0.01887
##  mainLanguageFamily.1 strongftr0  0.6283060 0.79266
##                       strongftr1  1.7129177 1.30878  0.94
##  Residual                         1.4226490 1.19275
## Number of obs: 94707, groups:  fyear, 20; indus, 9; mainLanguageFamily, 9
##
## Fixed effects:
##              Estimate Std. Error t value Pr(>|z|)
## (Intercept) -1.620158   0.266232   -6.09 1.16e-09 ***
## invpro      -0.153494   0.005415  -28.35  < 2e-16 ***
## pd          -0.035222   0.010760   -3.27  0.00106 **
## indiv        0.410556   0.018690   21.97  < 2e-16 ***
## mas          0.108590   0.007622   14.25  < 2e-16 ***
## ua           0.016180   0.008635    1.87  0.06096 .
## lto         -0.612745   0.011464  -53.45  < 2e-16 ***
## indul       -0.016607   0.009616   -1.73  0.08417 .
## ggr         -0.067387   0.006620  -10.18  < 2e-16 ***
## SIZE        -0.019870   0.004320   -4.60 4.24e-06 ***
## BTM         -0.035865   0.003402  -10.54  < 2e-16 ***
## LEV          0.002873   0.003653    0.79  0.43153
## ROA         -0.002389   0.003930   -0.61  0.54320
## MEET1        0.047749   0.007056    6.77 1.31e-11 ***
## LOSS1        0.252019   0.013080   19.27  < 2e-16 ***
## strongftr1  -0.169744   0.212595   -0.80  0.42461
## ---
## 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
```


Without language family controls
With language family controls
With language family controls (and random slopes)
strongftr1
LOSS1
MEET1
ROA
LEV
BTM
SIZE
ggr
indul
lto
ua
mas
indiv
pd
invpro
−0.4
−0.2
0.0
0.2
0.4

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

Decision tree

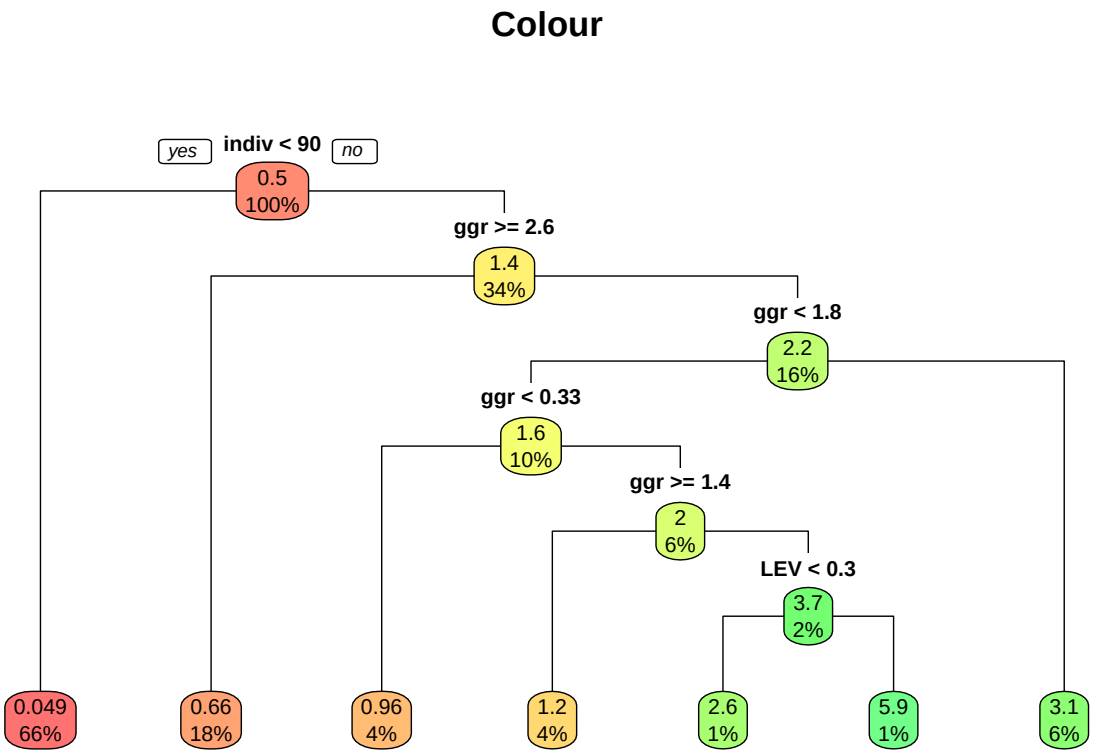
A decision tree is a machine learning technique that tries to find patterns in data. It finds a series of yes/no questions which divide datapoints into partitions that look similar. 'Variable importance' is a measure of how influential each variable is in making decisions in the tree. This is a useful 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`.

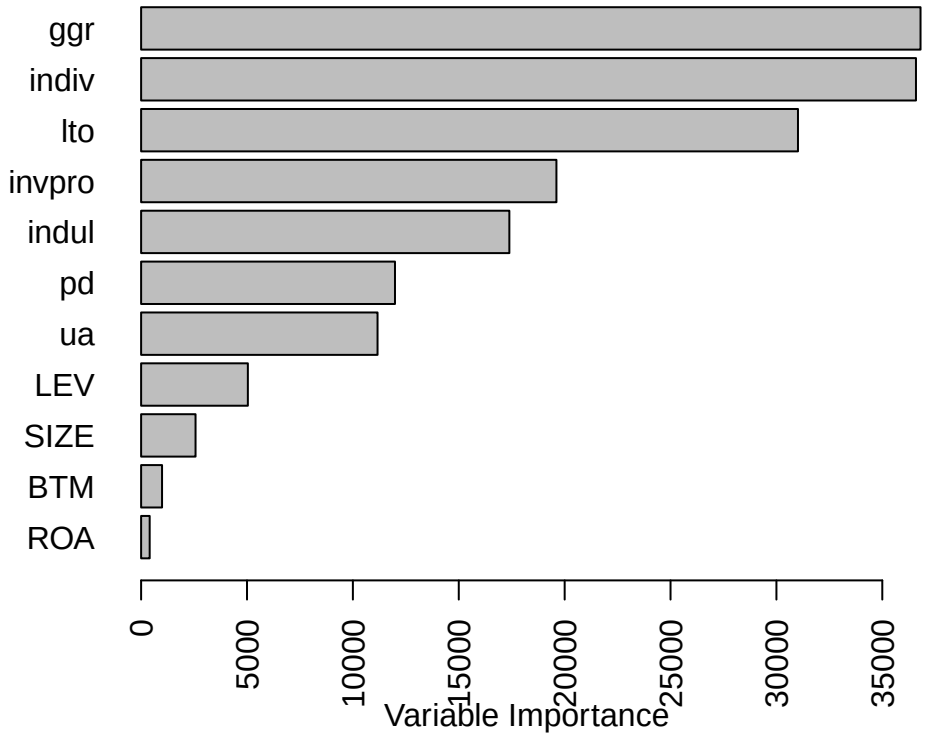
```
set.seed(1111) # set random seed for reproducability
rt = REEMtree(AAM ~
            strongftr +
            invpro +
            pd + indiv + mas + ua + lto + indul +
            ggr +
            SIZE + BTM + LEV + ROA +
            MEET + LOSS,
          data = d2Orig,
          random = ~1|mainLanguageFamily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

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




```
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)
```

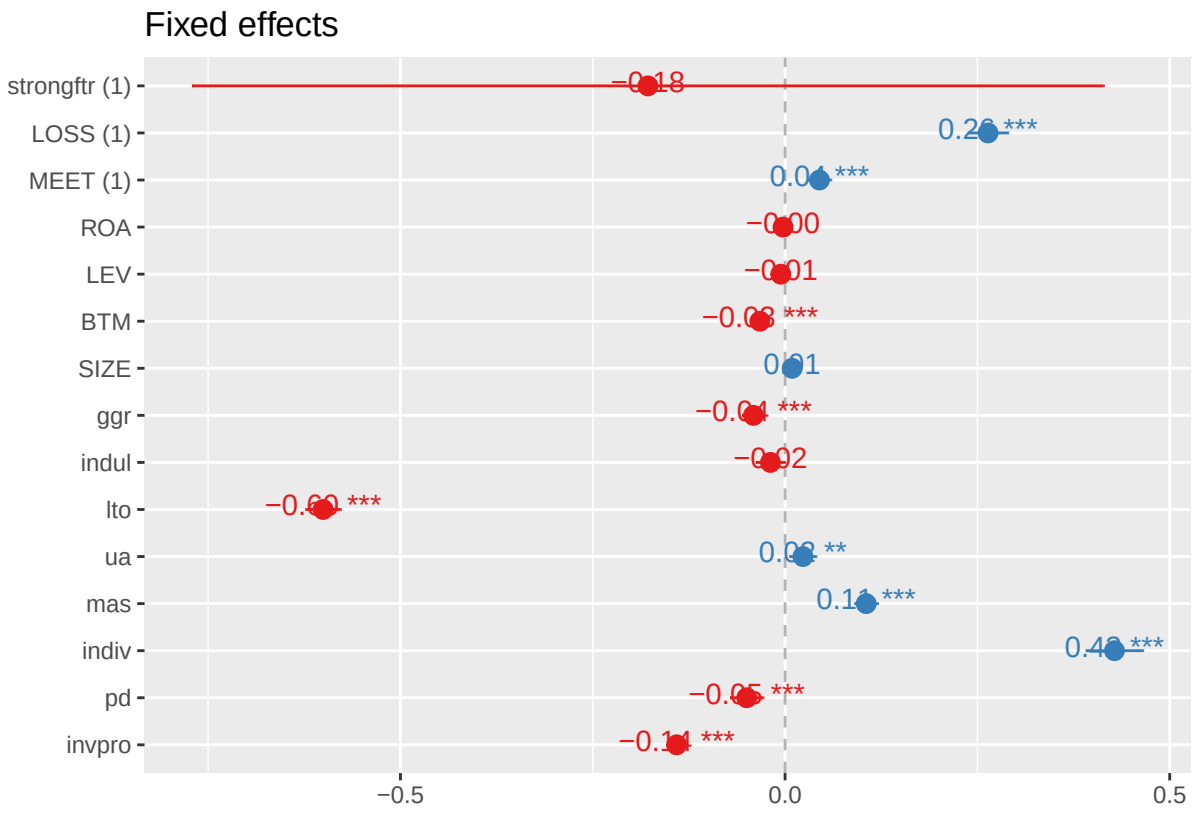
Random slopes

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)
```

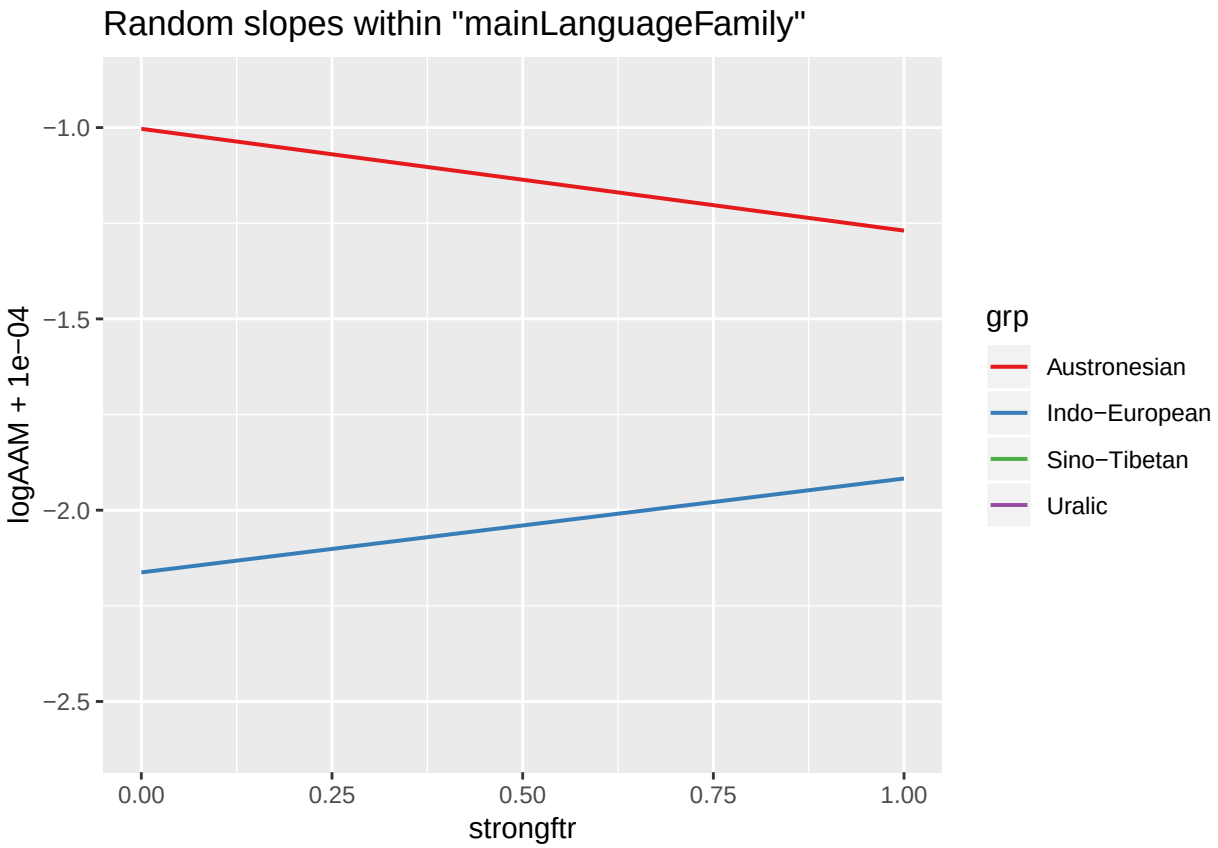
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.0013761 (tol =
## 0.001, component 1)
```

```
sjp.lmer(mB2GammaFamily,type="fe",p.kr = F)
```

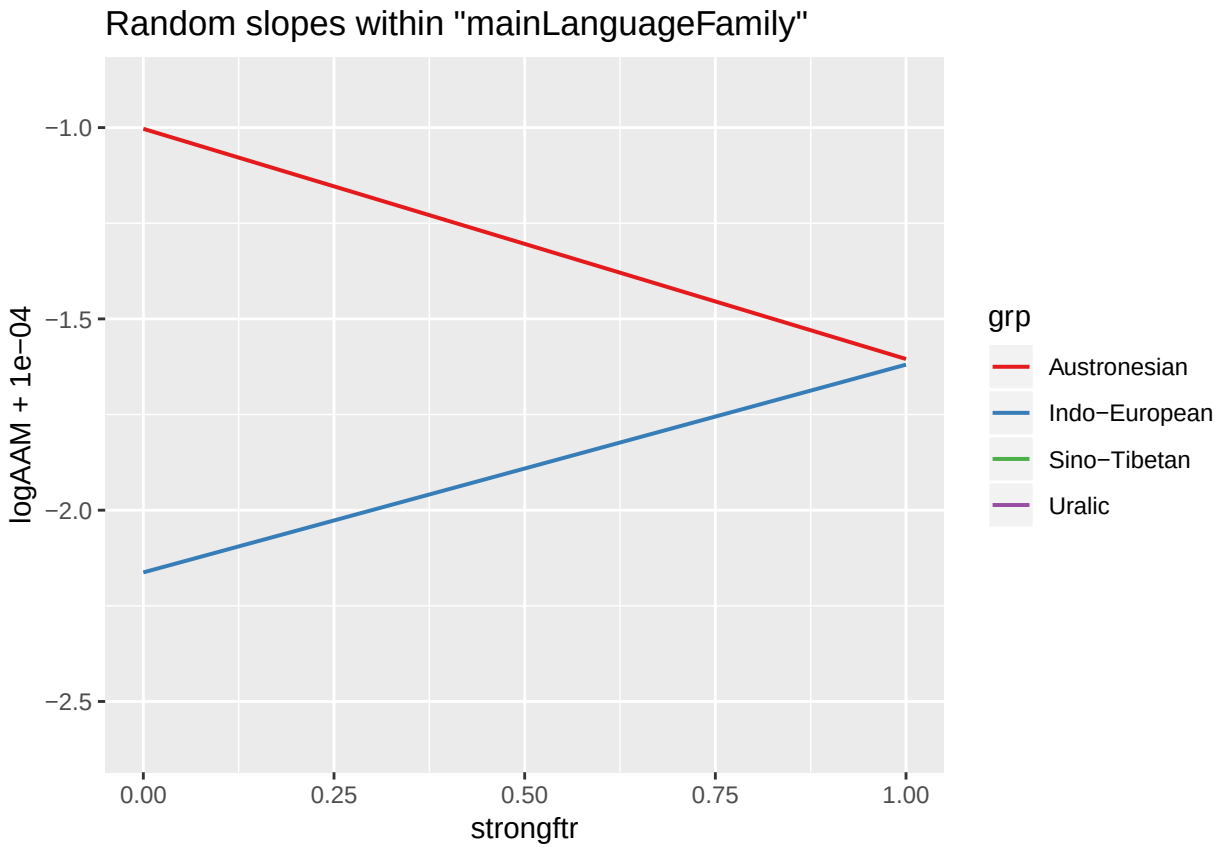



```
sjp.lmer(mB2GammaFamily,type="rs.ri",vars="strongftr",show.legend = T)
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

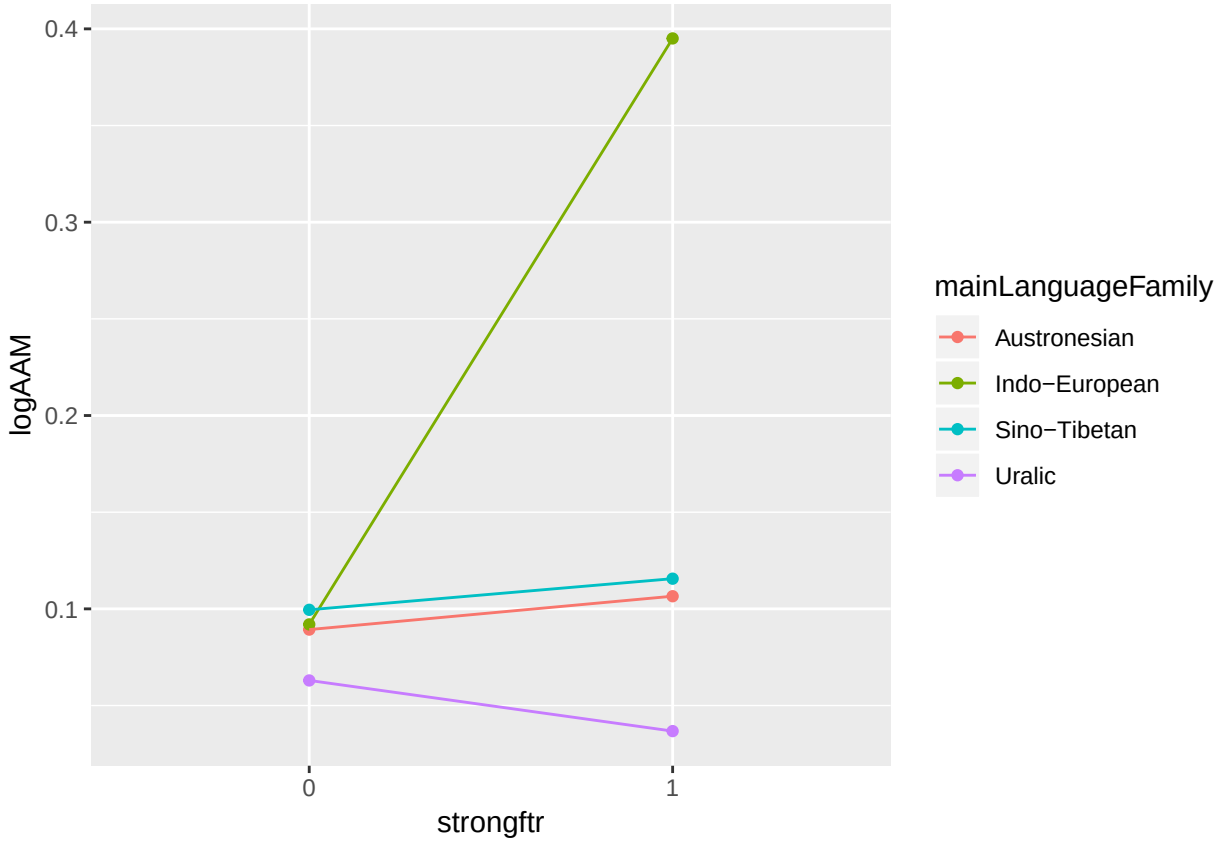
```
## The following object is masked from 'package:nlme':
##
##     collapse
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
x = d2[d2$mainLanguageFamily %in%
          c("Austronesian","Indo-European",
            "Sino-Tibetan","Uralic"),] %>%
  group_by(mainLanguageFamily,strongftr) %>%
  summarise(logAAM=mean(logAAM))
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 ~ 1 + invpro + pd + indiv + mas + ua + lto + indul +
##     ggr + SIZE + BTM + LEV + ROA + MEET + LOSS + strongftr +
##     (1 | fyear) + (1 | indus)
##    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) -0.326903   0.179472   -1.82 0.068535 .
## invpro      -0.460479   0.010504  -43.84  < 2e-16 ***
## pd           0.369501   0.013969   26.45  < 2e-16 ***
## indiv        0.222799   0.021596   10.32  < 2e-16 ***
## mas          0.244860   0.008143   30.07  < 2e-16 ***
## ua          -0.534175   0.013591  -39.30  < 2e-16 ***
## lto         -0.483771   0.016042  -30.16  < 2e-16 ***
## indul       -0.002515   0.013145   -0.19 0.848292
## ggr         -0.184346   0.013150  -14.02  < 2e-16 ***
## SIZE        -0.006981   0.005361   -1.30 0.192843
## BTM         -0.034560   0.004684   -7.38 1.61e-13 ***
## LEV         -0.014920   0.004513   -3.31 0.000947 ***
## ROA         -0.020073   0.004390   -4.57 4.81e-06 ***
## MEET1        0.015266   0.008982    1.70 0.089208 .
## LOSS1        0.176986   0.013648   12.97  < 2e-16 ***
## strongftr1  -0.105782   0.026847   -3.94 8.14e-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.Cod
           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
```