Codability across languages and domains

Contents

Introduction

This document provides R code for reproducing the test of the relative codability of the senses. We use mixed effects modelling to test the influence of language and domain. The full model included random intercepts for stimulus, domain, language, and the interaction between language and domain. Log-likelihood comparison was used to compare the full model to a model without one of those intercepts

Data

The main data comes from `../data/DiversityIndices_ND.csv`, which includes the Simpson's diversity index, counting no-responses as unique responses. Each row lists the codability of a particular stimulus for a particular community. The variables are:

- `Language`: Language/Community name
- `domain`: Sense domain
- `Stimulus.code`: Identity of the stimulus
- `simpson.diversityIndex`: Simpson's diversity index
- `shannon.diversityIndex`: Shannon diversity index
- `N`: Number of responses
- `BnL.diversityIndex`: Brown & Lenneberg diversity index
- `mean.number.of.words`: Mean number of words in full response

Load libraries

```
library(lme4)
library(sjPlot)
library(REEMtree)
library(ggplot2)
library(party)
library(reshape2)
library(rpart.plot)
library(lattice)
library(dplyr)
library(mgcv)
library(lmtest)
library(itsadug)
```

Load data

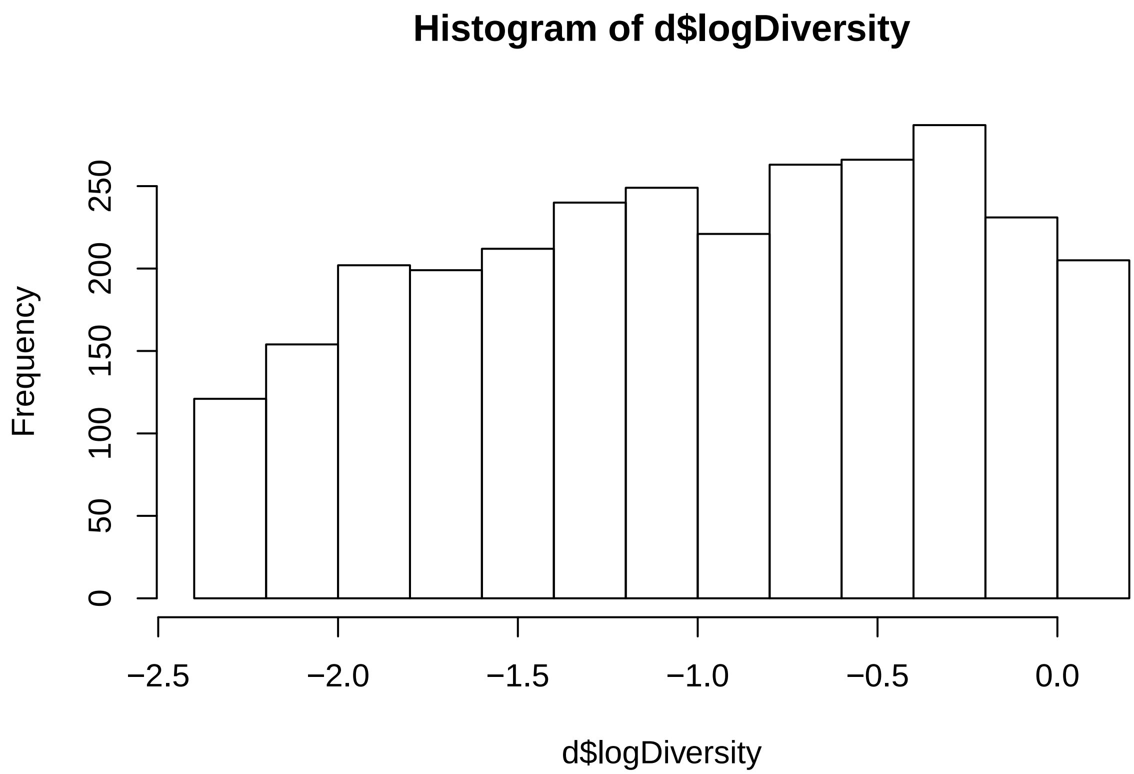
```
d = read.csv("../data/DiversityIndices_ND.csv", stringsAsFactors = F)

d = d[!is.na(d$simpson.diversityIndex),]

d$Language= as.factor(d$Language)
d$domain = factor(d$domain, levels=c("colour",'shape','sound','touch','taste','smell'))
```

Get log of diversity index, (add 0.1 to avoid infinite values)

```
d$logDiversity = log(d$simpson.diversityIndex+0.1)
```

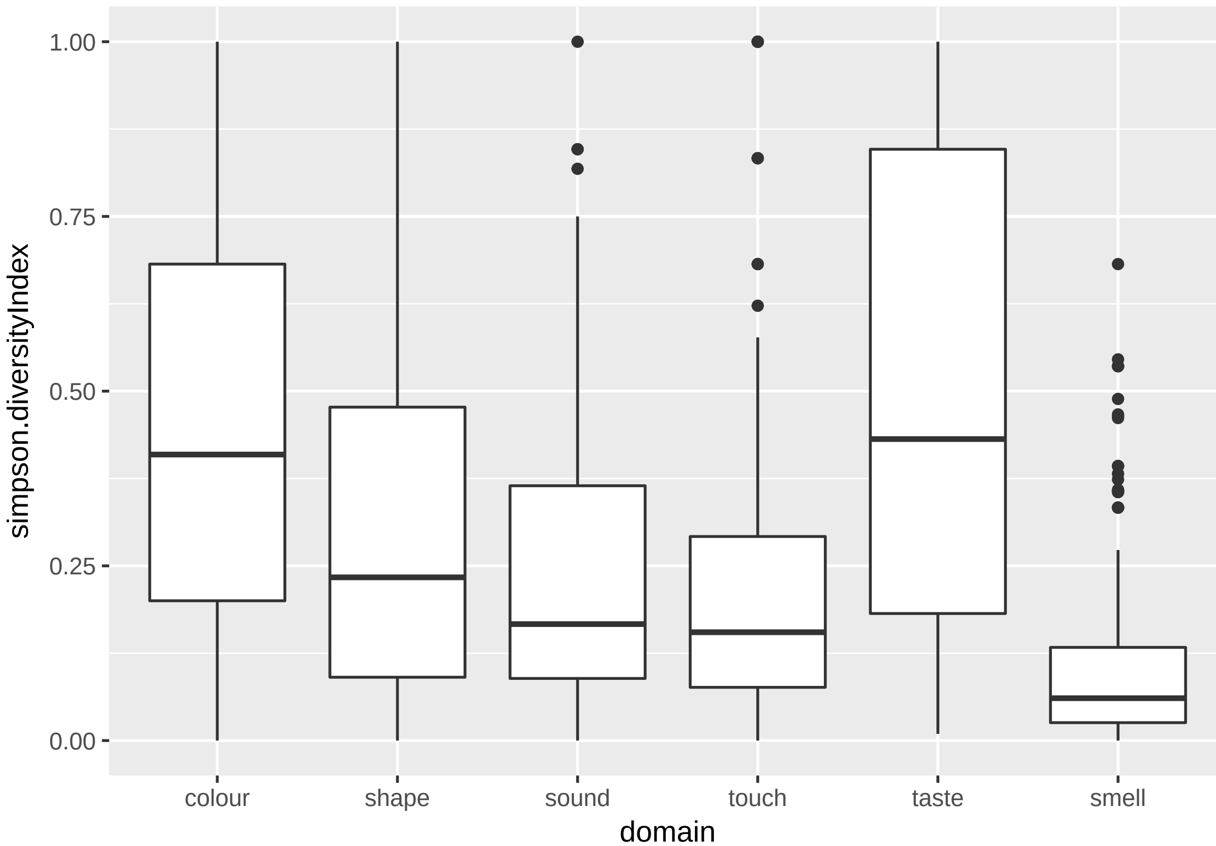
Distribution is not very normal, but it's difficult to approximate this distribution anyway.

```
hist(d$logDiversity)
```

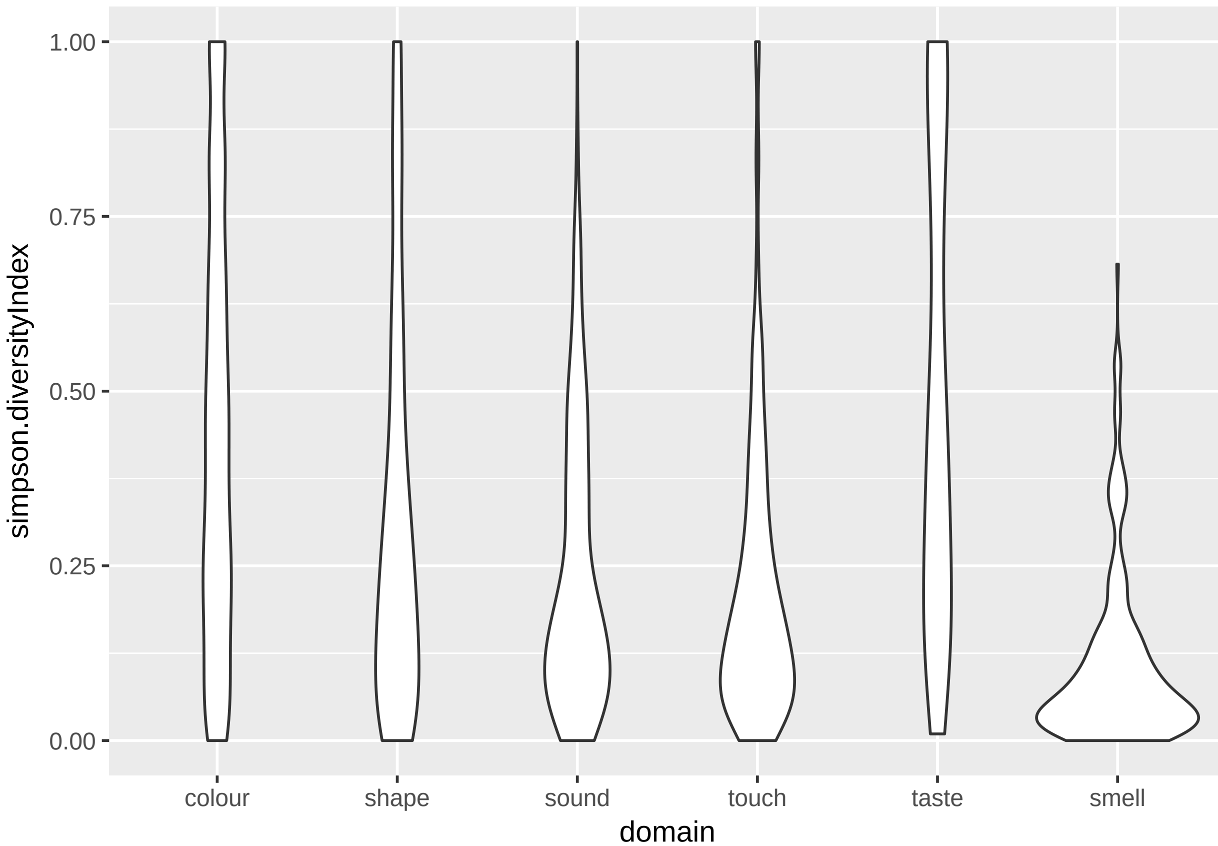


Plot data

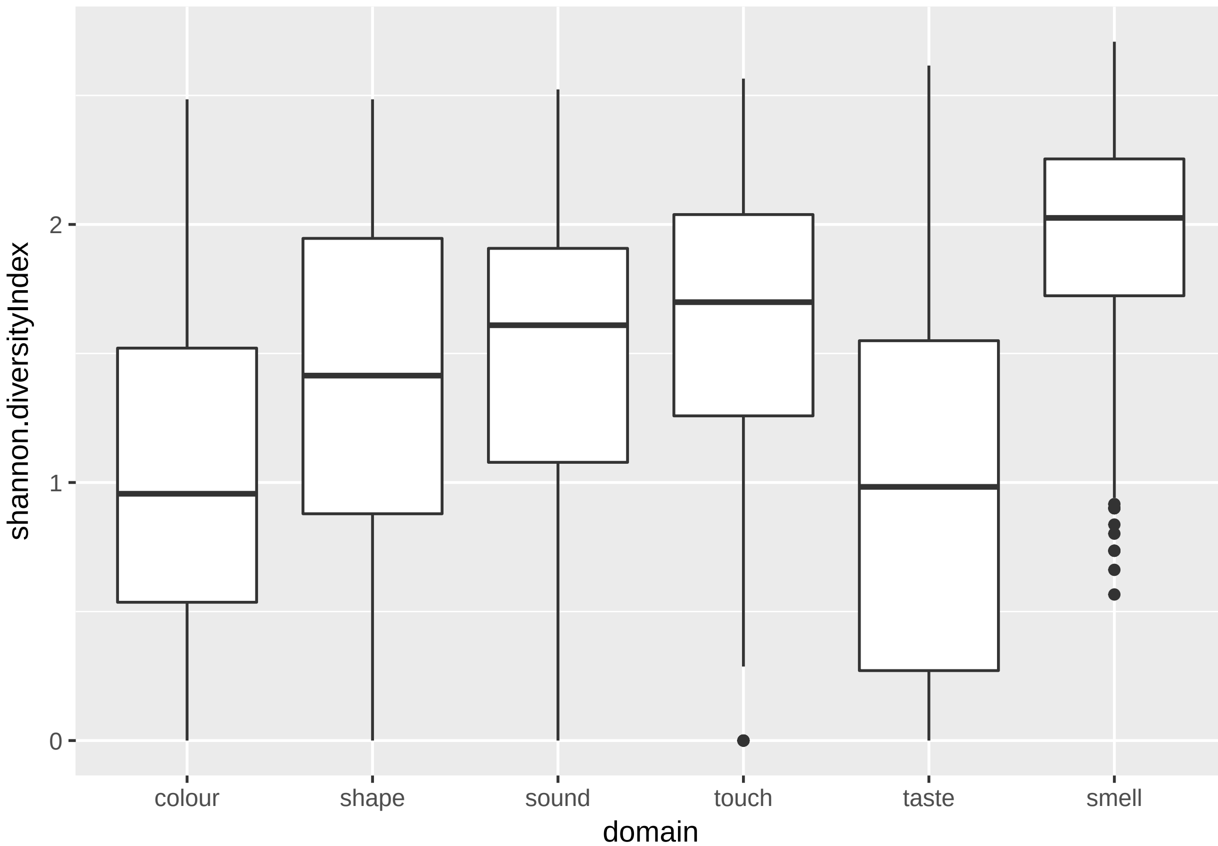
```
g = ggplot(d, aes(y=simpson.diversityIndex, x=domain))
g + geom_boxplot()
```



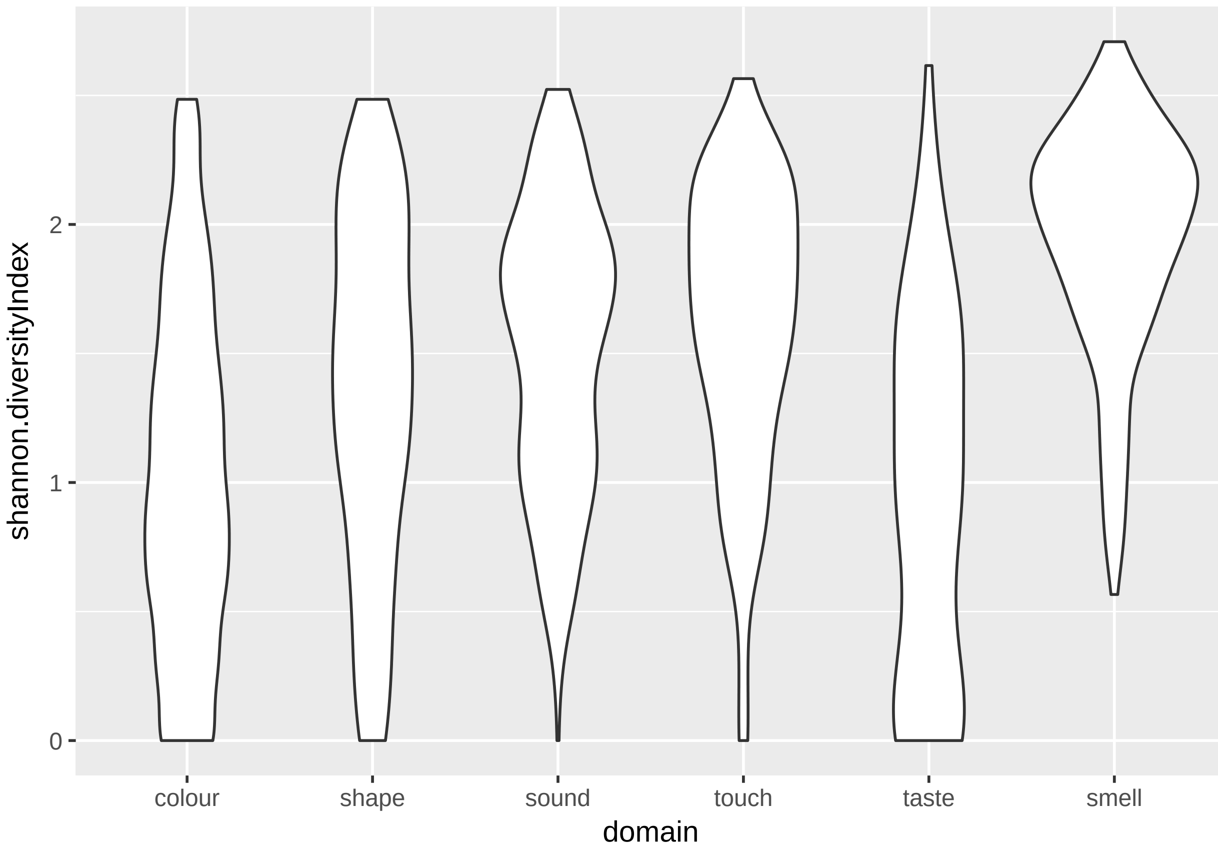

```
g + geom_violin()
```




```
g = ggplot(d, aes(y=shannon.diversityIndex, x=domain))
g + geom_boxplot()
```




```
g + geom_violin()
```



Run models

```
m.full = lmer( logDiversity ~ 1 +
              (1|Language) +
              (1|domain/Stimulus.code) +
              (1|Language:domain),
            data=d)

m.noL = lmer( logDiversity ~ 1 +
              (1|domain/Stimulus.code) +
              (1|Language:domain),
            data=d)

m.noDom = lmer( logDiversity ~ 1 +
              (1|Language) +
              (1|Stimulus.code) +
              (1|Language:domain),
            data=d)

m.noStim = lmer( logDiversity ~ 1 +
              (1|Language) +
              (1|domain) +
              (1|Language:domain),
            data=d)
```

```r
m.noLxD = lmer( logDiversity ~ 1 +
              (1|Language) +
              (1|domain/Stimulus.code),
            data=d)
```

Test models:

```r
anova(m.full, m.noL)
```

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

## Data: d
## Models:
## m.noL: logDiversity ~ 1 + (1 | domain/Stimulus.code) + (1 | Language:domain)
## m.full: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
## m.full:     (1 | Language:domain)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m.noL   5 3941.9 3971.6 -1965.9   3931.9
## m.full  6 3939.3 3975.0 -1963.7   3927.3 4.5491      1    0.03294 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.full, m.noDom)
```

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

## Data: d
## Models:
## m.noDom: logDiversity ~ 1 + (1 | Language) + (1 | Stimulus.code) + (1 |
## m.noDom:     Language:domain)
## m.full: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
## m.full:     (1 | Language:domain)
##         Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m.noDom  5 3965.1 3994.9 -1977.6   3955.1
## m.full   6 3939.3 3975.0 -1963.7   3927.3 27.827      1  1.326e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.full, m.noStim)
```

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

## Data: d
## Models:
## m.noStim: logDiversity ~ 1 + (1 | Language) + (1 | domain) + (1 | Language:domain)
## m.full: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
## m.full:     (1 | Language:domain)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m.noStim  5 4452.1 4481.9 -2221.1   4442.1
## m.full    6 3939.3 3975.0 -1963.7   3927.3 514.81      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.full, m.noLxD)
```

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

## Data: d
## Models:
```

```
## m.noLxD: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code)
## m.full: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
## m.full:     (1 | Language:domain)
##         Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m.noLxD  5 4637.3 4667.1 -2313.7   4627.3
## m.full   6 3939.3 3975.0 -1963.7   3927.3 700.03      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Details:

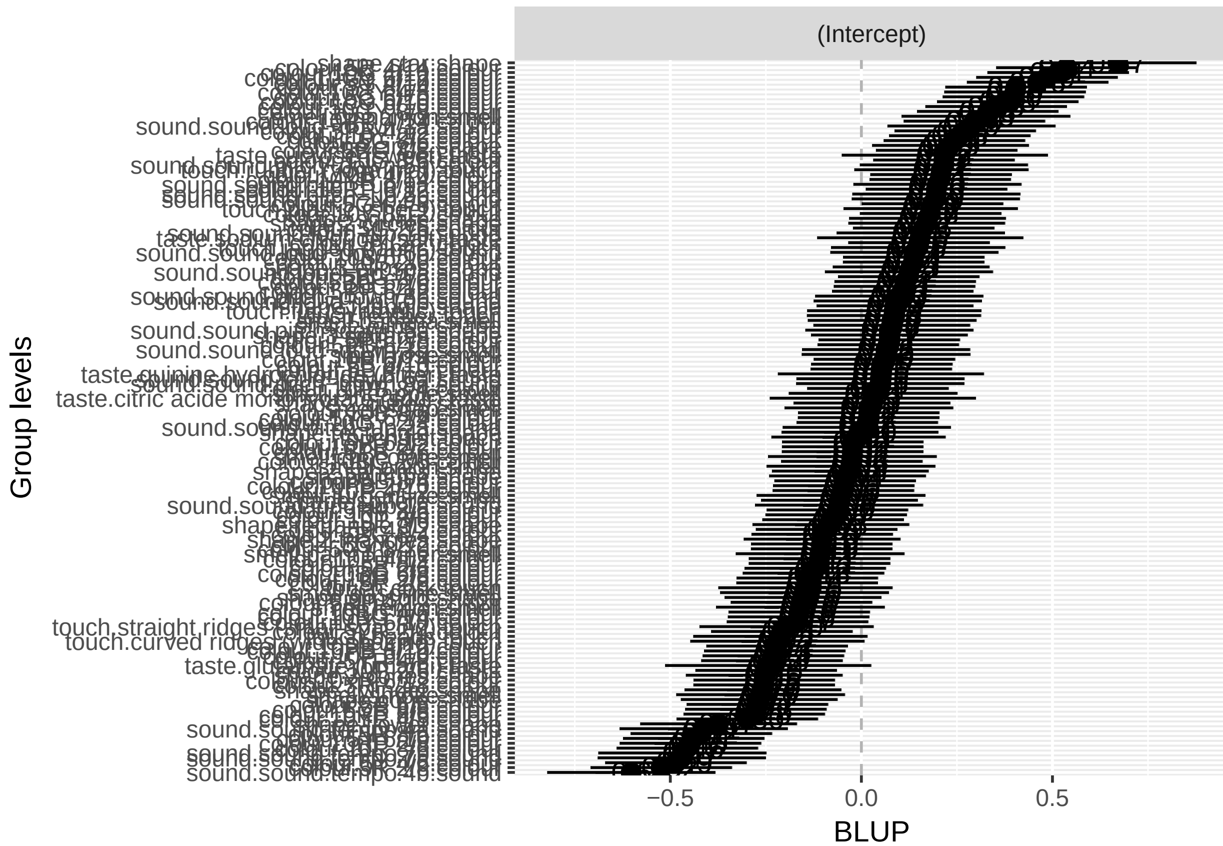
```r
summary(m.full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
##     (1 | Language:domain)
##    Data: d
##
## REML criterion at convergence: 3929.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0123 -0.6368  0.0153  0.6567  3.7445
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  Stimulus.code:domain (Intercept) 0.06416  0.2533
##  Language:domain      (Intercept) 0.13367  0.3656
##  Language             (Intercept) 0.02731  0.1653
##  domain               (Intercept) 0.12029  0.3468
##  Residual                         0.18754  0.4331
## Number of obs: 2850, groups:
## Stimulus.code:domain, 147; Language:domain, 114; Language, 20; domain, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  -1.1423     0.1536  -7.436
```

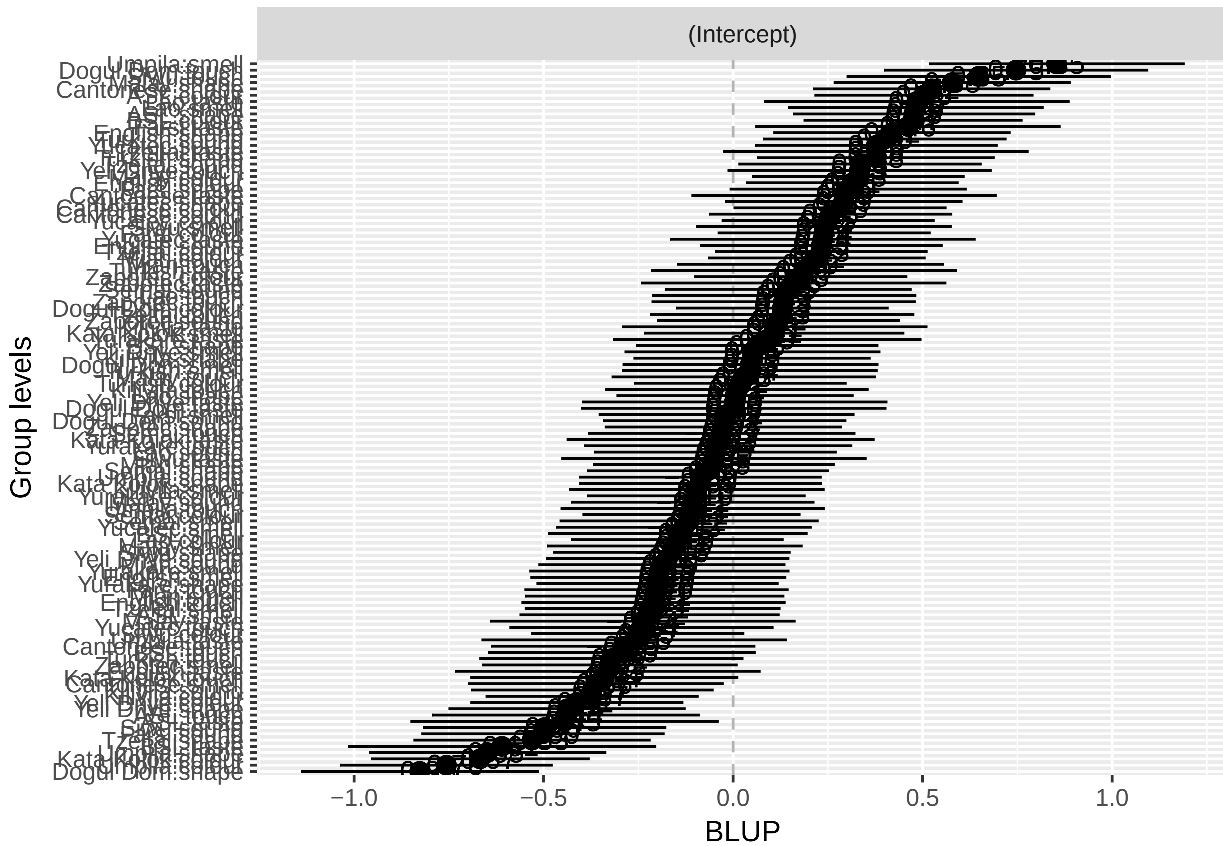
Random effects:

```r
sjp.lmer(m.full, 're', sort.est = T, geom.colors=c(1,1))
```

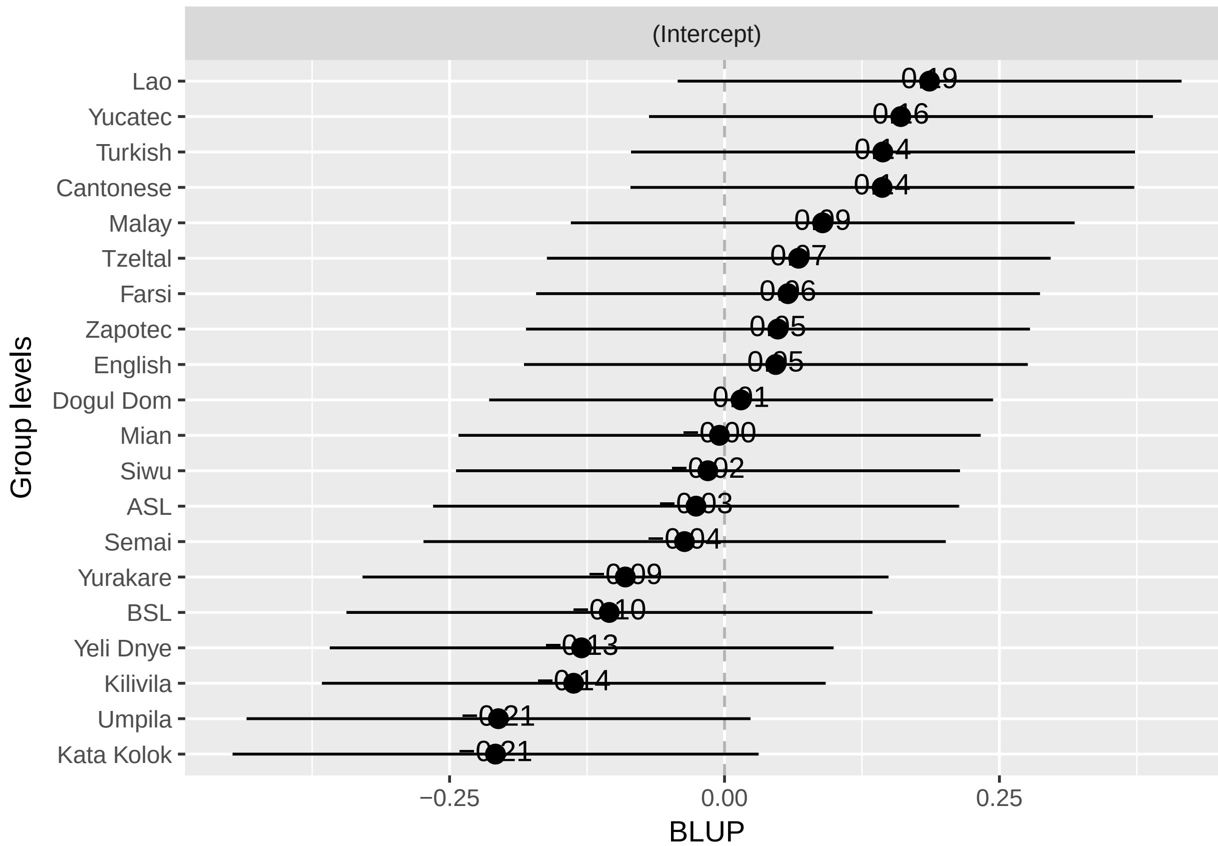
```
## Plotting random effects...
## Plotting random effects...
```

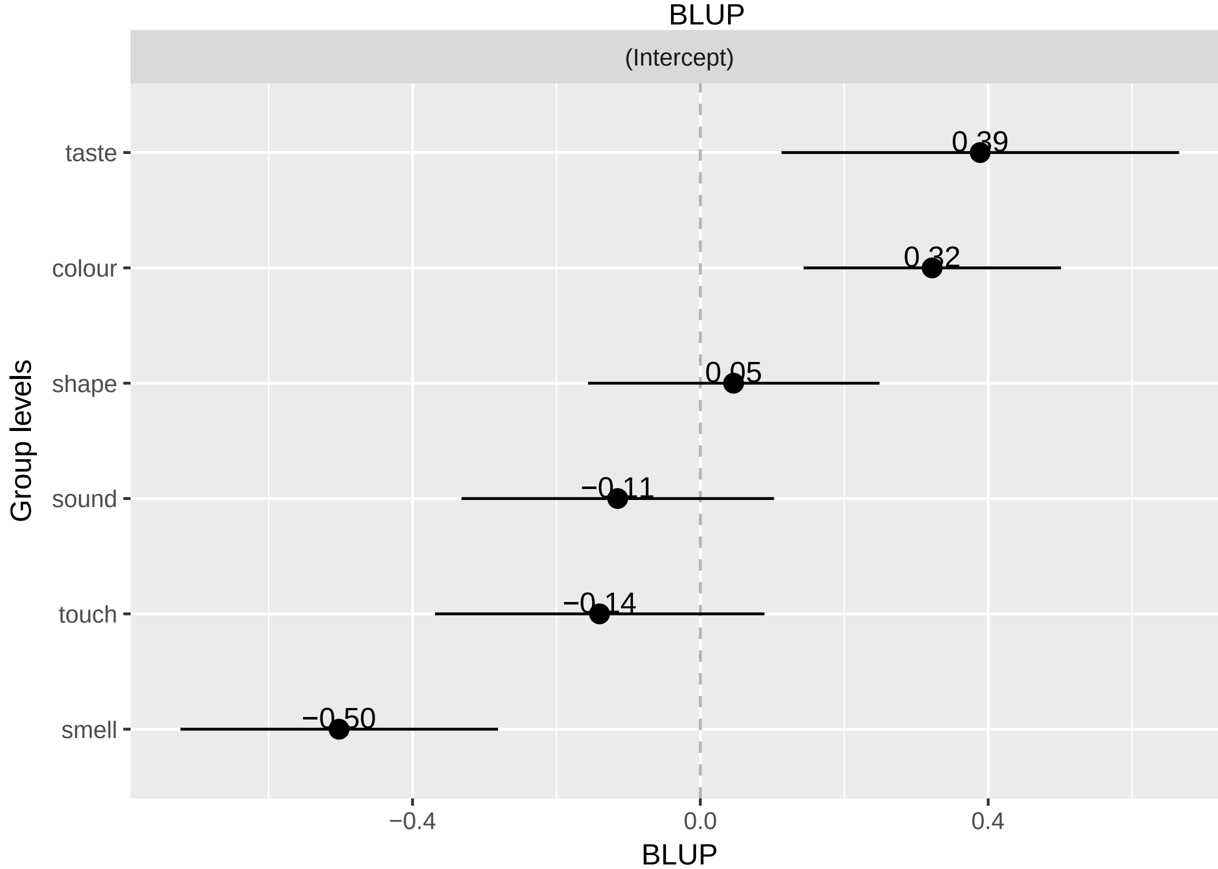



```
## Plotting random effects...
```

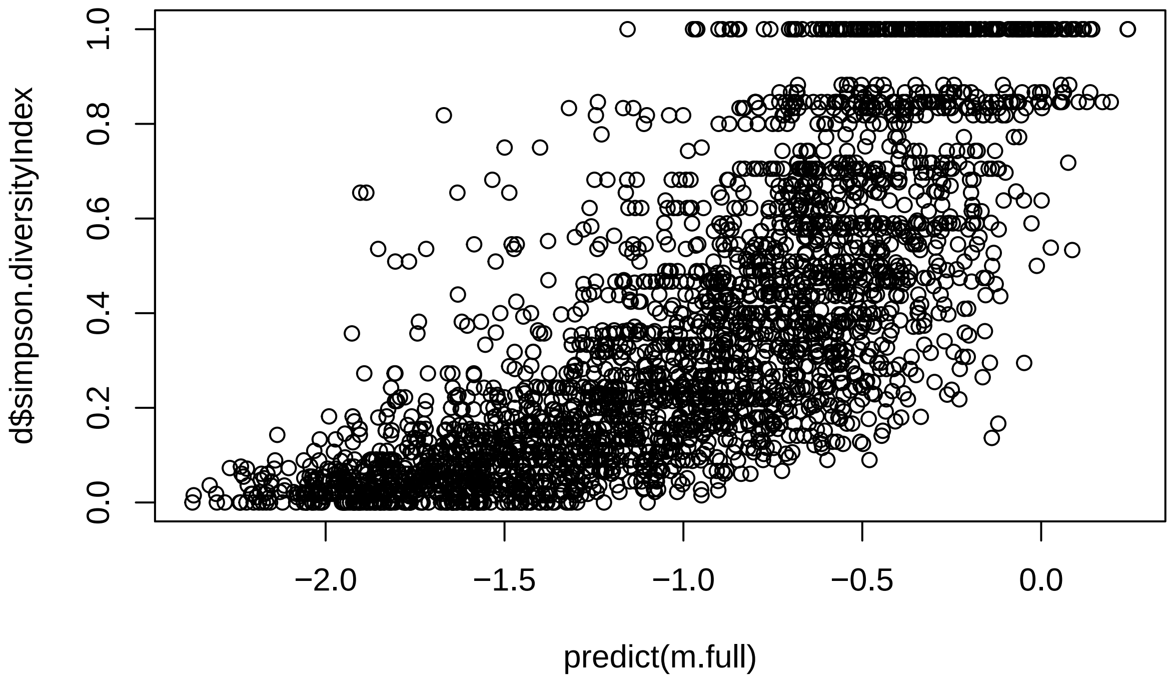


Plotting random effects...


(Intercept)
Lao
Yucatec
Turkish
Cantonese
Malay
Tzeltal
Farsi
Zapotec
English
Dogul Dom
Mian
Siwu
ASL
Semai
Yurakare
BSL
Yeli Dnye
Kilivila
Umpila
Kata Kolok
0.19
0.16
0.14
0.14
0.09
0.07
0.06
0.05
0.05
0.01
−0.00
−0.02
−0.03
−0.04
−0.09
−0.10
−0.13
−0.14
−0.21
−0.21
Group levels
−0.25
0.00
0.25
BLUP


(Intercept)
taste
colour
shape
sound
touch
smell
0.39
0.32
0.05
−0.11
−0.14
−0.50
Group levels
−0.4
0.0
0.4
BLUP

```r
plot(predict(m.full), d$simpson.diversityIndex)
```



Summary

The influence of each of the following random effects was tested:

- Language
- Domain
- Stimulus item (nested within domains)
- Domains within languages (interaction)

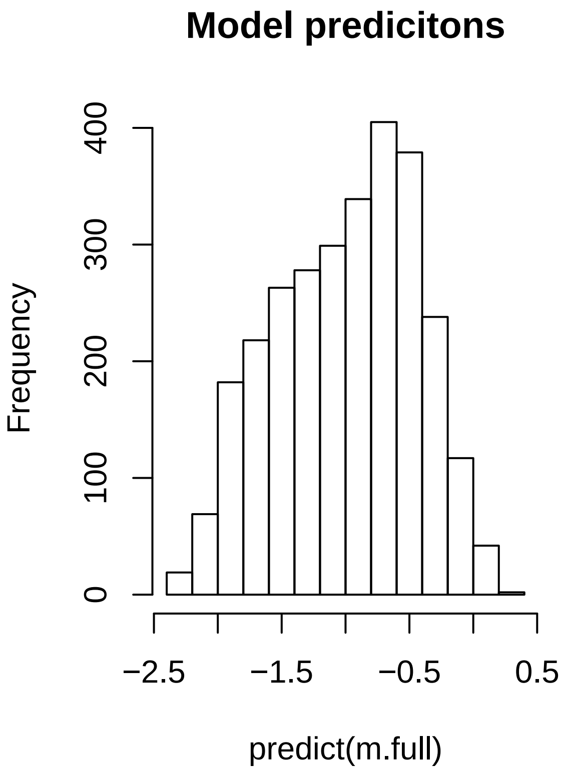
The influence of each was tested by comparing a full model with all random effects to one where the given random effect was removed.

There was a significant random effect for language (log likelihood difference = 2.3 , df = 1 , Chi Squared = 4.55 , p = 0.033). There was a significant random effect for domain (log likelihood difference = 14 , df = 1 , Chi Squared = 27.83 , p = 1.3e-07). There was a significant random effect for stimulus item (within domain) (log likelihood difference = 260 , df = 1 , Chi Squared = 514.81 , p = 5.7e-114). There was a significant random effect for the interaction between language and domain (log likelihood difference = 350 , df = 1 , Chi Squared = 700.03 , p = 3e-154).

Distribution assumpsions

The fit of the model distributions is reasonable:

```
par(mfrow=c(1,2))
hist(predict(m.full), main="Model predicitons")
hist(d$logDiversity, main="Actual data")
```



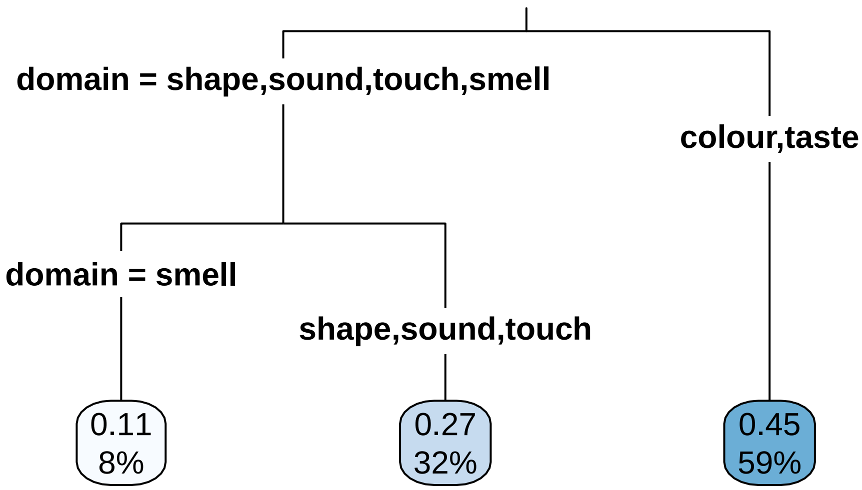
We can try fitting the data with different distributions (Gamma, inverse gaussian):

The gaussian model is the best fit by AIC.

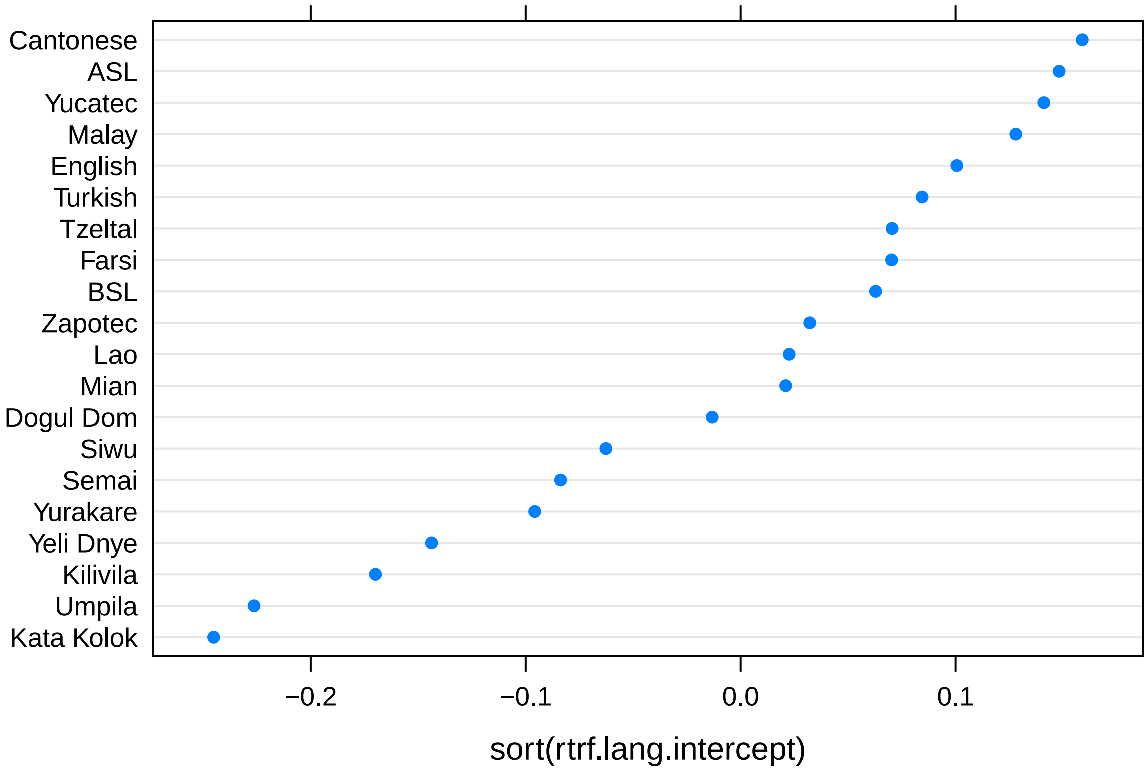
Differences between domains

How do the different domains cluster according to codability? We can use REEMtree which allows random effects for Language and Stimulus:

```
REEMresult<-REEMtree(simpson.diversityIndex~ domain,
                     data=d,
                     random= list(~ 1|Language,~1|Stimulus.code),
                     MaxIterations = 100000)
rpart.plot(tree(REEMresult), type=3)
```




```
rtrf.lang = REEMresult$RandomEffects$Language
rtrf.lang.intercept = rtrf.lang[,1]
names(rtrf.lang.intercept) = rownames(rtrf.lang)
dotplot(sort(rtrf.lang.intercept))
```



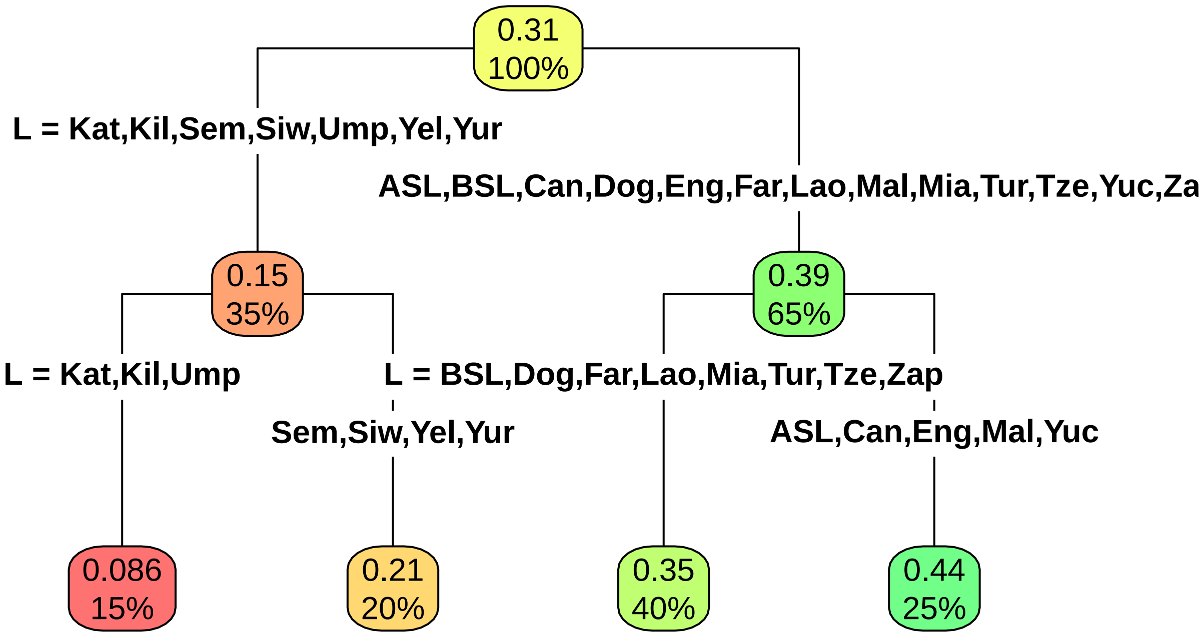
According to the decision tree results, the hierarchy of codability is:

[Colour, Taste] > [Shape, Sound, Touch] > Smell

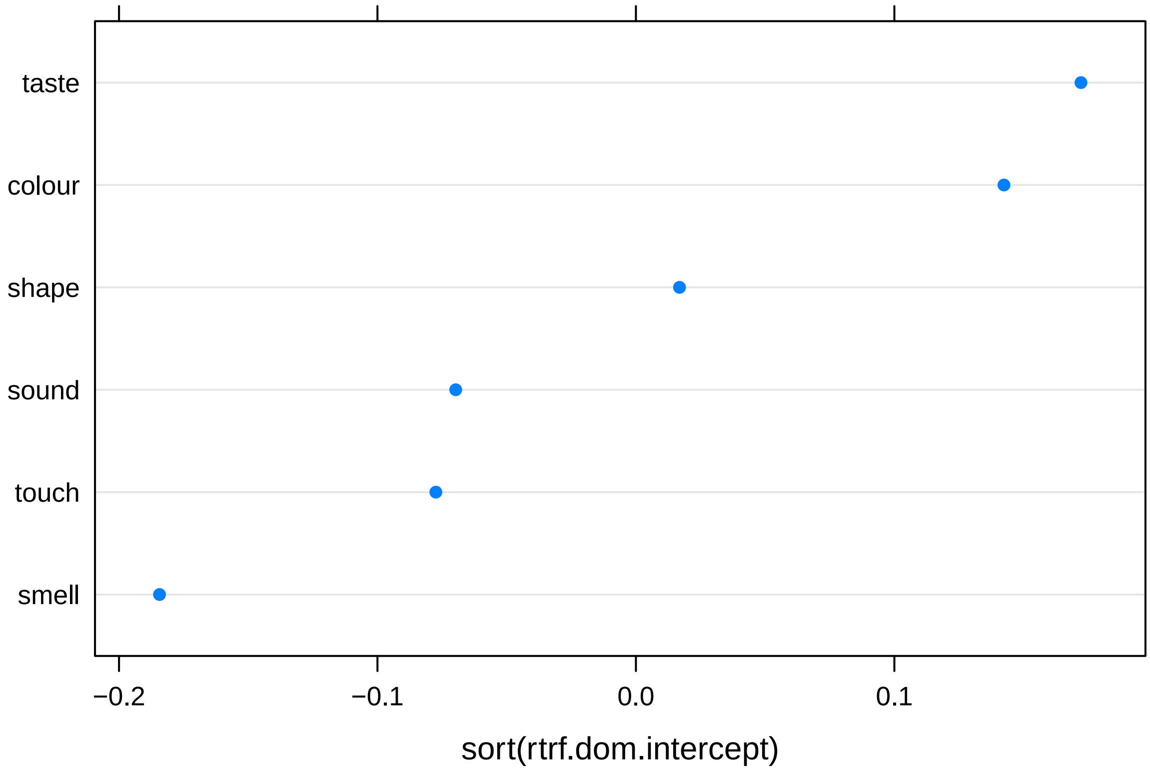
Differences between langauges

We can also ask which languages cluster together:

```r
d$L = substr(d$Language,1,3)
REEMresult.lang<-REEMtree(simpson.diversityIndex~ L,
                     data=d,
                     random= c(~ 1|domain,~1|Stimulus.code),
                     MaxIterations = 100000)
rpart.plot(tree(REEMresult.lang), type=4, extra=100,
           box.palette="RdYlGn")
```

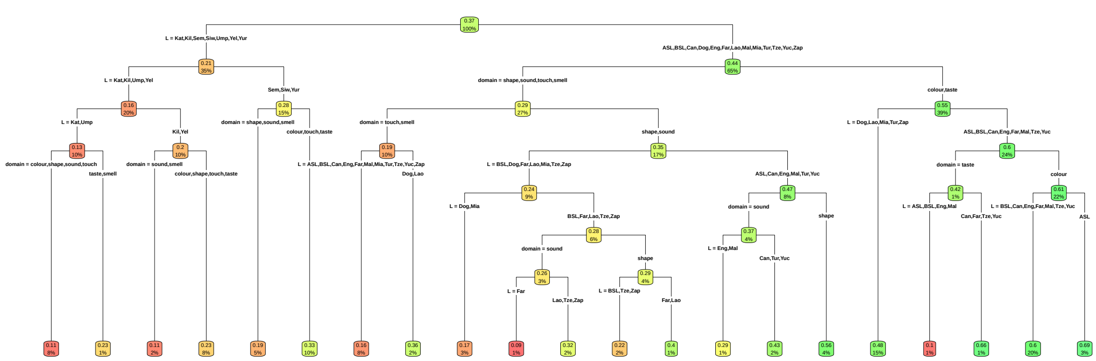



```r
rtrf.dom = REEMresult.lang$RandomEffects$domain
rtrf.dom.intercept = rtrf.dom[,1]
names(rtrf.dom.intercept) = rownames(rtrf.dom)
dotplot(sort(rtrf.dom.intercept))
```



Here's a decision tree splitting the data by domain and language. It is harder to understand the splits here, which is to say that it is not easy to make a generalisation about the differences. The main point is that there are many interactions between langauge and domain, not just one big difference.

```
REEMresult.both<-REEMtree(simpson.diversityIndex~ L+domain,
                         data=d,
                         random= ~ 1|Stimulus.code,
                         MaxIterations = 100000)
rpart.plot(tree(REEMresult.both), type=4, extra=100,
           box.palette="RdYlGn")
```



```
pdf("../results/graphs/DecisionTree_LanguagesAndDomains.pdf",
    width=15, height=10)
rpart.plot(tree(REEMresult.both), type=4, extra=100,
           box.palette="RdYlGn")
dev.off()
```

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

```r
tree(REEMresult.both)$variable.importance
```

```
##        L   domain
## 56.08135 41.60614
```

We were also interested in whether languages differ by modality:

```r
d$L = substr(d$Language,1,3)
d$modality = "Spoken"
d$modality[d$Language %in% c("ASL", "BSL", "Kata Kolok")] = "Signed"
REEMresult.mod<-REEMtree(simpson.diversityIndex~ L + domain + modality,
                     data=d,
                     random= c(~1|Stimulus.code),
                     MaxIterations = 100000)
tree(REEMresult.mod)$variable.importance
```

```
##         L    domain  modality
## 56.952422 42.103148  2.010103
```

Modality is not used in the tree (not shown), and is not used if it is the only variable in available to the tree (not shown). The importance measure for modality is 20 times lower than for language and domain. That is, languages do not cluster by modality.

Description types and codability

Is there better codability for more abstract terms?

```
sae = read.csv("../data/AllData_LoP.csv", stringsAsFactors = F)
sae = sae[!is.na(sae$head),]
sae = sae[!sae$head %in% c("n/a","no description"),]
sae = sae[!is.na(sae$SAE),]
sae = sae[sae$Response==1,]
prop.sae = sae %>% group_by(Language,domain,SAE) %>%
  summarise (n = n()) %>%
  mutate(prop = n / sum(n))
```

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

```
d$Abstract = NA
# Match up each diversity measure with the proportion of
#  abstract terms used
for(lang in unique(d$Language)){
  for(dom in unique(d$domain)){
    propx = prop.sae[prop.sae$Language==lang & prop.sae$domain==dom & prop.sae$SAE=="A",]$prop
    if(length(propx)==0){
      propx = 0
    }
    sel = d$Language==lang & d$domain==dom
    if(sum(sel)!=0){
      d[sel,]$Abstract = propx
    }
  }
}
d$Abstract.scaled = scale(d$Abstract^2)
abs.scale = attr(d$Abstract.scaled,"scaled:scale")
abs.center = attr(d$Abstract.scaled,"scaled:center")
d$Abstract.scaled = as.numeric(d$Abstract.scaled)
```

Plot raw data:

```
rawgx  = ggplot(d, aes(Abstract, simpson.diversityIndex)) +
  geom_point(alpha=0.2) +
  stat_smooth(method = 'gam') +
  xlab("Proportion of abstract terms") +
  ylab("Codability")
pdf("../results/graphs/Codability_by_AbstractUse_Raw.pdf", width=4, height=4)
rawgx
dev.off()
```

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

Raw correlation:

```
cor(d$Abstract,d$simpson.diversityIndex)
```

```
## [1] 0.3416066
```

Compare models with and without the main effect of abstract types.

```r
m0.sae = lmer( logDiversity ~ 1 +
              (1+Abstract.scaled|Language) +
              (1+Abstract.scaled|domain/Stimulus.code) +
              (1|Language:domain),
            data=d)

m1.sae = update(m0.sae, ~.+Abstract.scaled)
anova(m0.sae,m1.sae)
```

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

```
## Data: d
## Models:
## m0.sae: logDiversity ~ 1 + (1 + Abstract.scaled | Language) + (1 + Abstract.scaled |
## m0.sae:     domain/Stimulus.code) + (1 | Language:domain)
## m1.sae: logDiversity ~ (1 + Abstract.scaled | Language) + (1 + Abstract.scaled |
## m1.sae:     domain/Stimulus.code) + (1 | Language:domain) + Abstract.scaled
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0.sae 12 3871.3 3942.8 -1923.7   3847.3
## m1.sae 13 3858.3 3935.7 -1916.1   3832.3 15.009      1   0.000107 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
px = sjp.lmer(m1.sae,'eff','Abstract.scaled', show.ci = T, prnt.plot = F)
px$plot$data$x = sqrt((px$plot$data$x  * abs.scale + abs.center))
px$plot$data$y = exp(px$plot$data$y) -0.1
px$plot$data$upper = exp(px$plot$data$upper) -0.1
px$plot$data$lower = exp(px$plot$data$lower) -0.1
px = px$plot+ xlab("Proportion of abstract terms") +
  ylab("Codability") +
  theme(strip.background = element_blank(),
        strip.text.y = element_blank(),
        plot.title = element_blank())
pdf("../results/graphs/Codability_by_AbstractUse.pdf", width=4, height=4)
px
dev.off()
```

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

```r
save(px,file="../results/graphs/Codability_by_AbstractUse_ggplot.RDat")
```

Permutation test

We can use a permutation test to test pairs of domains against each other. We test whether the difference in mean diversity between each pair of domains is greater than would be expected by chance. For each possible pairing of domains, calculate the difference in means for codability. Then randomly swap observations between the two doamins (permutation) and calculate the mean again. The difference in the two means is an indication of the extent of the difference between domains. Arguably, the decision tree in the seciton above is a better way of doing this, because it takes into account random effects for language and stimulus. However, the permutation test makes fewer assumptions about the shape of the distribution.

```
# The distribution of the variable is not important
#  in permutaiton, so we just use the raw index:
d$diversity = d$simpson.diversityIndex

permuteX = function(d,fact,p){
  pDiff = tapply(d[d[,fact] %in% p,]$diversity,
                    sample(d[d[,fact] %in% p,fact]),
                    mean)
  pDiff = abs(diff(pDiff[!is.na(pDiff)]))
  return(pDiff)
}

compareWithPermutation = function(d,fact, numPerms = 1000){
  pairs = combn(unique(as.character(d[,fact])),2)
  set.seed(2387)
  permTests = apply(pairs,2, function(p){
    trueDiff = tapply(d[d[,fact] %in% p,]$diversity,
                      d[d[,fact] %in% p,fact],
                      mean)
    trueDiff = abs(diff(trueDiff[!is.na(trueDiff)]))
    permDiff = replicate(numPerms,permuteX(d,fact,p))
    z = (trueDiff -mean(permDiff))/sd(permDiff)
    p = sum(permDiff >= trueDiff)/length(permDiff)
    if(p==0){
      p = 1/length(permDiff)
    }
    return(c(z,p))
  })

  res = data.frame(
    pair = apply(pairs,2,paste,collapse=','),
    perm.z = permTests[1,],
    perm.p = permTests[2,],
    perm.p.adjusted =
      p.adjust(permTests[2,],'bonferroni'))

  res
}

compareWithPermutation(d,'domain')
```

```
##              pair    perm.z perm.p perm.p.adjusted
## 1  colour,shape 10.229982  0.001           0.015
## 2  colour,smell 23.346891  0.001           0.015
## 3  colour,taste  1.527214  0.081           1.000
```

```
## 4  colour,touch 14.925907  0.001           0.015
## 5  colour,sound 16.716660  0.001           0.015
## 6   shape,smell 15.457834  0.001           0.015
## 7   shape,taste  7.292646  0.001           0.015
## 8   shape,touch  6.009034  0.001           0.015
## 9   shape,sound  5.685990  0.001           0.015
## 10  smell,taste 17.711571  0.001           0.015
## 11  smell,touch  9.603543  0.001           0.015
## 12  smell,sound 12.351339  0.001           0.015
## 13  taste,touch 11.245055  0.001           0.015
## 14  taste,sound 12.567821  0.001           0.015
## 15  touch,sound  0.463503  0.288           1.000
```

The mean codability for all paris of domains are different, except for:

- Colour and Taste
- Touch and Sound

So, the hierarchy is:

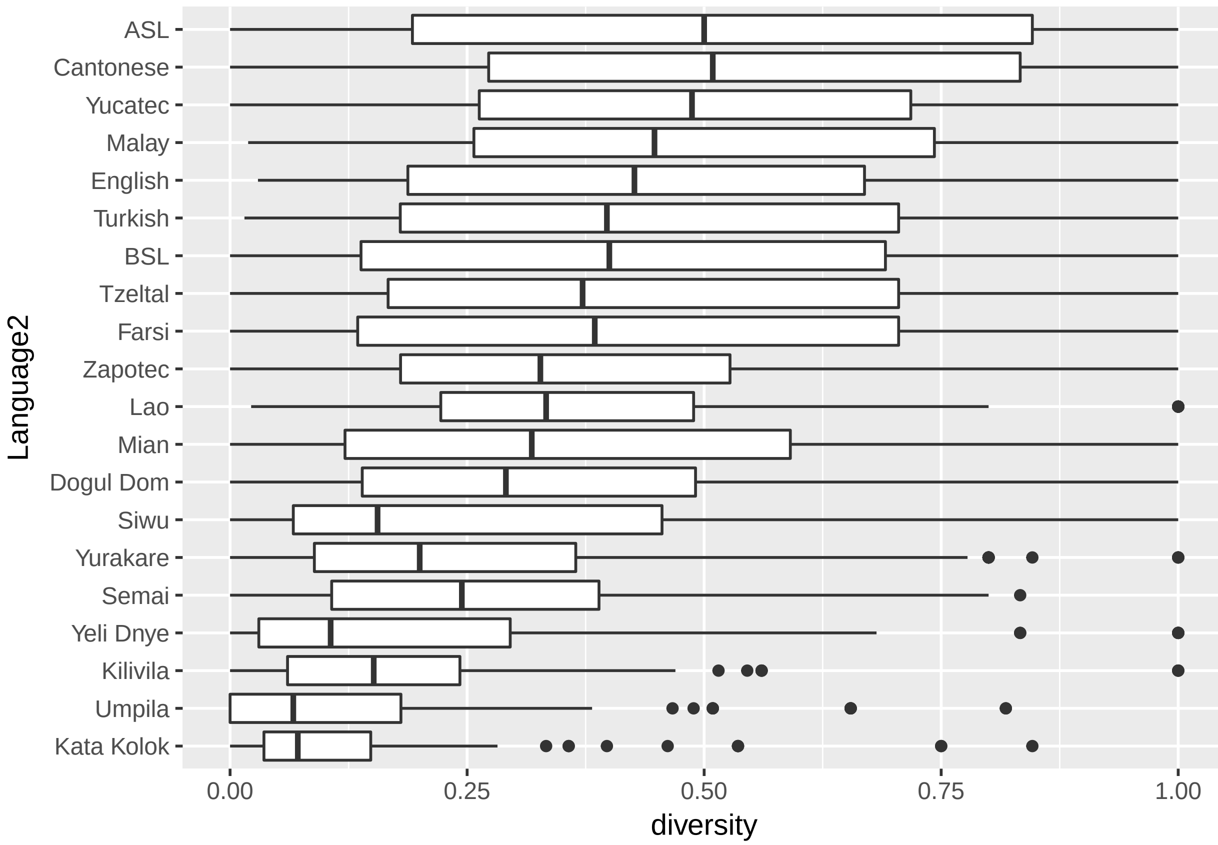
[Colour, Taste] > Shape > [Sound, Touch] > Smell

Which matches the decision tree hierarchy very well.

Permutation between languages

Test whether the mean diversity differs between each pair of languages. This is just to double-check that the results from the mixed effects model above are not artefacts of the shape of the codability distribution. A large number of permutations is needed so that the p-value remains significant when controlling for multiple comparisons.

```
d$Language2 = factor(d$Language,
                     levels =
                       names(sort(
                         tapply(d$simpson.diversityIndex,
                              d$Language,
                              mean))))

ggplot(d, aes(y=diversity,x=Language2)) +
  geom_boxplot() + coord_flip()
```




```r
# (need 20000 permutations for Bonferroni correction)
langPerm = compareWithPermutation(d,'Language', numPerms = 20000)
```

List of significant differences:

```r
langPerm[langPerm$perm.p.adjusted<0.01,]
```

```
##                    pair    perm.z perm.p perm.p.adjusted
## 3         ASL,Dogul Dom  6.386793  5e-05          0.0095
## 6        ASL,Kata Kolok 14.415690  5e-05          0.0095
## 7          ASL,Kilivila 13.020372  5e-05          0.0095
## 8               ASL,Lao  5.119374  5e-05          0.0095
## 10             ASL,Mian  4.755508  5e-05          0.0095
## 11            ASL,Semai 10.017882  5e-05          0.0095
## 12             ASL,Siwu  7.888725  5e-05          0.0095
## 15           ASL,Umpila 14.872409  5e-05          0.0095
## 16        ASL,Yeli Dnye 11.306146  5e-05          0.0095
## 18         ASL,Yurakare  8.723735  5e-05          0.0095
## 24       BSL,Kata Kolok 12.718123  5e-05          0.0095
## 25         BSL,Kilivila 10.739672  5e-05          0.0095
## 29            BSL,Semai  6.978705  5e-05          0.0095
## 30             BSL,Siwu  4.778091  5e-05          0.0095
## 33           BSL,Umpila 12.883928  5e-05          0.0095
## 34        BSL,Yeli Dnye  8.647660  5e-05          0.0095
## 36         BSL,Yurakare  5.794757  5e-05          0.0095
## 38  Cantonese,Dogul Dom  7.019165  5e-05          0.0095
## 41 Cantonese,Kata Kolok 15.741879  5e-05          0.0095
## 42   Cantonese,Kilivila 14.306751  5e-05          0.0095
```

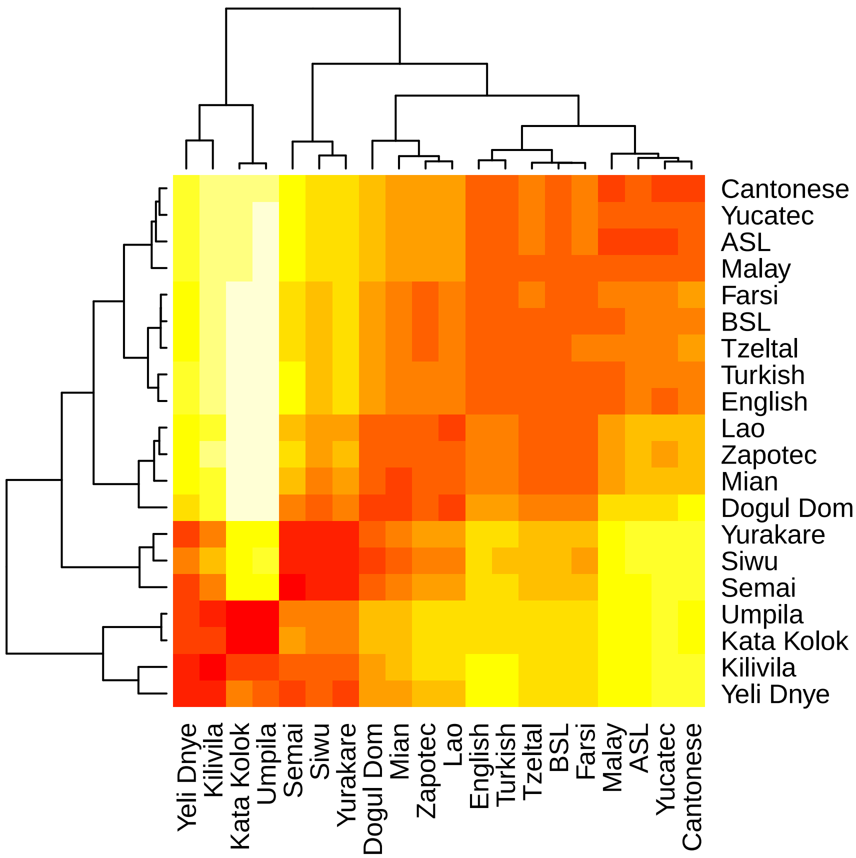
```
## 43          Cantonese,Lao  5.606992  5e-05            0.0095
## 45         Cantonese,Mian  5.207652  5e-05            0.0095
## 46        Cantonese,Semai 10.881228  5e-05            0.0095
## 47         Cantonese,Siwu  8.545814  5e-05            0.0095
## 50       Cantonese,Umpila 16.091613  5e-05            0.0095
## 51    Cantonese,Yeli Dnye 12.195978  5e-05            0.0095
## 53     Cantonese,Yurakare  9.526955  5e-05            0.0095
## 54      Cantonese,Zapotec  4.934960  5e-05            0.0095
## 57  Dogul Dom,Kata Kolok 11.168765  5e-05            0.0095
## 58    Dogul Dom,Kilivila  8.305659  5e-05            0.0095
## 60       Dogul Dom,Malay  5.805998  5e-05            0.0095
## 66      Dogul Dom,Umpila 11.144685  5e-05            0.0095
## 67   Dogul Dom,Yeli Dnye  5.845490  5e-05            0.0095
## 68     Dogul Dom,Yucatec  6.578805  5e-05            0.0095
## 72    English,Kata Kolok 14.261218  5e-05            0.0095
## 73      English,Kilivila 12.595069  5e-05            0.0095
## 77         English,Semai  8.608346  5e-05            0.0095
## 78          English,Siwu  6.186397  5e-05            0.0095
## 81        English,Umpila 14.512258  5e-05            0.0095
## 82     English,Yeli Dnye 10.281170  5e-05            0.0095
## 84      English,Yurakare  7.224110  5e-05            0.0095
## 86      Farsi,Kata Kolok 12.842217  5e-05            0.0095
## 87        Farsi,Kilivila 10.844462  5e-05            0.0095
## 91           Farsi,Semai  6.809733  5e-05            0.0095
## 95          Farsi,Umpila 13.043482  5e-05            0.0095
## 96       Farsi,Yeli Dnye  8.778286  5e-05            0.0095
## 98        Farsi,Yurakare  5.748382  5e-05            0.0095
## 101       Kata Kolok,Lao 13.387166  5e-05            0.0095
## 102     Kata Kolok,Malay 15.130729  5e-05            0.0095
## 103      Kata Kolok,Mian 11.453272  5e-05            0.0095
## 104     Kata Kolok,Semai  9.349440  5e-05            0.0095
## 105      Kata Kolok,Siwu  7.786608  5e-05            0.0095
## 106   Kata Kolok,Turkish 14.183490  5e-05            0.0095
## 107   Kata Kolok,Tzeltal 12.948404  5e-05            0.0095
## 110   Kata Kolok,Yucatec 16.100007  5e-05            0.0095
## 111  Kata Kolok,Yurakare  7.727963  5e-05            0.0095
## 112   Kata Kolok,Zapotec 12.986692  5e-05            0.0095
## 113         Kilivila,Lao 10.455641  5e-05            0.0095
## 114       Kilivila,Malay 13.540861  5e-05            0.0095
## 115        Kilivila,Mian  8.856748  5e-05            0.0095
## 116       Kilivila,Semai  4.971180  5e-05            0.0095
## 118     Kilivila,Turkish 12.264060  5e-05            0.0095
## 119     Kilivila,Tzeltal 11.005224  5e-05            0.0095
## 122     Kilivila,Yucatec 14.469779  5e-05            0.0095
## 124     Kilivila,Zapotec 10.455408  5e-05            0.0095
## 127            Lao,Semai  5.647939  5e-05            0.0095
## 131           Lao,Umpila 13.234881  5e-05            0.0095
## 132        Lao,Yeli Dnye  7.753110  5e-05            0.0095
## 133          Lao,Yucatec  5.017567  5e-05            0.0095
## 137          Malay,Semai  9.956038  5e-05            0.0095
## 138           Malay,Siwu  7.352764  5e-05            0.0095
## 141         Malay,Umpila 15.483867  5e-05            0.0095
## 142      Malay,Yeli Dnye 11.330134  5e-05            0.0095
## 144       Malay,Yurakare  8.461861  5e-05            0.0095
```

```
## 150          Mian,Umpila 11.574841  5e-05         0.0095
## 151       Mian,Yeli Dnye  6.770497  5e-05         0.0095
## 156        Semai,Turkish  8.180076  5e-05         0.0095
## 157        Semai,Tzeltal  7.038307  5e-05         0.0095
## 158         Semai,Umpila  8.942022  5e-05         0.0095
## 160        Semai,Yucatec 10.851852  5e-05         0.0095
## 162        Semai,Zapotec  5.921801  5e-05         0.0095
## 163         Siwu,Turkish  5.617457  5e-05         0.0095
## 165          Siwu,Umpila  7.902980  5e-05         0.0095
## 167         Siwu,Yucatec  8.138316  5e-05         0.0095
## 171       Turkish,Umpila 14.392934  5e-05         0.0095
## 172    Turkish,Yeli Dnye  9.926558  5e-05         0.0095
## 174     Turkish,Yurakare  6.823549  5e-05         0.0095
## 176       Tzeltal,Umpila 13.248762  5e-05         0.0095
## 177    Tzeltal,Yeli Dnye  8.751177  5e-05         0.0095
## 179     Tzeltal,Yurakare  5.780601  5e-05         0.0095
## 182       Umpila,Yucatec 16.471421  5e-05         0.0095
## 183      Umpila,Yurakare  7.599645  5e-05         0.0095
## 184       Umpila,Zapotec 13.166837  5e-05         0.0095
## 185    Yeli Dnye,Yucatec 12.207134  5e-05         0.0095
## 187    Yeli Dnye,Zapotec  8.009367  5e-05         0.0095
## 188     Yucatec,Yurakare  9.268163  5e-05         0.0095
```

Languages really are different, but some languages are closer than others. This heatmap gives an idea of which languages are similar to which.

```
d$Language2 = factor(d$Language, levels =
              names(sort(tapply(
                d$diversity,
                d$Language,
                mean))))

langPerm$l1 = sapply(as.character(langPerm$pair), function(X){strsplit(X,',')[[1]][1]})
langPerm$l2 = sapply(as.character(langPerm$pair), function(X){strsplit(X,',')[[1]][2]})

lxs = unique(d$Language)
langPerm2 = langPerm
langPerm2 = rbind(
  langPerm,
  data.frame(
    pair = paste(lxs,lxs,sep=','),
    perm.z = 0,
    perm.p = 1,
    perm.p.adjusted = 1,
    l1 = lxs,
    l2 = lxs
  )
)

langPerm2$perm.z = abs(langPerm2$perm.z)

langPermDist = acast(langPerm2, l1~l2, value.var="perm.z")

langPermDist[lower.tri(langPermDist)] = t(langPermDist)[lower.tri(langPermDist)]
```

```
heatmap(langPermDist)
```



Stimulus set size

A check that the size of the stimulus set does not predict the codability:

```
numStimuli = tapply(d$Stimulus.code,d$domain, function(X){length(unique(X))})

d$numStimuli = numStimuli[d$domain]

m.full = lmer( logDiversity ~ 1 +
                 (1|Language) +
                 (1|domain/Stimulus.code) +
                 (1|Language:domain),
               data=d)

m.ns = lmer( logDiversity ~ 1 +
               numStimuli +
                 (1|Language) +
                 (1|domain/Stimulus.code) +
                 (1|Language:domain),
               data=d)

anova(m.full, m.ns)
```

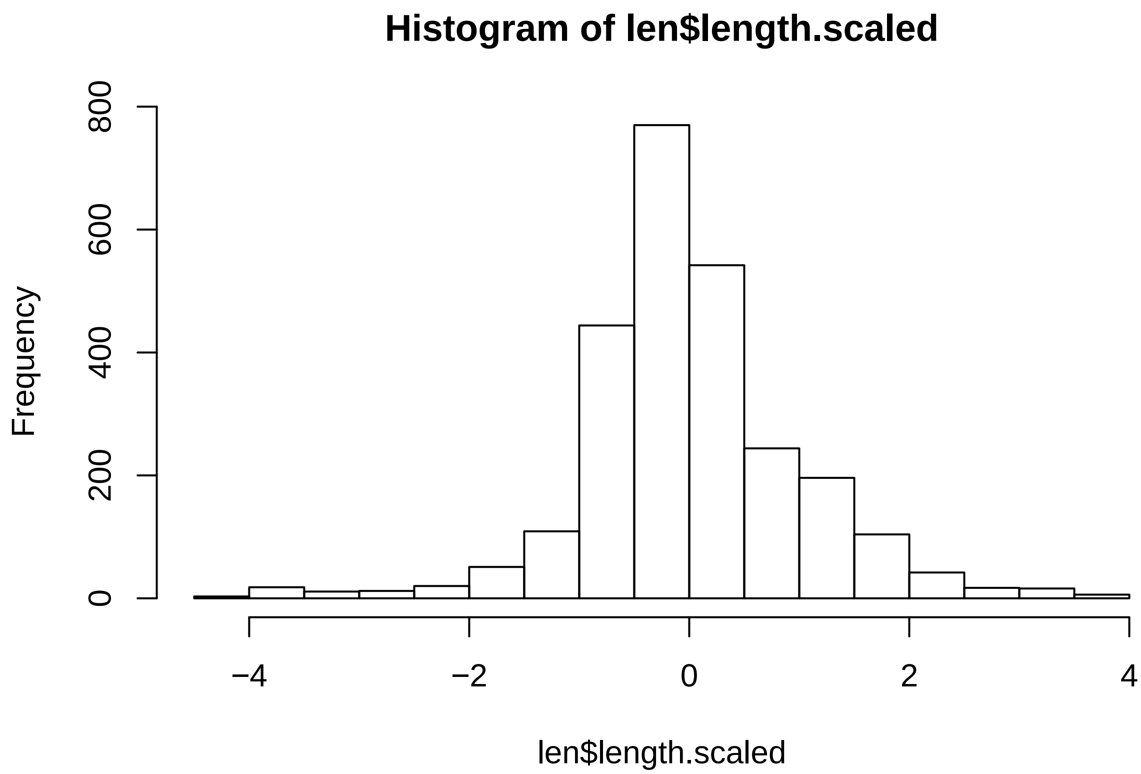
```
## refitting model(s) with ML (instead of REML)

## Data: d
## Models:
## m.full: logDiversity ~ 1 + (1 | Language) + (1 | domain/Stimulus.code) +
## m.full:     (1 | Language:domain)
## m.ns: logDiversity ~ 1 + numStimuli + (1 | Language) + (1 | domain/Stimulus.code) +
## m.ns:     (1 | Language:domain)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m.full  6 3939.3 3975.0 -1963.7   3927.3
## m.ns    7 3940.2 3981.9 -1963.1   3926.2 1.1438      1     0.2848
```

No significant prediction. This is easy to see: Taste and Colour have roughly equal mean codability, but colour has the largest number of stimuli (80) and taste has the least (5).

Description lengths

Load the data. The column `mean` is the mean length, and `mean.log` is the mean of the log of the lengths.

```
len = read.csv("../data/DiversityIndices_ND_withLengths.csv",
               stringsAsFactors = F)
len = len[complete.cases(len),]

# Remove 2 outliers that had very few responses:
len = len[len$mean>=1,]

len$logDiversity = log(len$simpson.diversityIndex+0.1)
len$logDiversity.scaled = scale(len$logDiversity)
len$length.scaled = scale(len$mean.log)
len$Stimulus.code = factor(len$Stimulus.code)
len$Language = factor(len$Language)
len$domain = factor(len$domain)

hist(len$length.scaled)
```



Raw correlation:

```
cor(len$mean.log, len$simpson.diversityIndex)
```

```
## [1] -0.1942662
```

Linear model with random intercepts and slopes by language, domain and the interaction between language and domain:

```
mLenl0 = lmer( logDiversity.scaled ~ 1 +
                 (1+length.scaled|Language) +
                 (1+length.scaled|domain/Stimulus.code) +
                 (1+length.scaled|Language:domain),
               data=len)
mLenl1 = update(mLenl0, ~.+length.scaled)
mLenl2 = update(mLenl1, ~.+I(length.scaled^2))
anova(mLenl0,mLenl1,mLenl2)
```

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

```
## Data: len
## Models:
## mLenl0: logDiversity.scaled ~ 1 + (1 + length.scaled | Language) + (1 +
## mLenl0:     length.scaled | domain/Stimulus.code) + (1 + length.scaled |
## mLenl0:     Language:domain)
## mLenl1: logDiversity.scaled ~ (1 + length.scaled | Language) + (1 + length.scaled |
## mLenl1:     domain/Stimulus.code) + (1 + length.scaled | Language:domain) +
## mLenl1:     length.scaled
## mLenl2: logDiversity.scaled ~ (1 + length.scaled | Language) + (1 + length.scaled |
## mLenl2:     domain/Stimulus.code) + (1 + length.scaled | Language:domain) +
## mLenl2:     length.scaled + I(length.scaled^2)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
```

```
## mLenl0 14 5049.5 5131.6 -2510.8   5021.5
## mLenl1 15 5045.9 5133.9 -2508.0   5015.9 5.5888      1    0.01808 *
## mLenl2 16 5046.7 5140.5 -2507.3   5014.7 1.2463      1    0.26426
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is a significant effect, but no quadratic effect. Since extremely short responses are unlikely to be informative, we wondered if there were higher-level non-linear effects. We ran the same model as above, but as a generalised additive model (GAM):

```
library(mgcv)
library(itsadug)
library(lmtest)
mLen = bam(logDiversity.scaled ~
      s(length.scaled) +
      s(Language,bs='re')+
      s(domain,bs='re')+
      s(Stimulus.code,bs='re')+
      s(Language,domain,bs='re'), # interaction
    data = len)
```

Test for need for random slopes:

```
mLen2 = update(mLen, ~.+s(Language,length.scaled,bs='re'))
lrtest(mLen,mLen2)
```

```
## Likelihood ratio test
##
## Model 1: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re")
## Model 2: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re")
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 214.93 -2192.5
## 2 232.94 -2103.9 18.008 177.13  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mLen3 = update(mLen2, ~.+s(domain,length.scaled,bs='re'))
lrtest(mLen2,mLen3)
```

```
## Likelihood ratio test
##
## Model 1: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re")
## Model 2: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re")
##      #Df  LogLik     Df  Chisq Pr(>Chisq)
## 1 232.94 -2103.9
## 2 235.20 -2097.8 2.2673 12.354   0.002077 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mLen4 = update(mLen3, ~.+s(Stimulus.code,length.scaled,bs='re'))
lrtest(mLen3,mLen4)
```

```
## Likelihood ratio test
##
## Model 1: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re")
## Model 2: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re") + s(Stimulus.code, length.scaled,
##     bs = "re")
##     #Df  LogLik         Df Chisq Pr(>Chisq)
## 1 235.2 -2097.8
## 2 235.2 -2097.8 0.00010188 2e-04  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mLen5 = update(mLen4, ~.+s(Language,domain,length.scaled,bs='re'))
lrtest(mLen4,mLen5)
```

```
## Likelihood ratio test
##
## Model 1: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re") + s(Stimulus.code, length.scaled,
##     bs = "re")
## Model 2: logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re") + s(Stimulus.code, length.scaled,
##     bs = "re") + s(Language, domain, length.scaled, bs = "re")
##      #Df  LogLik    Df  Chisq Pr(>Chisq)
## 1 235.20 -2097.8
## 2 261.83 -2029.0 26.63 137.43  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All random slopes improve the model. Final model:

```r
summary(mLen5)
```

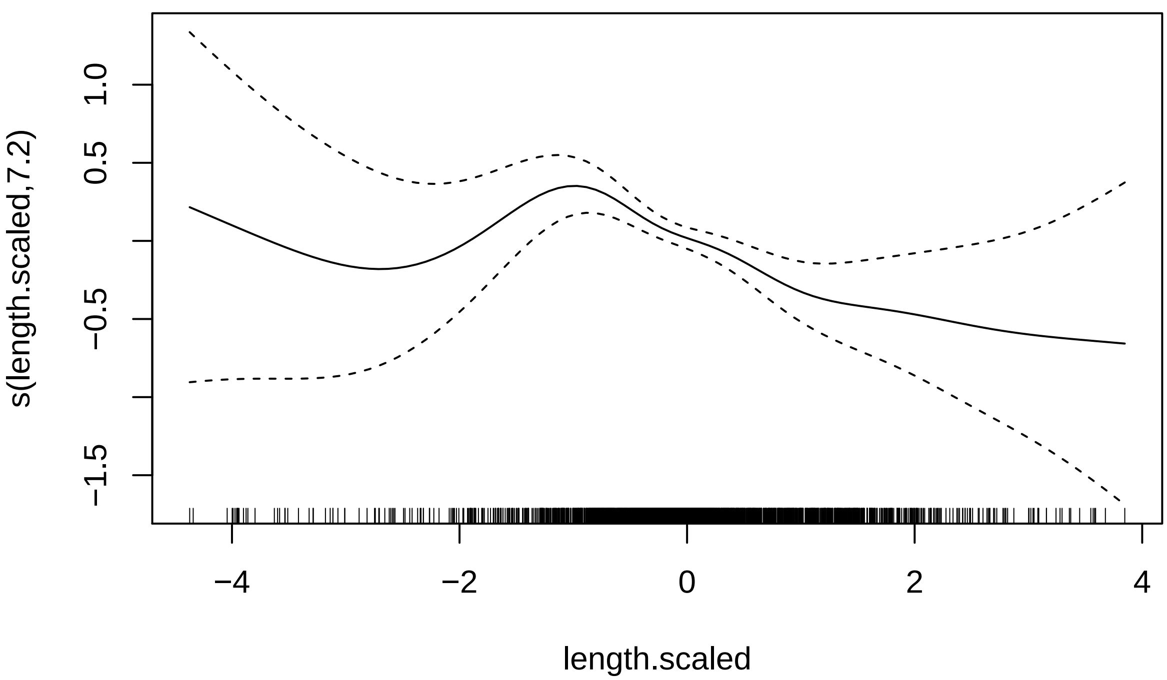
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## logDiversity.scaled ~ s(length.scaled) + s(Language, bs = "re") +
##     s(domain, bs = "re") + s(Stimulus.code, bs = "re") + s(Language,
##     domain, bs = "re") + s(Language, length.scaled, bs = "re") +
##     s(domain, length.scaled, bs = "re") + s(Stimulus.code, length.scaled,
##     bs = "re") + s(Language, domain, length.scaled, bs = "re")
##
```

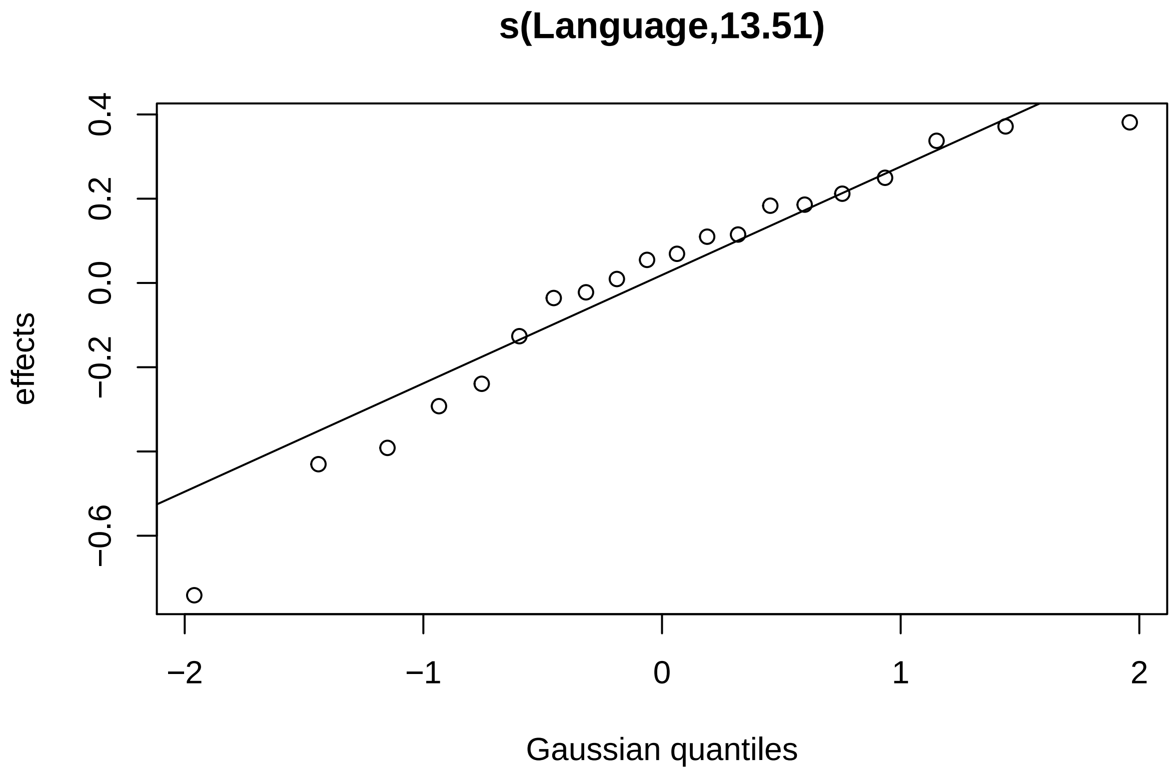
```
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1470     0.2136  -0.688    0.491
##
## Approximate significance of smooth terms:
##                                        edf Ref.df        F  p-value
## s(length.scaled)                 7.202e+00   8.23    5.177 1.29e-06 ***
## s(Language)                      1.351e+01  19.00 1565.987 3.54e-07 ***
## s(domain)                        4.489e+00   5.00 4818.072 2.96e-07 ***
## s(Stimulus.code)                 1.063e+02 131.00    8.931 6.37e-09 ***
## s(Language,domain)               6.323e+01 113.00  279.468 3.16e-13 ***
## s(Language,length.scaled)        1.081e+01  19.00  702.913  0.00188 **
## s(domain,length.scaled)          1.507e+00   5.00  116.030  0.29047
## s(Stimulus.code,length.scaled)   1.998e-05 131.00    0.000  0.99332
## s(Language,domain,length.scaled) 5.003e+01 113.00  115.193  0.00236 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.691   Deviance explained = 72.2%
## fREML =   2504  Scale est. = 0.30859   n = 2605
```

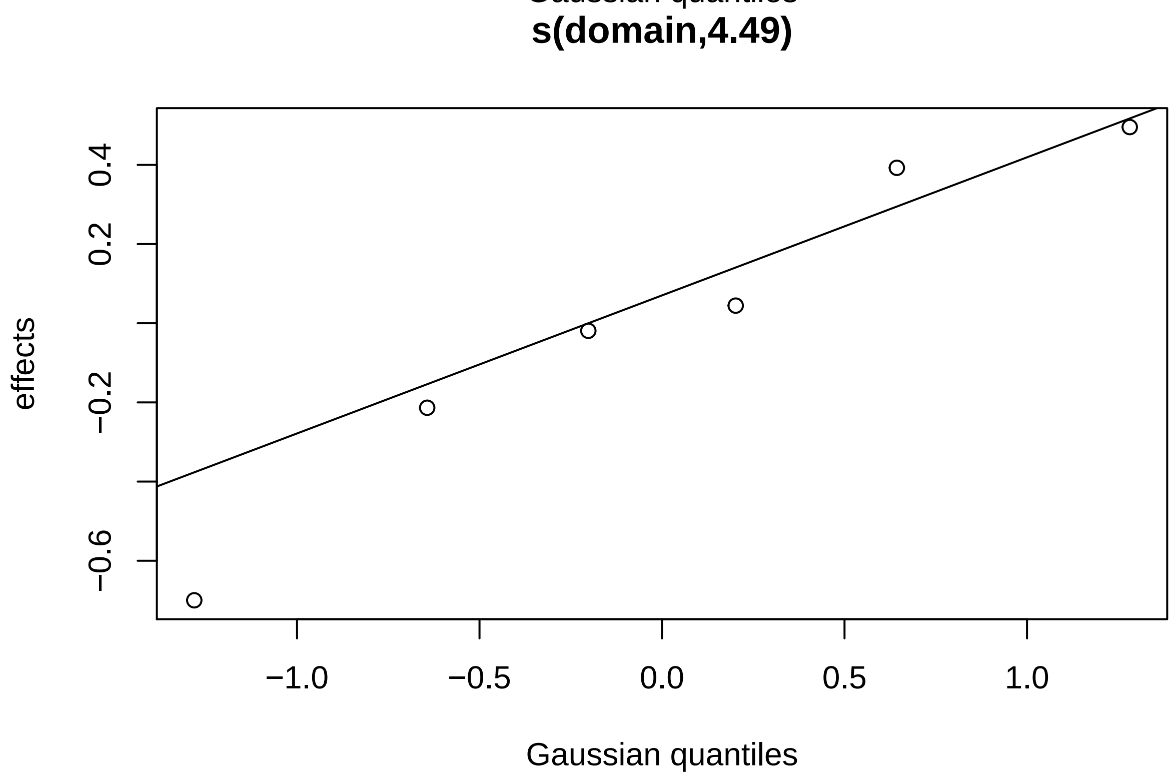
Note that there are significant random slopes for length by Language x domain and by Language, indicating that the stength of the length effect differes across cultures.

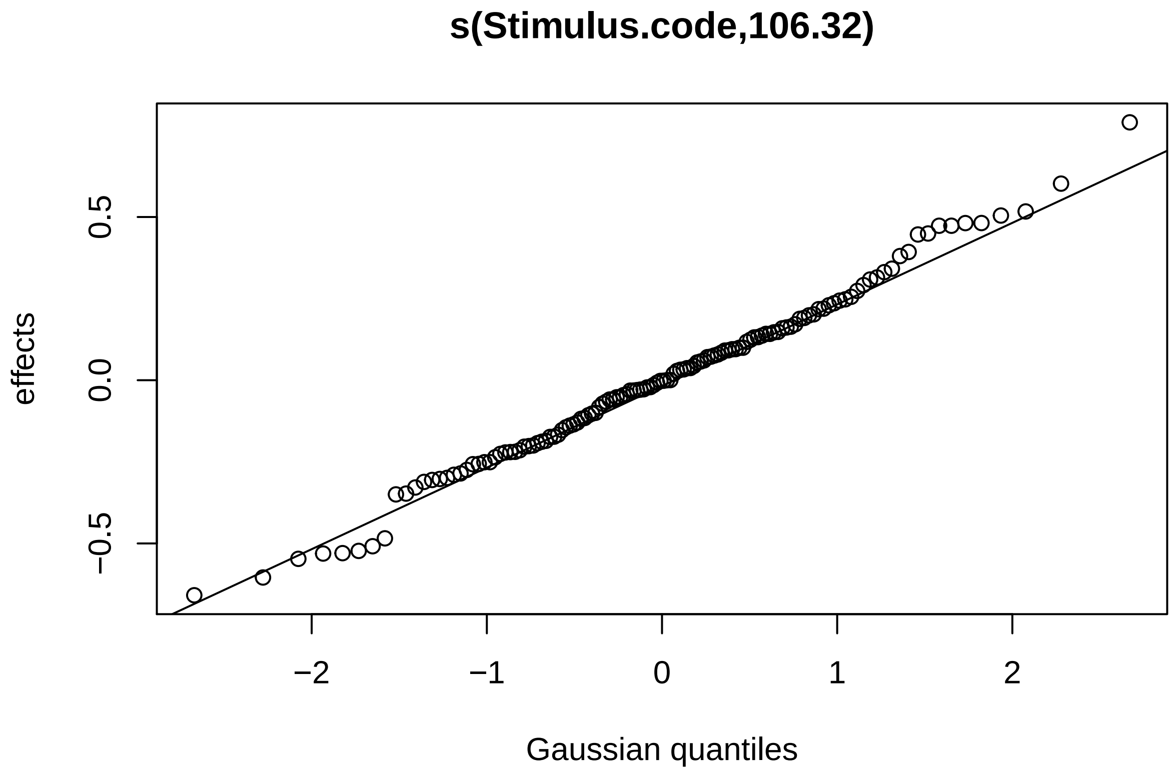
We can plot the model smooths.

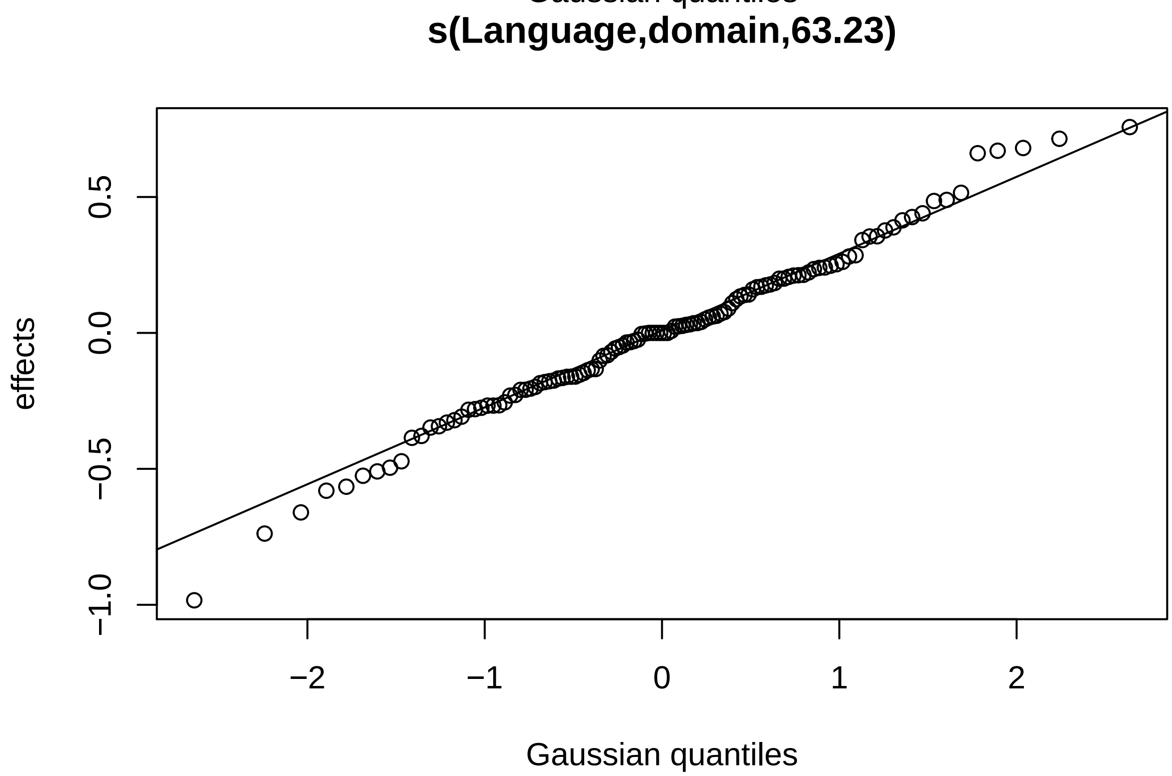
```
plot(mLen5)
```

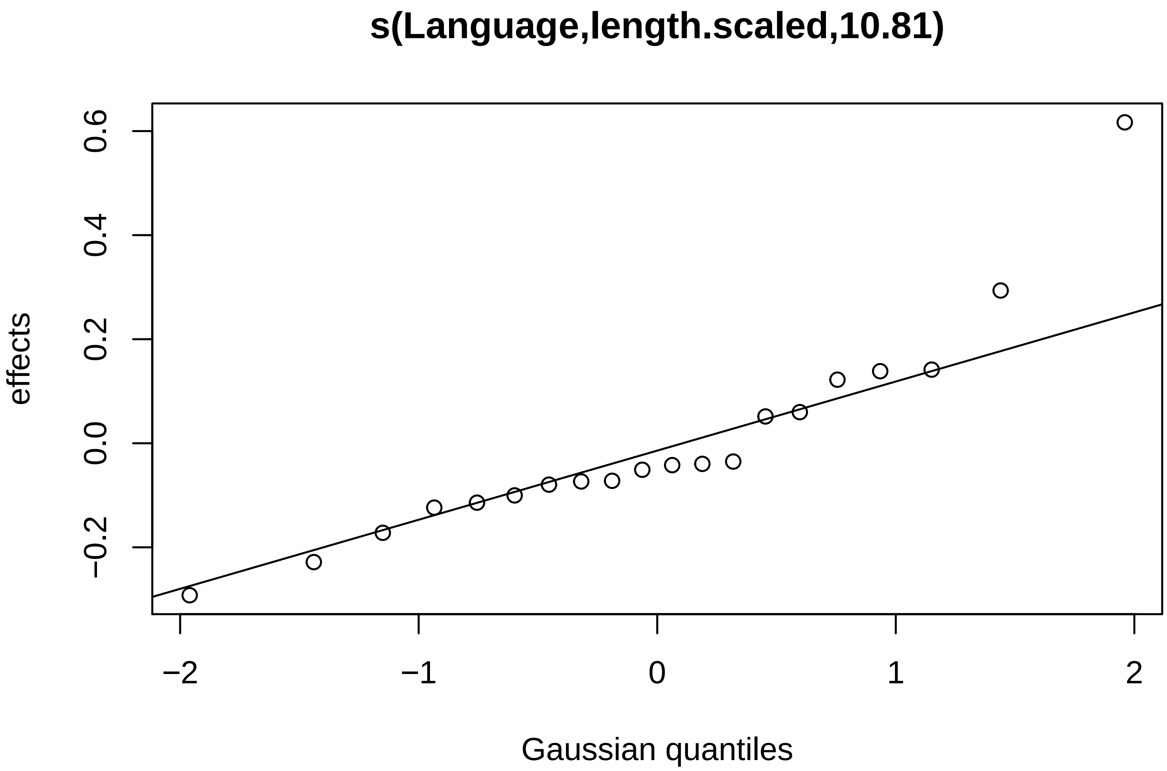


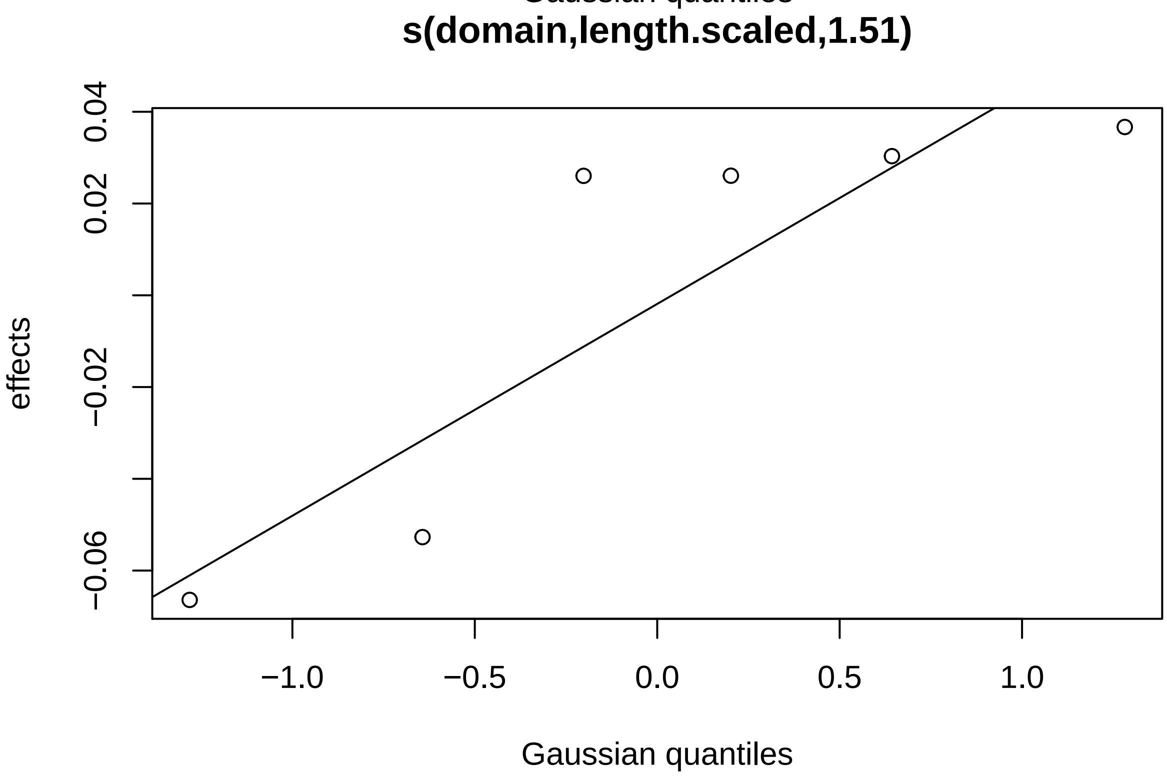

s(Language,13.51)
effects
0.4
0.2
0.0
−0.2
−0.6
−2
−1
0
1
2
Gaussian quantiles

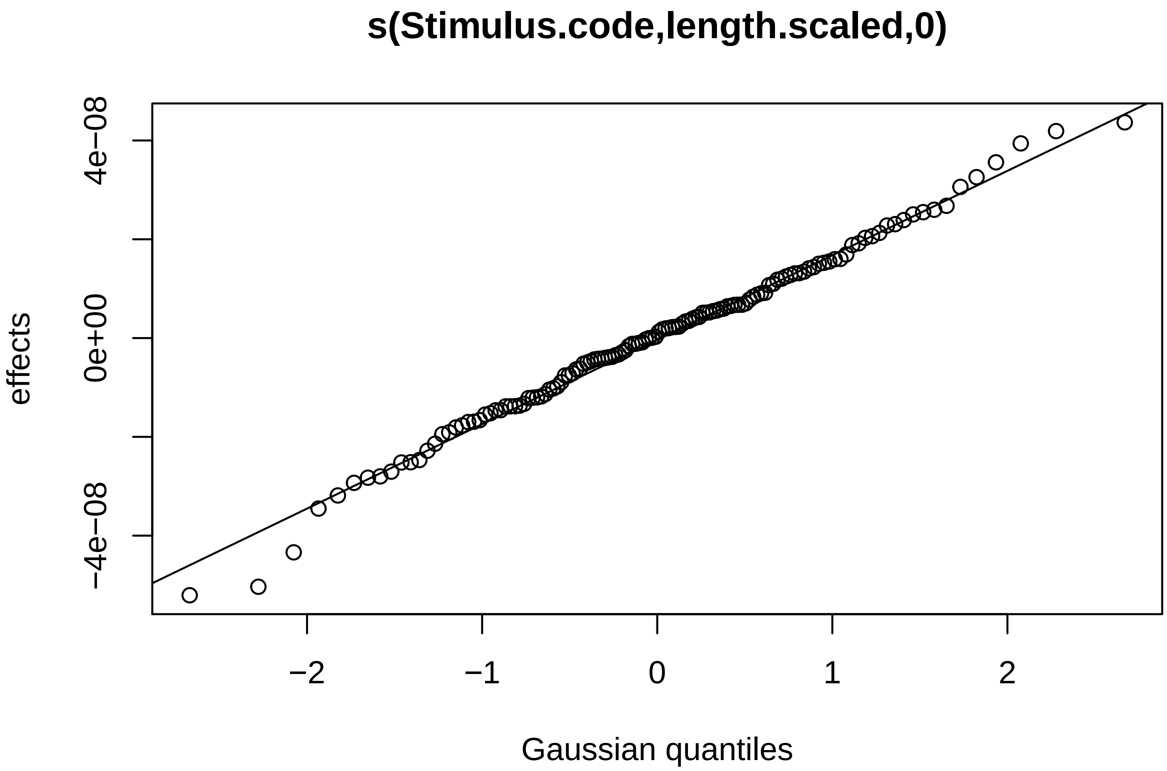

s(domain,4.49)
effects
0.4
0.2
−0.2
−0.6
−1.0
−0.5
0.0
0.5
1.0
Gaussian quantiles

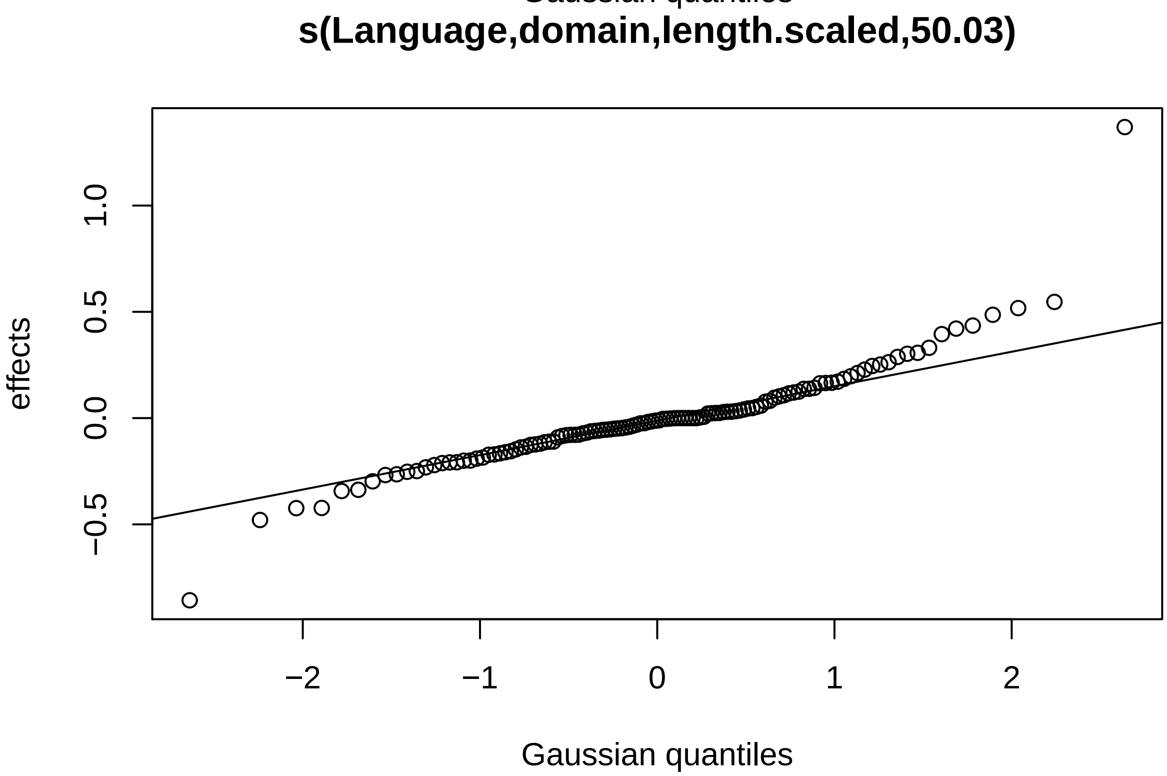

s(Stimulus.code,106.32)
effects
0.5
0.0
−0.5
−2
−1
0
1
2
Gaussian quantiles


s(Language,domain,63.23)
effects
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2
Gaussian quantiles


s(Language,length.scaled,10.81)
effects
0.6
0.4
0.2
0.0
−0.2
−2
−1
0
1
2
Gaussian quantiles


s(domain,length.scaled,1.51)
effects
0.04
0.02
−0.02
−0.06
−1.0
−0.5
0.0
0.5
1.0
Gaussian quantiles





Look at the variation between domains (random slopes). These show the difference in how sensitive domains are to length. For example, the effect of length is less pronounced for smell and more pronounced for touch (though this differs between languages).

```
mDom = mLen5$coefficients
mDom = mDom[grepl("s\\(domain,length.scaled\\)",names(mDom))]
```

```
names(mDom) = as.character(levels(len$domain))
t(t(sort(mDom)))
```

```
##                [,1]
## touch  -0.06639260
## taste  -0.05270594
## colour  0.02603199
## sound   0.02606074
## shape   0.03032637
## smell   0.03667944
```

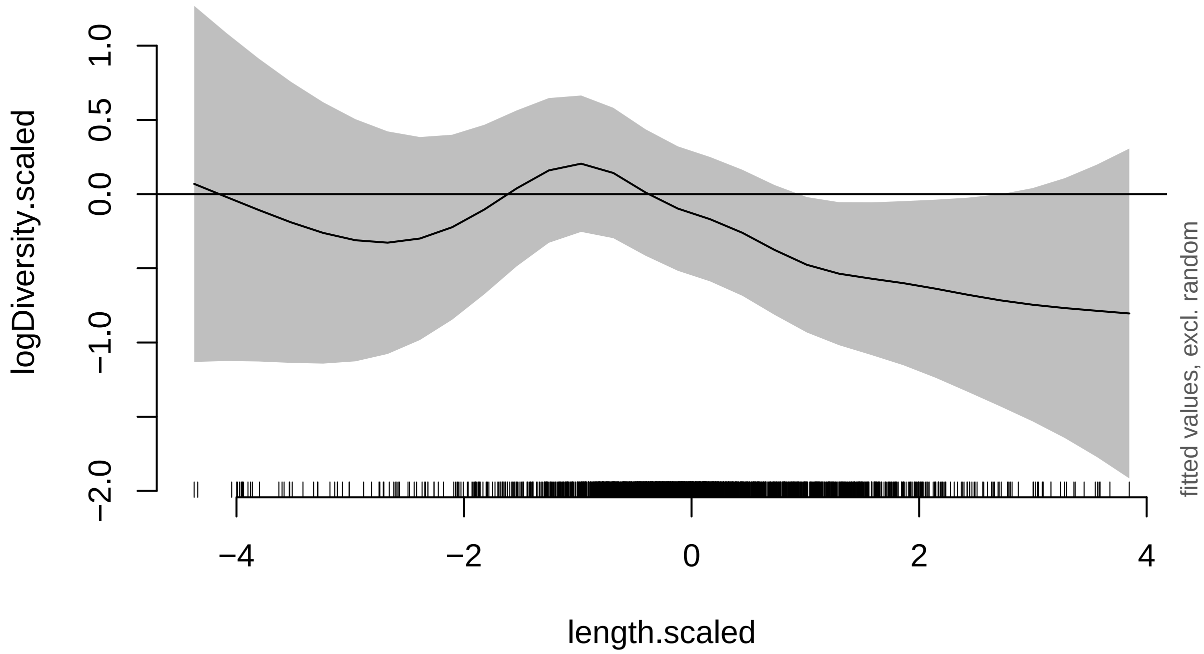
And between languages:

```
mLang = mLen5$coefficients
mLang = mLang[grepl("s\\(Language,length.scaled\\)",names(mLang))]
names(mLang) = as.character(levels(len$Language))
t(t(sort(mLang)))
```

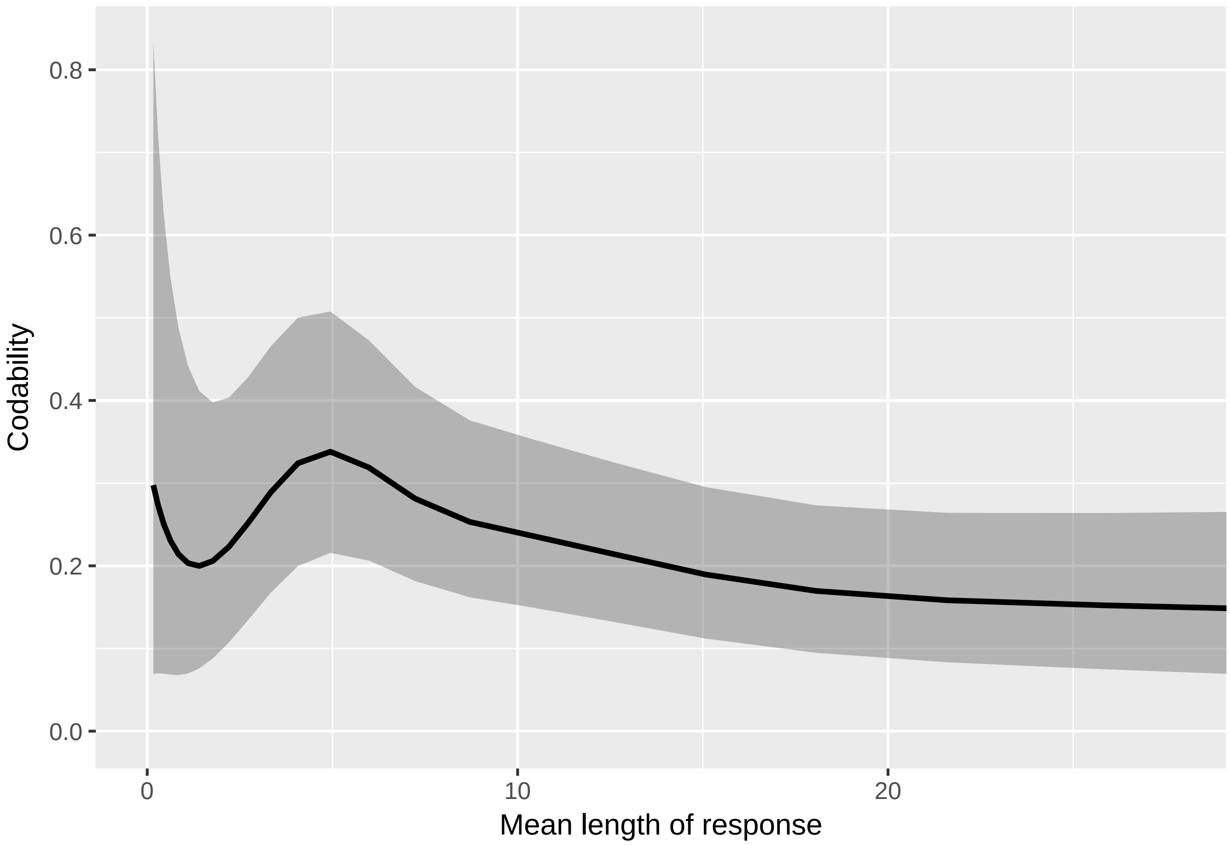
```
##                    [,1]
## Yurakare   -0.29214367
## Dogul Dom  -0.22843955
## Tzeltal    -0.17213762
## BSL        -0.12373691
## Kilivila   -0.11421187
## Turkish    -0.10026139
## Kata Kolok -0.07953295
## Semai      -0.07343907
## ASL        -0.07220170
## English    -0.05093427
## Farsi      -0.04204591
## Lao        -0.03968849
## Zapotec    -0.03518524
## Cantonese   0.05152590
## Malay       0.05983917
## Yucatec     0.12218071
## Siwu        0.13851096
## Mian        0.14155083
## Yeli Dnye   0.29353532
## Umpila      0.61681574
```

We can also use the `itsadug` library to plot the effect of length independent of the random effects. This also scales everything back into the original units, but cuts the length range to show 90% of the data (to hide the long tail)

```
convertLogDiversity = function(X){
  # Convert the scaled diversity measure
  #  back to the original units
  exp(X * attr(len$logDiversity.scaled,"scaled:scale") +
        attr(len$logDiversity.scaled,"scaled:center"))-0.1
}

px = plot_smooth(mLen5,view="length.scaled", rm.ranef = T, print.summary=F)
```




```
px$fv$fit = convertLogDiversity(px$fv$fit)
px$fv$ul = convertLogDiversity(px$fv$ul)
px$fv$ll = convertLogDiversity(px$fv$ll)
px$fv$length.scaled = exp(px$fv$length.scaled *
                            attr(len$length.scaled,"scaled:scale") +
                            attr(len$length.scaled,"scaled:center"))-0.5

gLen = ggplot(px$fv, aes(x=length.scaled,y=fit)) +
  geom_ribbon(aes(ymin=ll,ymax=ul), alpha=0.3) +
  geom_line(size=1) +
  ylab("Codability") +
  xlab("Mean length of response") +
  coord_cartesian(xlim=c(0,quantile(len$mean, 0.90)))
gLen
```




```
pdf("../results/graphs/Codability_by_Length.pdf",
    width = 4, height = 4)
gLen
dev.off()
```

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