

Explaining codability

Contents

Introduction	1
Load libraries	4
Load data	4
General properties	6
Specific domains	17
Colour	17
Colour Results	17
Shape	23
Shape Results	23
Specific hypotheses about shape	28
Sound	30
Sound Results	30
Touch	33
Touch Results	33
Taste	34
Taste Results	34
Specific taste hypotheses	35
Smell	37
Smell Results	37
Extra Graphs	43

Introduction

This set of analyses tries to explain the codability scores according to non-linguistic factors. A list of a-priori hypotheses about external factors that could influence agreement in each domain was compiled. The analysis of agreement across domains used a linear mixed effects model with random effects for domains nested within languages and for stimuli nested within domains. Predictor variables were added to the model as fixed effects and remained if they significantly improved the model according to log likelihood comparison.

The factors that were predicted to explain general levels of codability within a population are:

- Number of speakers of the language (from Ethnologue or field linguist's estimate)
- Formal schooling (low, medium, high)
- Subsistence type (hunter-gatherer, horticultural, stable agriculture, industrial, post-industrial)
- Environment type (closed, open, both)
- Environment subtype (urban, jungle, fields, rocky plateau, forest, forest & sea)
- Settlement type (nucleated, dispersed)
- Society has a market (yes, no)

- Society supports specialists (yes, no)

The Environment subtype variable was too varied to provide meaningful explanation in such a small sample, so is left out of this analysis.

For domains where predictor variables were highly co-linear, a two-step approach was taken. A random forests regression with random intercepts for languages was used to predict variation in agreement according to all a-priori predictor variables. Variable importance measures were used to identify key variables. The effect of these variables was tested with a full mixed effects model with random intercepts for language and stimulus.

The hypotheses for each domain were as follows:

Colour

- Presence of paints
- Number of paints (none, few, many)
- Presence of dyes
- Number of dyes (none, few, many)
- Ritual use of colour (yes, no)
- Professional colour (yes, no)
- Coloured (yes, no)
- Weaving patterns (yes, no)

Shape

- formal schooling (low, medium, high)
- pottery (yes, no)
- patterned pottery (yes, no)
- containers (yes, no)
- number of abstract shape categories (none, one, many)
- professional builders (yes, no)
- society makes boats (yes, no)
- boat specialists (yes, no)
- craft specialists (yes, no)
- spinning thread (yes, no)
- weaving (yes, no)
- weaving specialists (yes, no)
- weave patterns (yes, no)
- what weave patterns (none < simple < angular < complex)
- leatherware (yes, no)
- decorated.leatherware (yes, no)

In addition, it is predicted that communities living in round houses will have lower codability for angular shapes than those living in angular (square or rectangular) houses.

Sound

- musical instrument (yes, no)
- specialist musician (yes, no)
- training music (yes, no)
- children music (yes, no)
- animal sounds (yes, no)

Touch

- pulverise spices (yes, no)
- fine surfaces on houses (yes, no)
- professional textures (yes, no)

Taste

- pulverise.spices (yes, no)
- spices herbs (yes, no)
- Number of additives (0-5)
- sweet additive (for sweet stimulus, yes, no)
- salt additive (for salty stimulus, yes, no)
- bitter additive (for bitter stimulus, yes, no)
- sour additive (for sour stimulus, yes, no)
- umami additive (for umami stimulus, yes, no)
- fragrant food (yes, no)

In addition, having particular additives is predicted to increase the codability for the particular taste stimuli.

Smell

- pulverise spices (yes, no)
- spices herbs (yes, no)
- fragrant food (yes, no)
- Substance type (particularly hunter-gratherers)
- latitude (proxy for humidity)

Colinearity

These pairs of variables that are co-linear in our sample, meaning that it is redundant to add them both:

- “market”, “spices/herbs”
- “pottery coloured”, “spinning specialists”
- “pottery coloured”, “leatherware”
- “spinning specialists”, “leatherware”
- “decorated leatherware”, “what leatherware patterns?”
- “pulverise spices”, “sweet additive”
- “pulverise spices”, “sour additive”
- “sweet additive”, “sour additive”

Load libraries

```
library(party)
library(rpart.plot)
library(XLConnect)
library(reshape2)
library(ggplot2)
library(usdm)
library(REEMtree)
library(lme4)
library(sjPlot)
library(gridExtra)
```

Load data

```
s = read.csv("../data/DiversityIndices_ND.csv")

ethnography = read.csv("../data/ethnography/LoP_ethnography_processed.csv")

v.with.variation = apply(ethnography,2,function(X){length(unique(X))>1})

ethnography = ethnography[,v.with.variation]

ethnography$paints.cat =
  factor(ethnography$paints.cat,
    levels=c("none", 'few', 'many'),
    ordered = TRUE)
ethnography$dyes.cat =
  factor(ethnography$dyes.cat,
    levels=c("none", 'few', 'many'),
    ordered = TRUE)

ethnography$ritual.colour.cat =
  factor(ethnography$dyes.cat,
    levels=c("none", 'few', 'many'),
    ordered = TRUE)

ethnography$environment =
  factor(ethnography$environment,
    levels = c("closed", 'both', 'open'))

ethnography$subsistence =
  factor(ethnography$subsistence,
    levels = c("hunter-gatherer", "horticultural",
      'stable agriculture', 'industrial',
      "post-industrial"), ordered = T)

ethnography$formal.schooling =
  factor(ethnography$formal.schooling,
    levels = c("low", 'medium', 'high'), ordered = T)
```

```

ethnography$num.additives.scaled = scale(ethnography$num.additives)

ethnography$what.weave.patterns2 =
  factor(ethnography$what.weave.patterns2,
    levels=c("no", 'simple', 'angular', 'complex'),
    ordered = T)

ethnography$shape2 = "many"
ethnography$shape2[ethnography$shape == "no"] = "none"
ethnography$shape2[ethnography$shape %in% c("oblong", 'cylindrical', 'round')] = "one"
ethnography$shape2 = factor(ethnography$shape2,
  levels=c("none", 'one', 'many'),
  ordered = T)

# Add ethnography data to diversity scores
s = cbind(s,
  ethnography[
    match(s$Language, ethnography$Language),
    !names(ethnography) %in%
    c("Language", "Language.orig")])

s$pop.logcenter = scale(log(s$pop))

# Cut population into 3 categories
s$pop.cat = cut(s$pop.logcenter,
  quantile(s$pop.logcenter,
    probs=seq(0,1,length.out=4)),
  include.lowest = T,
  labels = c("Low", "Medium", "High"))

# Transform the diversity index to log scale
# (note that this does not actually change the results much)
s$simpson.diversityIndex.log = log(0.1+s$simpson.diversityIndex)
# scale and center
s$simpson.diversityIndex.log = scale(s$simpson.diversityIndex.log)

```

Set random seed:

```
set.seed(9999)
```

General properties

Use a mixed effects model to test whether the overall codability is affected by general properties of the communities. We start with a null model and add variables if their inclusion significantly improves the fit of the model.

We are quite conservative here, and add random effects for domains within languages and also for stimuli within domains.

```
m0 = lmer(simpson.diversityIndex.log~
          1 +
          (1 | Language/domain) +
          (1|domain/Stimulus.code),data = s)
mPop = update(m0, ~.+pop.logcenter)
anova(m0,mPop)

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

## Data: s
## Models:
## m0: simpson.diversityIndex.log ~ 1 + (1 | Language/domain) + (1 |
## m0:      domain/Stimulus.code)
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop:      pop.logcenter
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0      6 5909.4 5945.1 -2948.7   5897.4
## mPop     7 5896.1 5937.7 -2941.0   5882.1 15.353      1 8.918e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mEnv = update(mPop, ~.+environment)
anova(mPop,mEnv)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop:      pop.logcenter
## mEnv: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mEnv:      pop.logcenter + environment
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mPop     7 5896.1 5937.7 -2941.0   5882.1
## mEnv     9 5897.2 5950.8 -2939.6   5879.2 2.8746      2 0.2376

mEnvD = update(mPop, ~.+environment.details)
anova(mPop,mEnvD)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop:      pop.logcenter
## mEnvD: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mEnvD:      pop.logcenter + environment.details
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mPop     7 5896.1 5937.7 -2941   5882.1
```

```
## mEnvD 12 5900.0 5971.5 -2938 5876.0 6.0605 5 0.3004
mSett = update(mPop, ~.+settlement)
anova(mPop, mSett)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop: pop.logcenter
## mSett: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mSett: pop.logcenter + settlement
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mPop   7 5896.1 5937.7 -2941 5882.1
## mSett   8 5898.0 5945.7 -2941 5882.0 0.0121      1 0.9125

mMark = update(mPop, ~.+market)
anova(mPop, mMark)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop: pop.logcenter
## mMark: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mMark: pop.logcenter + market
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mPop   7 5896.1 5937.7 -2941.0 5882.1
## mMark   8 5896.8 5944.5 -2940.4 5880.8 1.2432      1 0.2649

mSpec = update(mPop, ~.+specialists)
anova(mPop, mSpec)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop: pop.logcenter
## mSpec: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mSpec: pop.logcenter + specialists
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mPop   7 5896.1 5937.7 -2941.0 5882.1
## mSpec   8 5897.9 5945.5 -2940.9 5881.9 0.1514      1 0.6972

mSubs = update(mPop, ~.+subsistence)
anova(mPop, mSubs)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop: pop.logcenter
## mSubs: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mSubs: pop.logcenter + subsistence
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
```

```
## mPop    7 5896.1 5937.7 -2941.0    5882.1
## mSubs 11 5898.9 5964.4 -2938.4    5876.9 5.1745      4      0.2699

mScho = update(mPop, ~.+formal.schooling)
anova(mPop,mScho)

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

## Data: s
## Models:
## mPop: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mPop:      pop.logcenter
## mScho: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 | domain/Stimulus.code) +
## mScho:      pop.logcenter + formal.schooling
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mPop    7 5896.1 5937.7 -2941.0    5882.1
## mScho    9 5894.3 5947.9 -2938.2    5876.3 5.7206      2    0.05725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The only significant variable is population size. We can also test whether the relationship survives allowing for random slopes by domain (random slopes by language don't make sense because population doesn't vary by language).

```
m0.popSlope = lmer(simpson.diversityIndex.log~
  1 +
  (1 | Language/domain) +
  (1 +pop.logcenter|domain/Stimulus.code),
  data = s)
mPop2 = update(m0.popSlope, ~.+pop.logcenter)
anova(m0.popSlope,mPop2)

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

## Data: s
## Models:
## m0.popSlope: simpson.diversityIndex.log ~ 1 + (1 | Language/domain) + (1 +
## m0.popSlope:      pop.logcenter | domain/Stimulus.code)
## mPop2: simpson.diversityIndex.log ~ (1 | Language/domain) + (1 + pop.logcenter |
## mPop2:      domain/Stimulus.code) + pop.logcenter
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0.popSlope 10 5751.5 5811.0 -2865.7    5731.5
## mPop2       11 5748.7 5814.2 -2863.3    5726.7 4.7884      1    0.02865 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(mPop2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## simpson.diversityIndex.log ~ (1 | Language/domain) + (1 + pop.logcenter |
##      domain/Stimulus.code) + pop.logcenter
## Data: s
##
## REML criterion at convergence: 5731.5
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
```



```

## -2.9443 -0.6193  0.0218  0.6113  3.6973
##
## Random effects:
##   Groups           Name              Variance Std.Dev. Corr
## Stimulus.code:domain (Intercept)    0.129513 0.35988
##                        pop.logcenter 0.040504 0.20126 -0.01
## domain:Language      (Intercept)    0.220798 0.46989
## Language             (Intercept)    0.006338 0.07961
## domain                (Intercept)    0.233937 0.48367
##                        pop.logcenter 0.039240 0.19809  0.80
## Residual                                0.331665 0.57590
## Number of obs: 2850, groups:
## Stimulus.code:domain, 147; domain:Language, 114; Language, 20; domain, 6
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)   -0.21071    0.20785  -1.014
## pop.logcenter  0.24036    0.09763   2.462
##
## Correlation of Fixed Effects:
##              (Intr)
## pop.logcntr  0.633

```

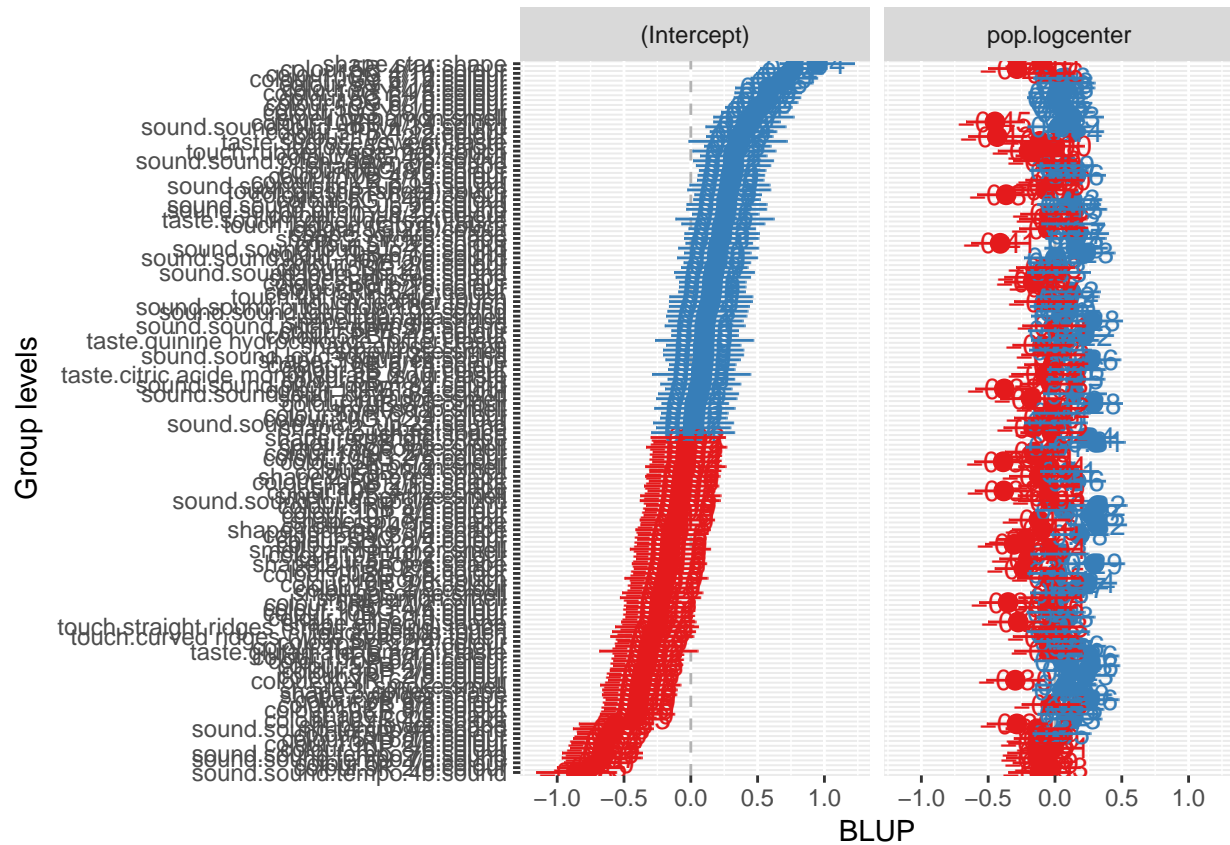
Statistical summary: There was a significant main effect of population size predicting general agreement (log likelihood difference = 2.4 , df = 1 , Chi Squared = 4.79 , p = 0.029).

Plot the random effects:

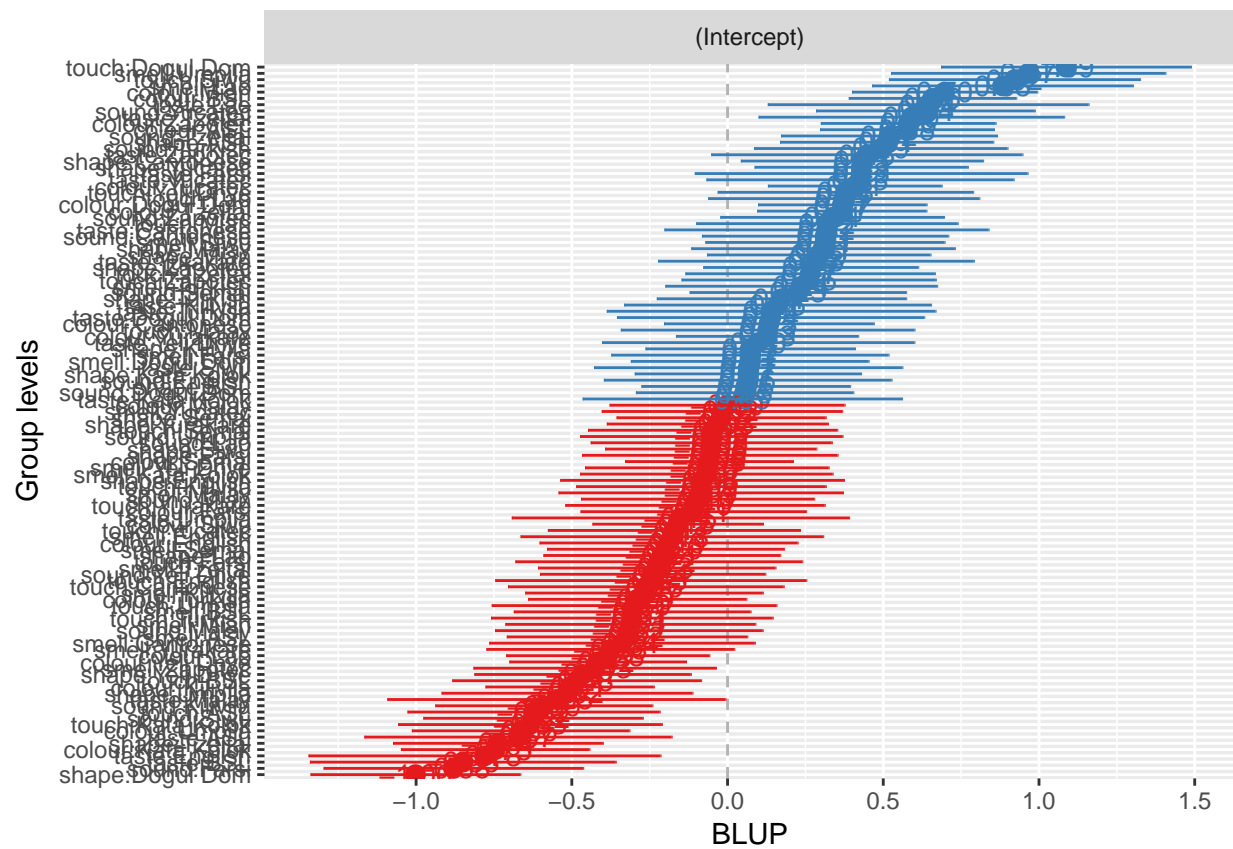
```
sjp.lmer(mPop2, 're', sort.est = "(Intercept)")
```

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

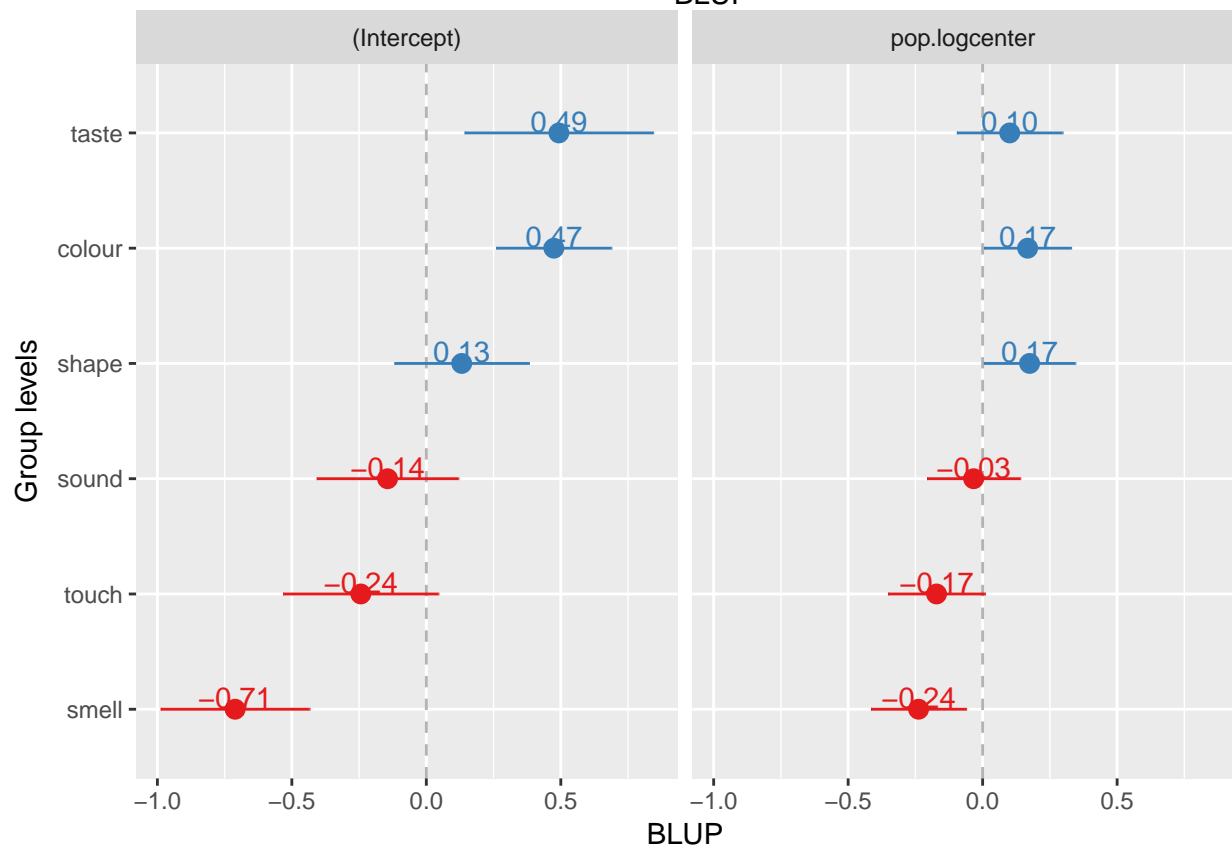
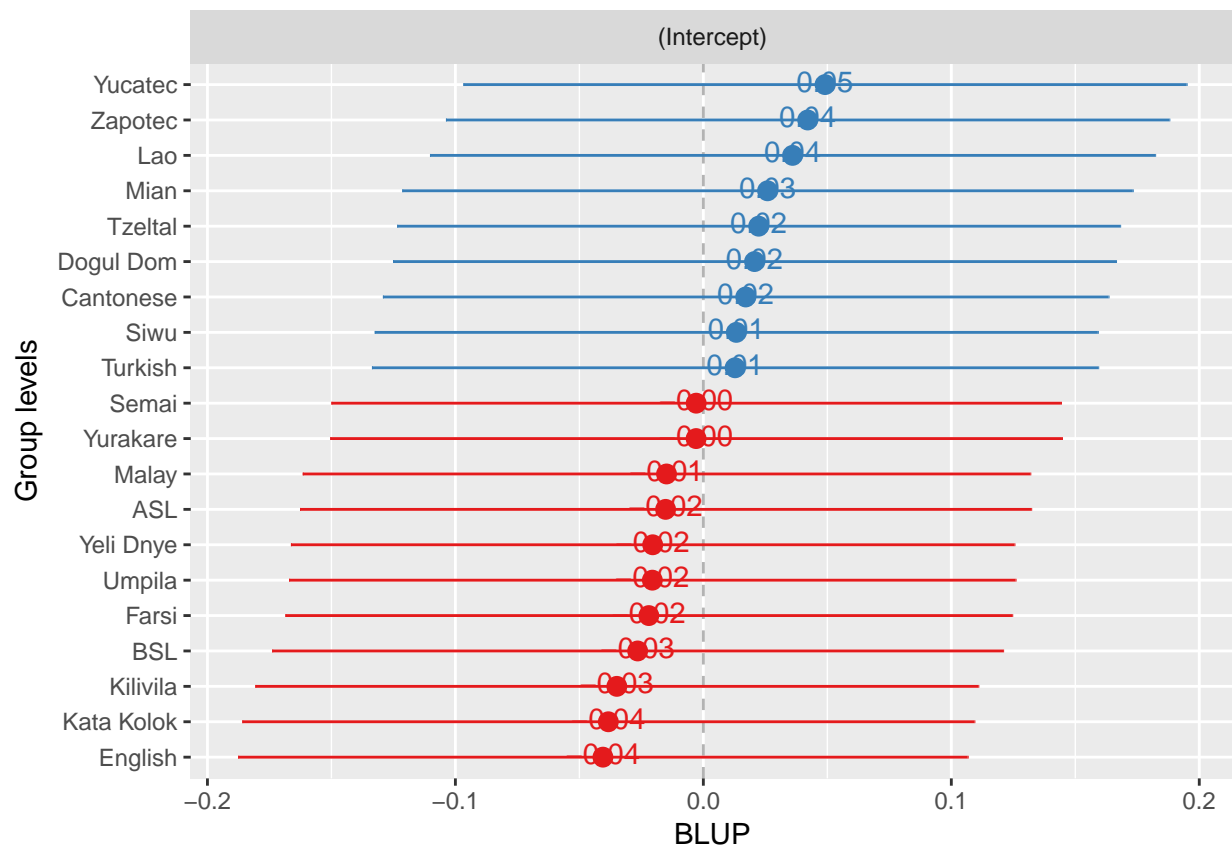
```
## Plotting random effects...
```



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



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



Plot the model effects for population size (extra code is to scale everything back into real numbers):

```

popxcod = data.frame(
  population=apply(s$pop.logcenter,s$Language, mean),
  mean.codability =
    tapply(s$simpson.diversityIndex.log,s$Language,mean))

popxcod$population = exp((popxcod$population*
  attr(s$pop.logcenter,'scaled:scale')) +
  attr(s$pop.logcenter,'scaled:center'))

popxcod$mean.codability = exp(((popxcod$mean.codability*
  attr(s$simpson.diversityIndex.log,'scaled:scale')) +
  attr(s$simpson.diversityIndex.log,'scaled:center')))-0.1

plotsX = sjs.lmer(mPop2, "eff", show.ci = T,
  show.scatter = T,prnt.plot=F,facet.grid=F)

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

plotsX$plot.list[[1]]$labels$title = ""

plotsX$plot.list[[1]]$data$x =
  exp((plotsX$plot.list[[1]]$data$x*
  attr(s$pop.logcenter,'scaled:scale')) +
  attr(s$pop.logcenter,'scaled:center'))

plotsX$plot.list[[1]]$data$y =
  exp(((plotsX$plot.list[[1]]$data$y*
  attr(s$simpson.diversityIndex.log,'scaled:scale')) +
  attr(s$simpson.diversityIndex.log,'scaled:center')))-0.1

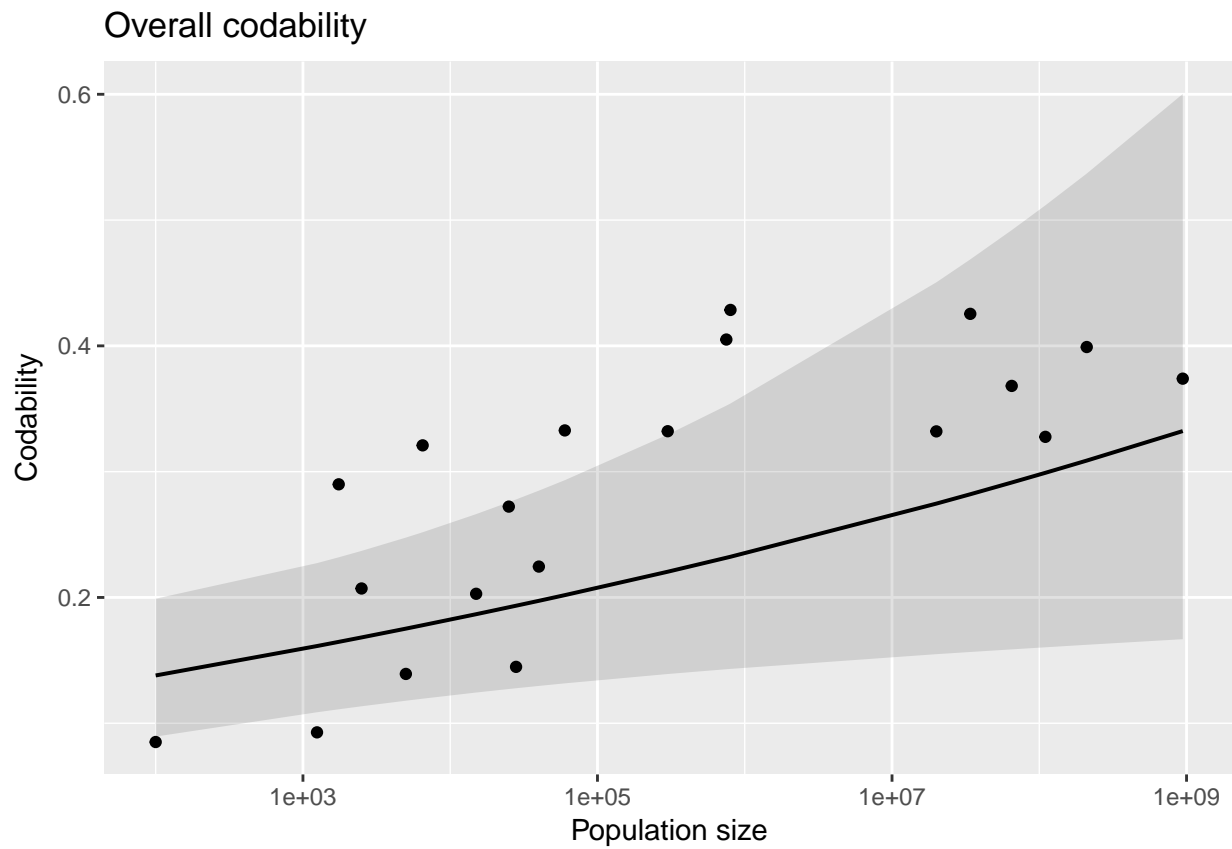
plotsX$plot.list[[1]]$data$lower =
  exp(((plotsX$plot.list[[1]]$data$lower*
  attr(s$simpson.diversityIndex.log,'scaled:scale')) +
  attr(s$simpson.diversityIndex.log,'scaled:center')))-0.1

plotsX$plot.list[[1]]$data$upper =
  exp(((plotsX$plot.list[[1]]$data$upper*
  attr(s$simpson.diversityIndex.log,'scaled:scale')) +
  attr(s$simpson.diversityIndex.log,'scaled:center')))-0.1

gx.pop = plotsX$plot.list[[1]] +
  scale_x_log10() +
  xlab("Population size") +
  ylab("Codability") +
  ggtitle("Overall codability") +
  geom_point(data=popxcod, aes(population,mean.codability))

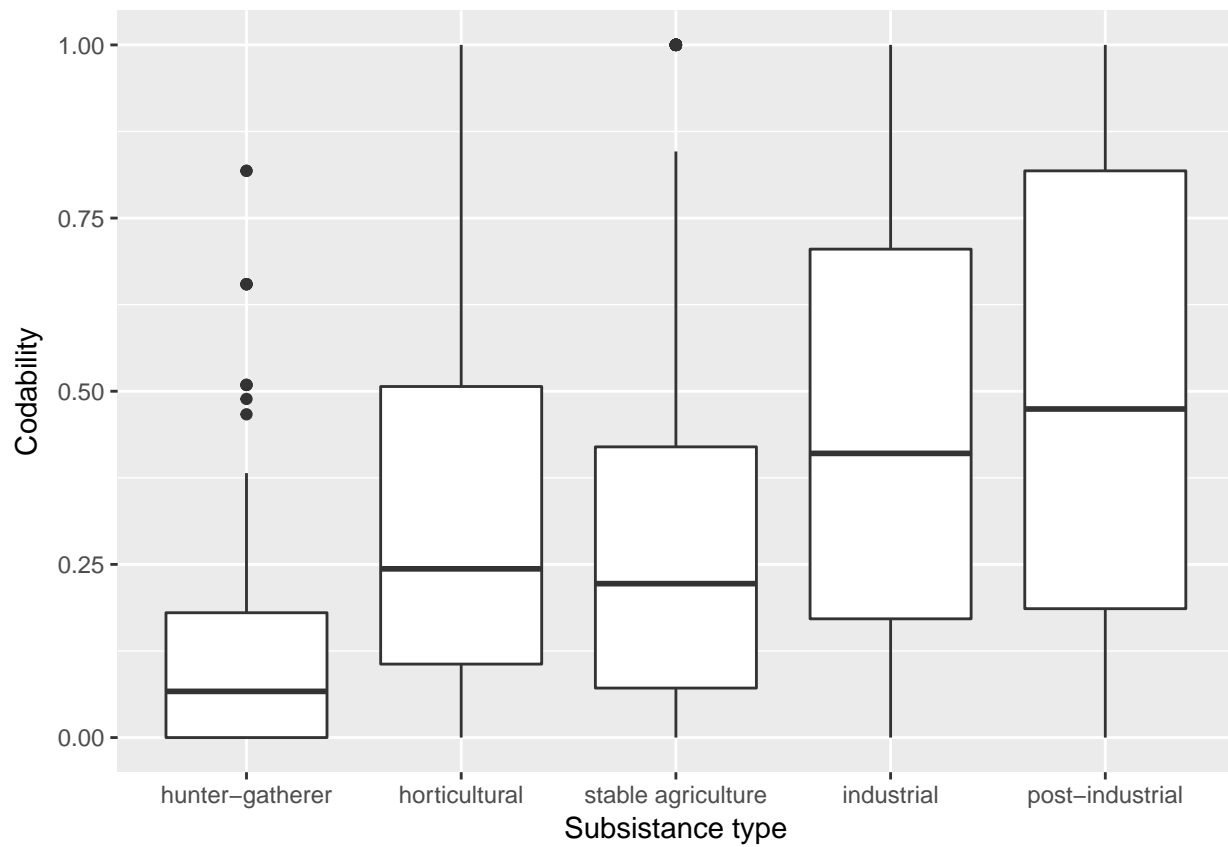
gx.pop

```

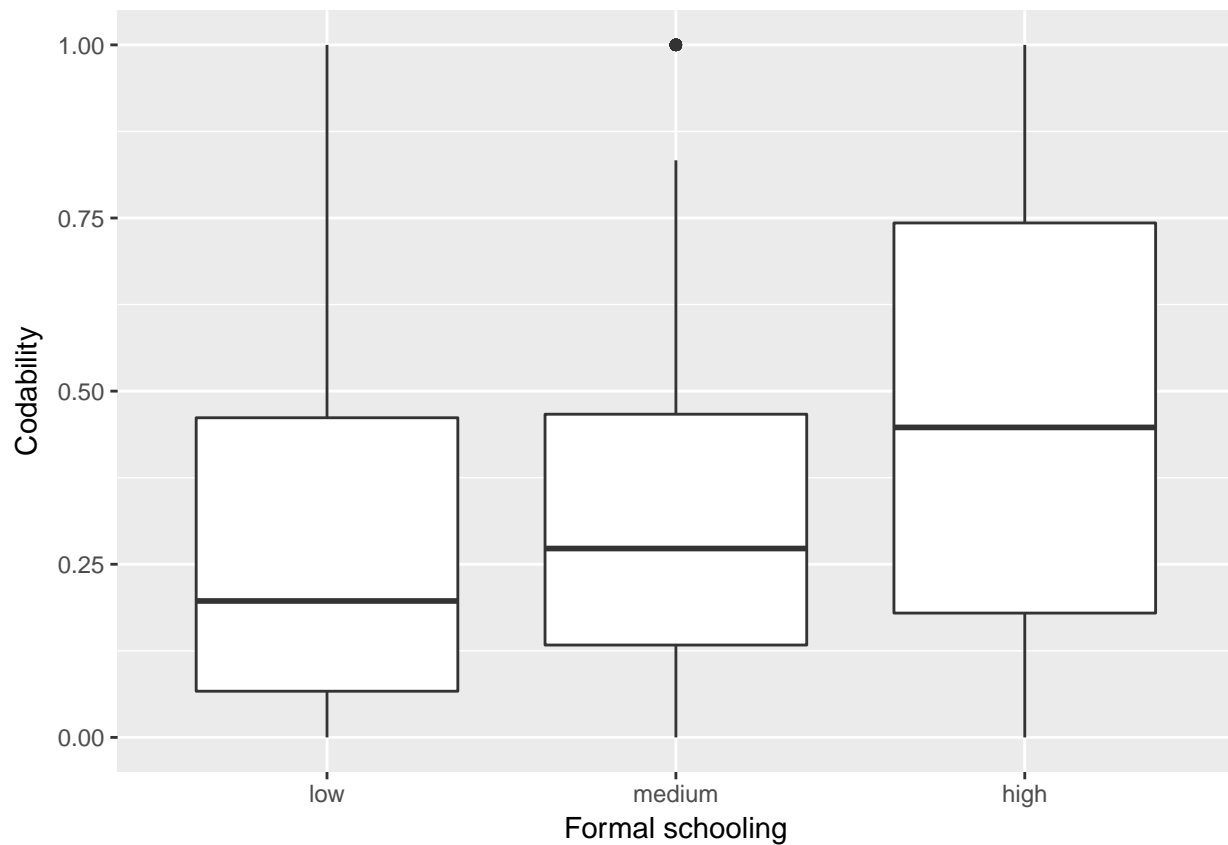


Plots for subsistence, formal schooling.

```
ggplot(s,  
  aes(subsistence,simpson.diversityIndex)) +  
  geom_boxplot() +  
  xlab("Subsistence type") +  
  ylab("Codability")
```



```
gg.formalschooling =  
  ggplot(s,  
    aes(formal.schooling,simpson.diversityIndex)) +  
  geom_boxplot() +  
  xlab("Formal schooling") +  
  ylab("Codability")  
gg.formalschooling
```



Correlation between population size and subsistence:

```
kruskal.test(ethnography$pop, ethnography$subsistance)
```

```
##  
##  Kruskal-Wallis rank sum test  
##  
## data:  ethnography$pop and ethnography$subsistance  
## Kruskal-Wallis chi-squared = 12.004, df = 4, p-value = 0.01732
```


Specific domains

There are many more variables to consider for the individual domains. Many are highly correlated, and there are many missing combinations in the data. This makes a regression approach difficult. Instead, we can use a binary decision tree to find clusters in the data based on salient properties. The package **REEMtree** also allows an additional random effect for Language (crossed random effects for stimulus type are not permitted). So we use random forests to identify key variables (or exclude unimportant variables), then mixed effects modelling to determine significance of main effects.

Note that we're using a categorical version of the population variable that splits the data into "small", "medium", "large". This is because the raw population variable is a continuous variable that can be used by a decision tree to split the languages into many arbitrary categories, giving it an unfair advantage over the other variables and making it difficult to interpret the plot.

Colour

Calculate the optimal decision tree, given variables related to colour:

```
s.colour = s[s$domain=='colour',]
rt = REEMtree(simpson.diversityIndex~
  formal.schooling +
  paints.cat+
  dyes.cat+
  ritual.colour.cat+
  professional.colour+
  pottery.coloured+
  weave.patterns,
  random = ~1|Language,
  data = s.colour,
  MaxIterations=100000)
```

Colour Results

Proportion of variance explained:

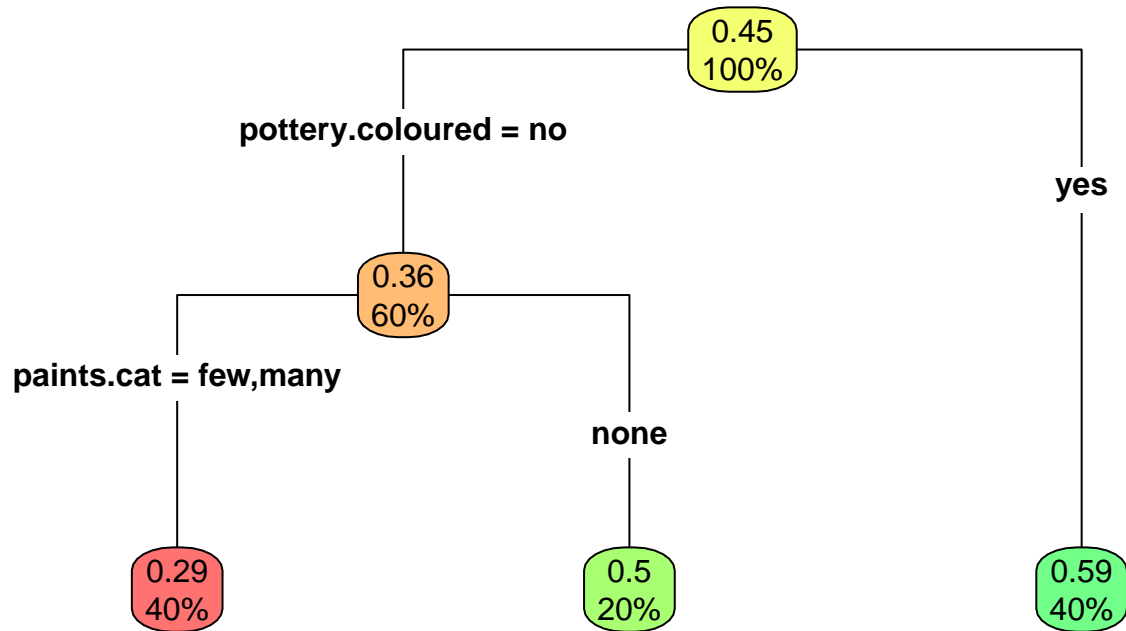
```
# R squared
cor(predict.REEMtree(rt,s.colour,id=s.colour$Language,
  EstimateRandomEffects = T),
  s.colour$simpson.diversityIndex)^2
```

```
## [1] 0.3132261
```

Plot the tree and calculate variable importance:

```
rpart.plot(tree(rt), type=4,extra=100, branch.lty=1, box.palette="RdYlGn", main="Colour")
```

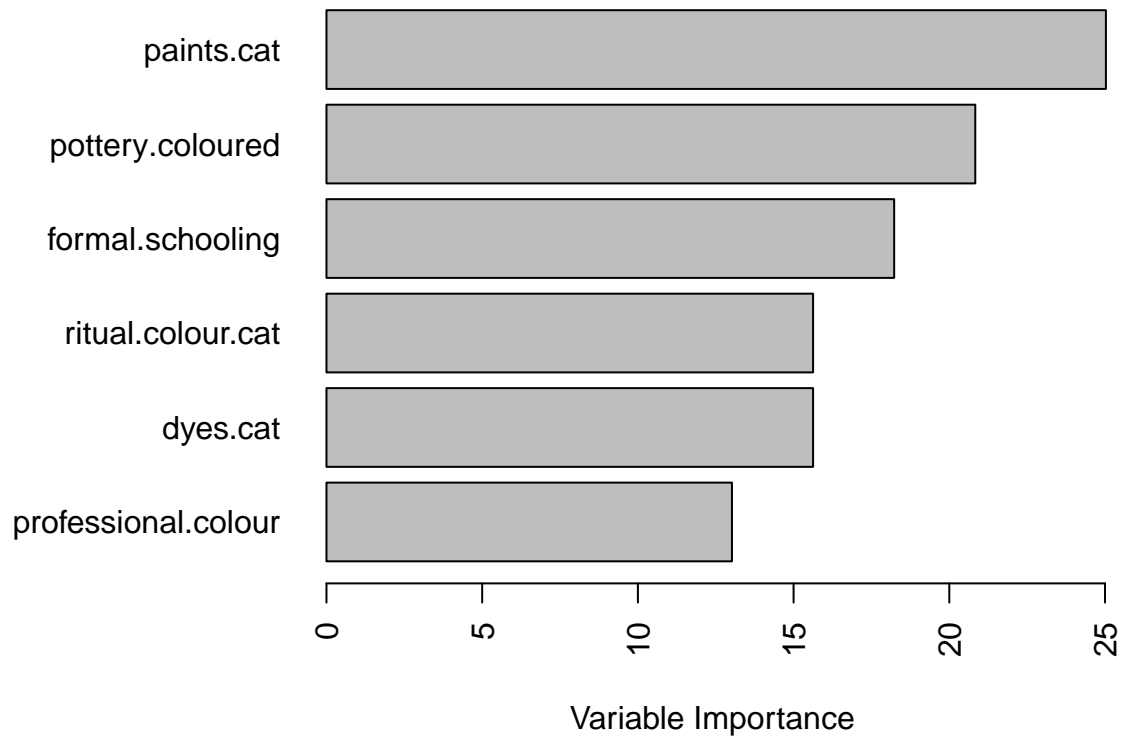
Colour



```

varimp = rt$Tree$variable.importance

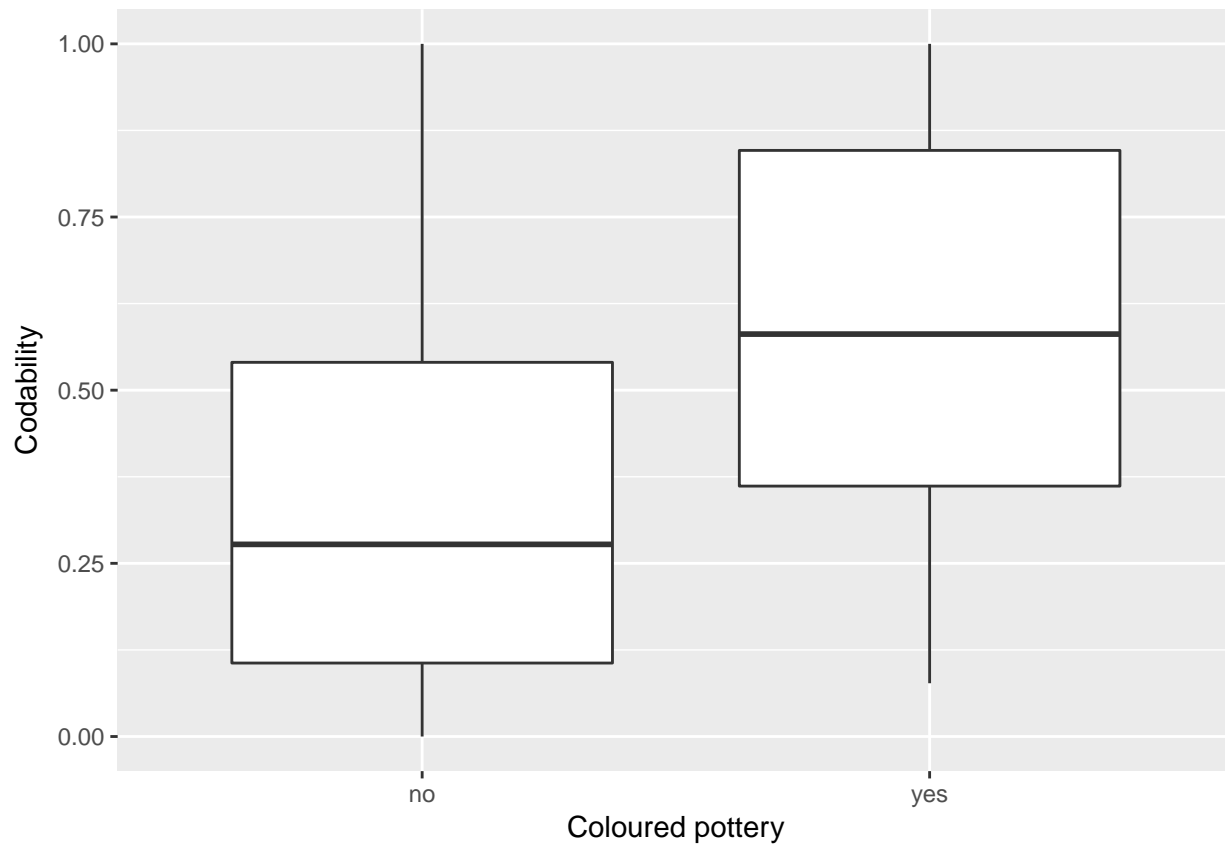
par(mar=c(5,10,2,2))
barplot(sort(varimp), horiz=T, las=2,
        xlab="Variable Importance")
  
```



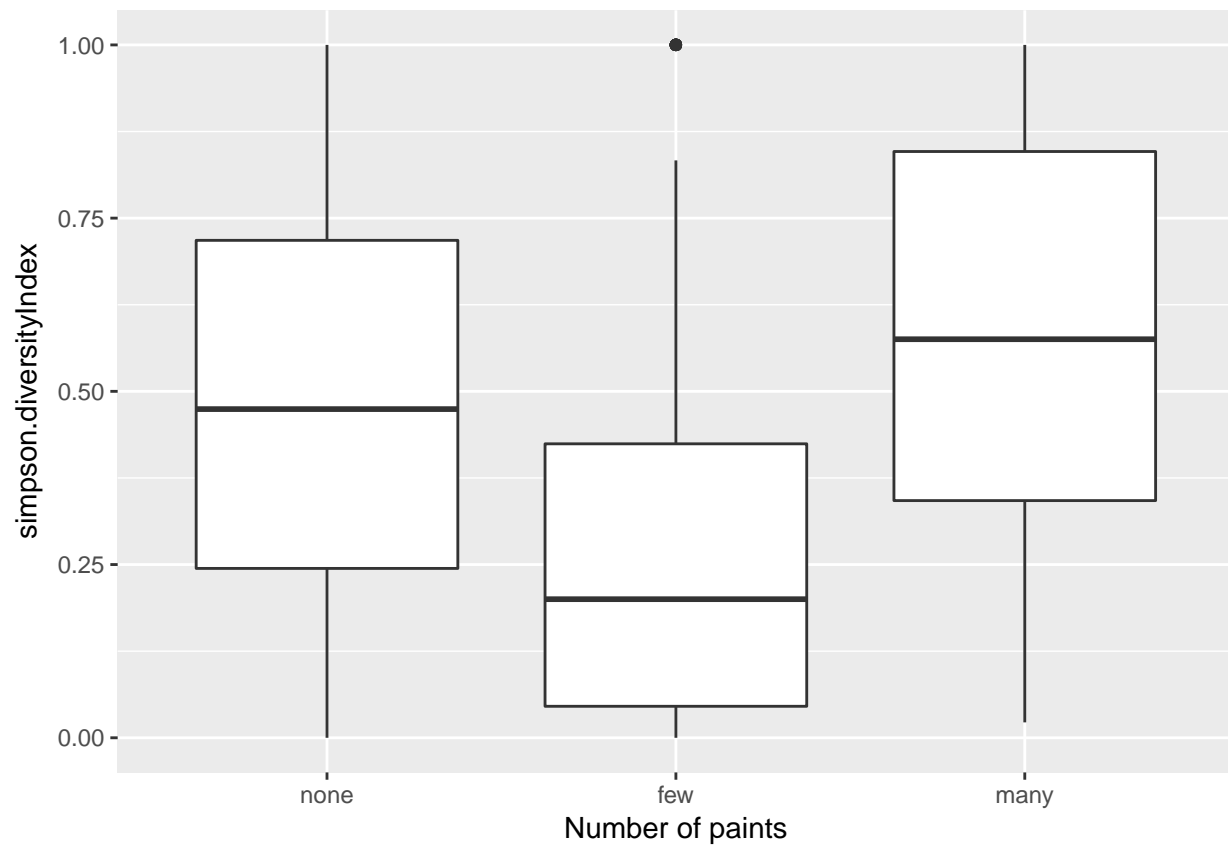
```
par(mar=c(5, 4, 4, 2) + 0.1)
```

Plot the relationships:

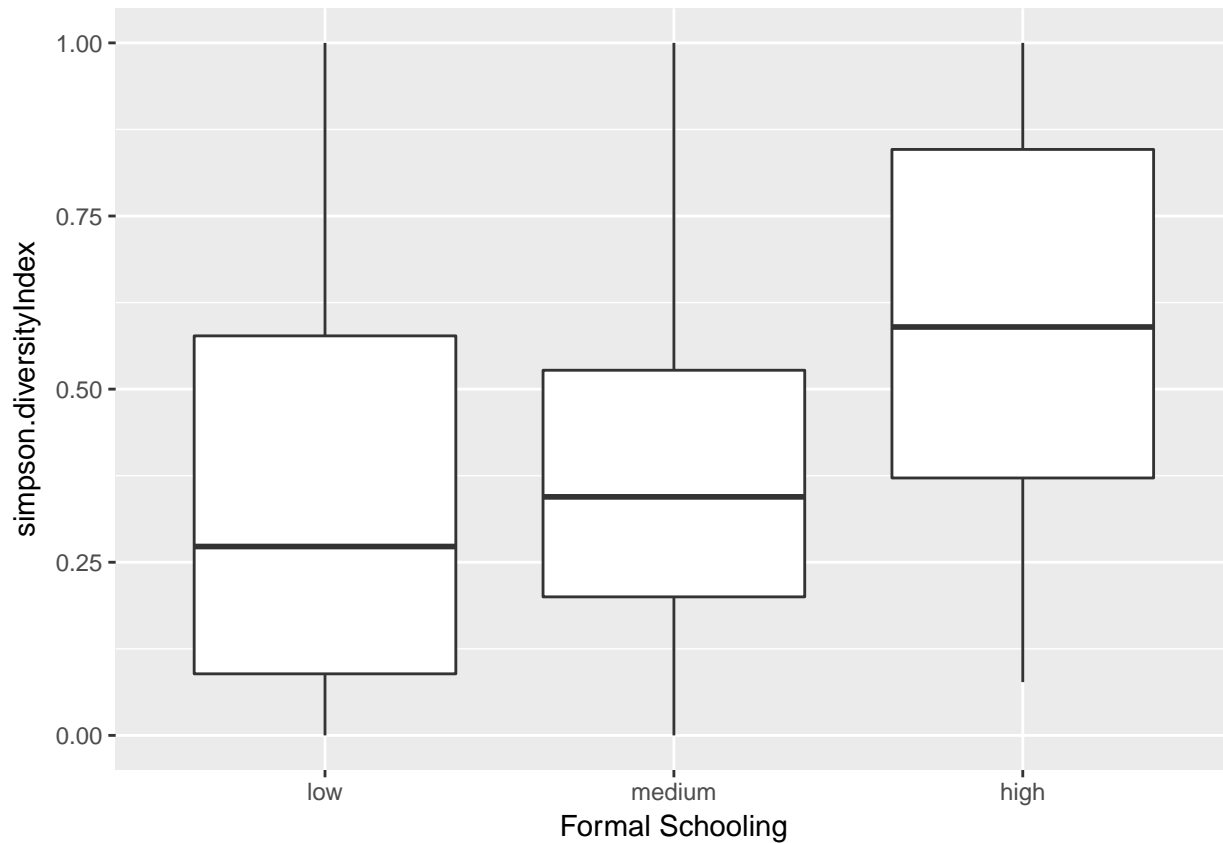
```
ggplot(s.colour,  
       aes(pottery.coloured,simpson.diversityIndex)) +  
  geom_boxplot() +  
  xlab("Coloured pottery") +  
  ylab("Codability")
```



```
ggplot(s.colour,  
       aes(paints.cat,simpson.diversityIndex)) +  
  geom_boxplot() +  
  xlab("Number of paints")
```



```
ggplot(s.colour,  
  aes(formal.schooling,simpson.diversityIndex)) +  
  geom_boxplot() +  
  xlab("Formal Schooling")
```



Test the most important variables with a mixed effects model:

```
mc.all = lmer(simpson.diversityIndex.log ~  
  1 +  
    paints.cat +  
    pottery.coloured +  
    formal.schooling +  
    (1 | Language) +  
    (1 | Stimulus.code),  
  data = s.colour)  
  
mc.noPaints = lmer(simpson.diversityIndex.log ~  
  1 +  
    pottery.coloured +  
    formal.schooling +  
    (1 | Language) +  
    (1 | Stimulus.code),  
  data = s.colour)  
  
mc.noSchool = lmer(simpson.diversityIndex.log ~  
  1 +  
    pottery.coloured +  
    paints.cat +  
    (1 | Language) +  
    (1 | Stimulus.code),  
  data = s.colour)
```

```

mc.noPott = lmer(simpson.diversityIndex.log ~
  1 +
  paints.cat +
  formal.schooling +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.colour)

anova(mc.all, mc.noPaints)

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

## Data: s.colour
## Models:
## mc.noPaints: simpson.diversityIndex.log ~ 1 + pottery.coloured + formal.schooling +
## mc.noPaints:      (1 | Language) + (1 | Stimulus.code)
## mc.all: simpson.diversityIndex.log ~ 1 + paints.cat + pottery.coloured +
## mc.all:      formal.schooling + (1 | Language) + (1 | Stimulus.code)
##          Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mc.noPaints  7 3396.1 3433.8 -1691.1   3382.1
## mc.all       9 3390.2 3438.6 -1686.1   3372.2 9.925    2  0.006996 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(mc.all, mc.noSchool)

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

## Data: s.colour
## Models:
## mc.noSchool: simpson.diversityIndex.log ~ 1 + pottery.coloured + paints.cat +
## mc.noSchool:      (1 | Language) + (1 | Stimulus.code)
## mc.all: simpson.diversityIndex.log ~ 1 + paints.cat + pottery.coloured +
## mc.all:      formal.schooling + (1 | Language) + (1 | Stimulus.code)
##          Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mc.noSchool  7 3386.7 3424.3 -1686.3   3372.7
## mc.all       9 3390.2 3438.6 -1686.1   3372.2 0.5037    2  0.7774

anova(mc.all, mc.noPott)

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

## Data: s.colour
## Models:
## mc.noPott: simpson.diversityIndex.log ~ 1 + paints.cat + formal.schooling +
## mc.noPott:      (1 | Language) + (1 | Stimulus.code)
## mc.all: simpson.diversityIndex.log ~ 1 + paints.cat + pottery.coloured +
## mc.all:      formal.schooling + (1 | Language) + (1 | Stimulus.code)
##          Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mc.noPott   8 3388.4 3431.4 -1686.2   3372.4
## mc.all      9 3390.2 3438.6 -1686.1   3372.2 0.1791    1  0.6721

```

There is only a contribution of number of paints beyond population size, but this goes in the opposite direction of the predicted one: societies with few paints have fewer categories than societies with none or many. There was a significant main effect of number of paints (log likelihood difference = 5 , df = 2 , Chi Squared = 9.92 , p = 0.007).

Shape

Calculate the optimal decision tree, given variables related to shape:

```
s.shape = s[s$domain=='shape',]

rt.shape = REEMtree(simpson.diversityIndex~
  formal.schooling +
  pottery +
  pottery.patterned +
  containers +
  shape2 +
  professional.builders +
  make.boats+
  boat.specialists+
  craft.specialists+
  spinning.thread+
  weaving+
  weaving.specialists+
  weave.patterns+
  what.weave.patterns2+
  leatherware+
  decorated.leatherware,
  random = ~1|Language,
  data = s.shape,
  MaxIterations=100000)
```

Shape Results

Proportion of variance explained:

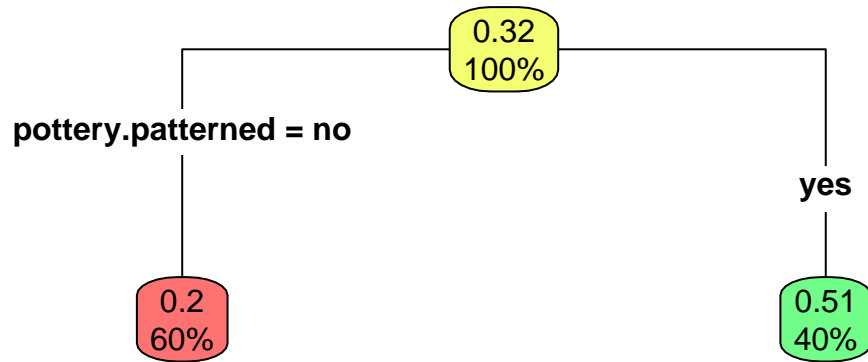
```
# R squared
cor(predict.REEMtree(rt.shape,s.shape,id=s.shape$Language, EstimateRandomEffects = T),
  s.shape$simpson.diversityIndex)^2
```

```
## [1] 0.410091
```

Plot the tree and calculate variable importance:

```
rpart.plot(tree(rt.shape), type=4,extra=100, branch.lty=1, box.palette="RdYlGn", main="Shape")
```

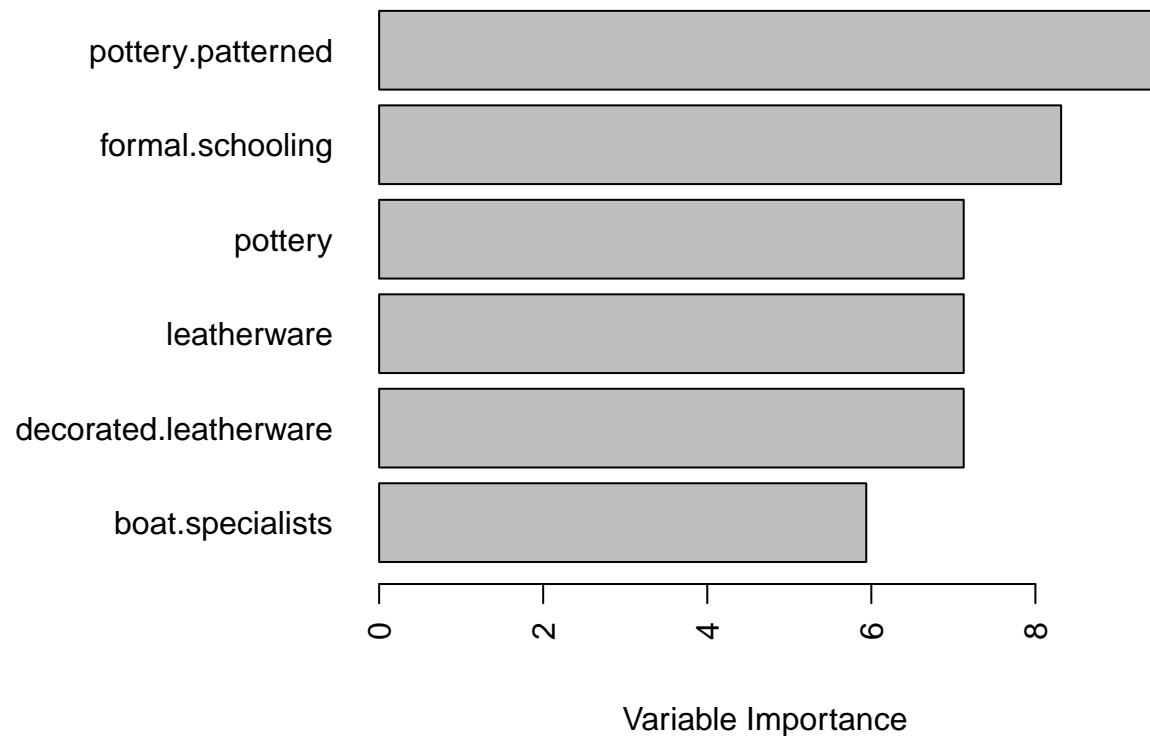
Shape



```

varimp = rt.shape$Tree$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)
```

Both `pottery.patterned` and `formal.schooling` have high importance. The two variables are highly co-linear (there are no societies with 'high' formal schooling without patterned pottery):

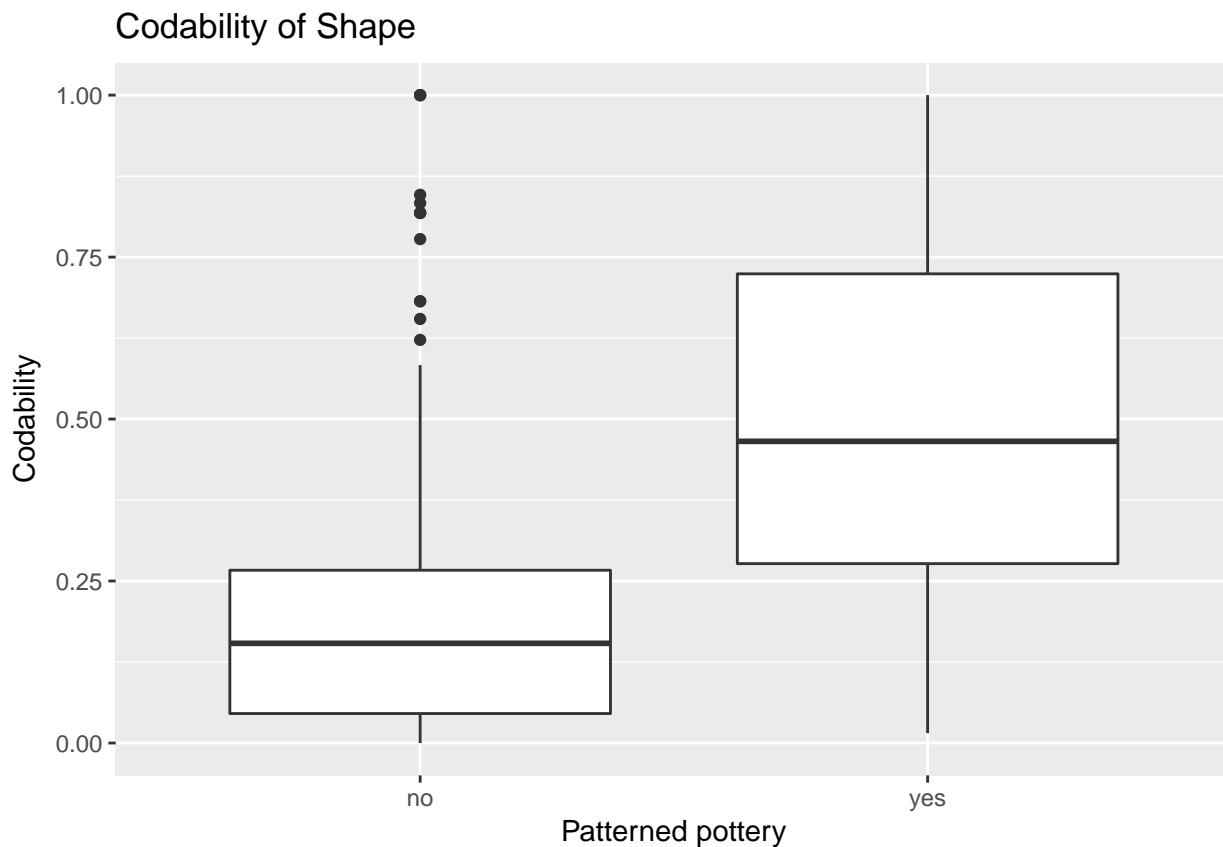

```
table(s$formal.schooling, s$pottery.patterned)
```

```
##
##           no  yes
##    low    1131 147
##   medium   583   0
##    high      0 989
```

So we can look at both:

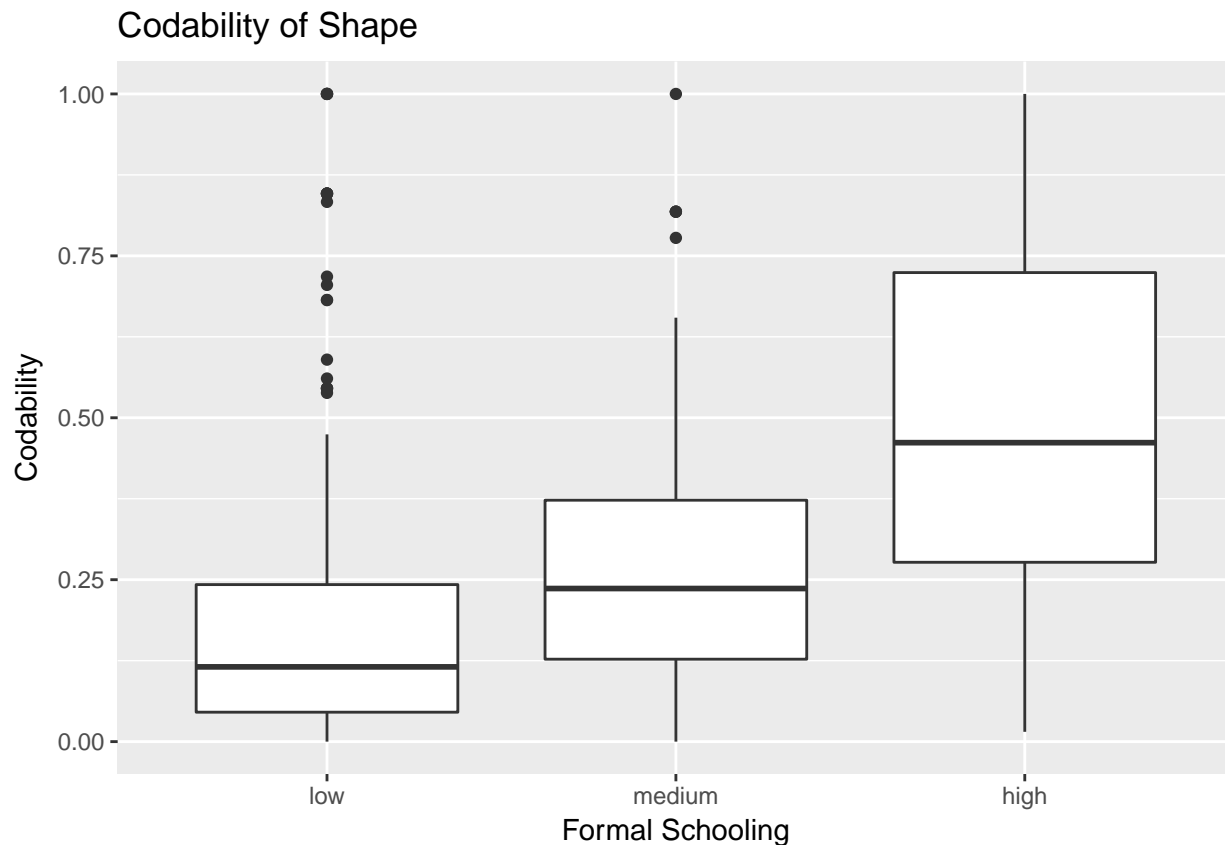
```
gx.shape = ggplot(s.shape, aes(pottery.patterned, simpson.diversityIndex))+
  geom_boxplot() +
  xlab("Patterned pottery") +
  ylab("Codability")+
  ggtitle("Codability of Shape")
```

```
gx.shape
```



```
gx.shape2 = ggplot(s.shape, aes(formal.schooling, simpson.diversityIndex))+
  geom_boxplot() +
  xlab("Formal Schooling") +
  ylab("Codability") +
  ggtitle("Codability of Shape")
```

```
gx.shape2
```



And use a mixed effects model to test the contribuion of each:

```
m.both = lmer(simpson.diversityIndex.log ~
  pottery.patterned +
  formal.schooling +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.shape)

m.noSchool = lmer(simpson.diversityIndex.log ~
  pottery.patterned +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.shape)

m.noPPottery = lmer(simpson.diversityIndex.log ~
  formal.schooling +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.shape)

anova(m.both,m.noSchool)

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

## Data: s.shape
## Models:
## m.noSchool: simpson.diversityIndex.log ~ pottery.patterned + (1 | Language) +
```

```

## m.noSchool:      (1 | Stimulus.code)
## m.both: simpson.diversityIndex.log ~ pottery.patterned + formal.schooling +
## m.both:      (1 | Language) + (1 | Stimulus.code)
##           Df      AIC      BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## m.noSchool  5 864.60 884.56 -427.30   854.60
## m.both      7 861.83 889.77 -423.92   847.83 6.771      2   0.03386 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m.both,m.noPPottery)

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

## Data: s.shape
## Models:
## m.noPPottery: simpson.diversityIndex.log ~ formal.schooling + (1 | Language) +
## m.noPPottery:      (1 | Stimulus.code)
## m.both: simpson.diversityIndex.log ~ pottery.patterned + formal.schooling +
## m.both:      (1 | Language) + (1 | Stimulus.code)
##           Df      AIC      BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## m.noPPottery  6 869.44 893.39 -428.72   857.44
## m.both      7 861.83 889.77 -423.92   847.83 9.6056      1   0.00194 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.both)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## simpson.diversityIndex.log ~ pottery.patterned + formal.schooling +
##      (1 | Language) + (1 | Stimulus.code)
##      Data: s.shape
##
## REML criterion at convergence: 854.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.73983 -0.68983  0.02328  0.68054  2.88331
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
##      Language      (Intercept) 0.1292   0.3595
##      Stimulus.code (Intercept) 0.1163   0.3410
##      Residual              0.4070   0.6380
## Number of obs: 400, groups: Language, 20; Stimulus.code, 20
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    -0.574896   0.193048  -2.978
## pottery.patternedyes  1.305659   0.410185   3.183
## formal.schooling.L  -0.003141   0.292338  -0.011
## formal.schooling.Q  -0.499201   0.256664  -1.945
##
## Correlation of Fixed Effects:
##              (Intr) pttry. frm..L
## pttry.pttrn -0.787

```

```
## frml.schl.L 0.714 -0.882
## frml.schl.Q 0.469 -0.725 0.658
```

```
# R squared:
cor(s.shape$simpson.diversityIndex.log, predict(m.both))^2
```

```
##           [,1]
## [1,] 0.6122963
```

```
# M
tapply(s.shape$simpson.diversityIndex, s.shape$formal.schooling, mean)
```

```
##           low      medium      high
## 0.1946866 0.2803914 0.5109820
```

There was a significant main effect of schooling (log likelihood difference = 3.4 , df = 2 , Chi Squared = 6.77 , p = 0.034).

There was a significant main effect of patterned pottery (log likelihood difference = 4.8 , df = 1 , Chi Squared = 9.61 , p = 0.0019).

Specific hypotheses about shape

Natural objects have (mostly) organic rounded shapes. Living in square/rectangular houses should give more names for angular shaped objects. We test whether living in rounded houses predicts codability for angular shapes. Note that there is only one community which lives in round houses (Umpila). Umpila are Hunter-gatherers, who are not necessarily low-codability across the board (they have very high codability for smell).

```
s.shape$angularShapes = s.shape$Stimulus.code %in%
  c("shape.2 cubes",
    "shape.3 squares",
    'shape.cube',
    'shape.square',
    'shape.rectangle',
    'shape.rectangle 3D')

# Null model
m0 = lmer(simpson.diversityIndex.log ~
  1 +
  (1 | Language) +
  (1 + I(houses=="round") | Stimulus.code),
  data = s.shape[s.shape$angularShapes,])

# Add fixed effect
m1 = lmer(simpson.diversityIndex.log ~
  1 +
  I(houses=="round") +
  (1 | Language) +
  (1 + I(houses=="round") | Stimulus.code),
  data = s.shape[s.shape$angularShapes,])
anova(m0, m1)
```

```
## refitting model(s) with ML (instead of REML)
## Data: s.shape[s.shape$angularShapes, ]
## Models:
## m0: simpson.diversityIndex.log ~ 1 + (1 | Language) + (1 + I(houses ==
## m0: "round") | Stimulus.code)
```

```
## m1: simpson.diversityIndex.log ~ 1 + I(houses == "round") + (1 |
## m1:      Language) + (1 + I(houses == "round") | Stimulus.code)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0  6 256.70 273.43 -122.35  244.70
## m1  7 254.77 274.29 -120.39  240.77 3.9308      1  0.04741 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: simpson.diversityIndex.log ~ 1 + I(houses == "round") + (1 |
##      Language) + (1 + I(houses == "round") | Stimulus.code)
##      Data: s.shape[s.shape$angularShapes, ]
##
## REML criterion at convergence: 240.8
##
## Scaled residuals:
##      Min      1Q   Median      3Q      Max
## -2.87221 -0.58415 -0.00682  0.59798  2.95397
##
## Random effects:
##      Groups      Name                Variance Std.Dev. Corr
##      Language      (Intercept)          0.645532 0.80345
##      Stimulus.code (Intercept)          0.009365 0.09677
##                  I(houses == "round")TRUE 0.017855 0.13362 -1.00
##      Residual                        0.276179 0.52553
## Number of obs: 120, groups:  Language, 20; Stimulus.code, 6
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)          -0.01047    0.19483  -0.054
## I(houses == "round")TRUE -1.69196    0.85495  -1.979
##
## Correlation of Fixed Effects:
##                  (Intr)
## I(=="")TRUE -0.231
```

Very weak significant effect for communities with rounded houses to have less agreement on angular shapes than those living in angular houses. Mean codability for angular houses = 0.35153; mean codability for round houses = 0.0121212

```
gx.shape.house = ggplot(s.shape[s.shape$angularShapes,],
  aes(houses!="round",simpson.diversityIndex)) +
  xlab("House shape") +
  scale_x_discrete(labels=c("Round", "Angular")) +
  ylab("Codability") +
  geom_boxplot()
```

Sound

Calculate the optimal decision tree, given variables related to shape:

```
s.sound = s[s$domain=='sound',]  
  
rt.sound = REEMtree(simpson.diversityIndex~  
  musical.instrument +  
  specialist.musician+  
  training.music+  
  children.music+  
  animal.sounds,  
  random = ~1|Language,  
  data = s.sound,  
  MaxIterations=100000,  
  tree.control = rpart.control(maxdepth = 1))
```

Sound Results

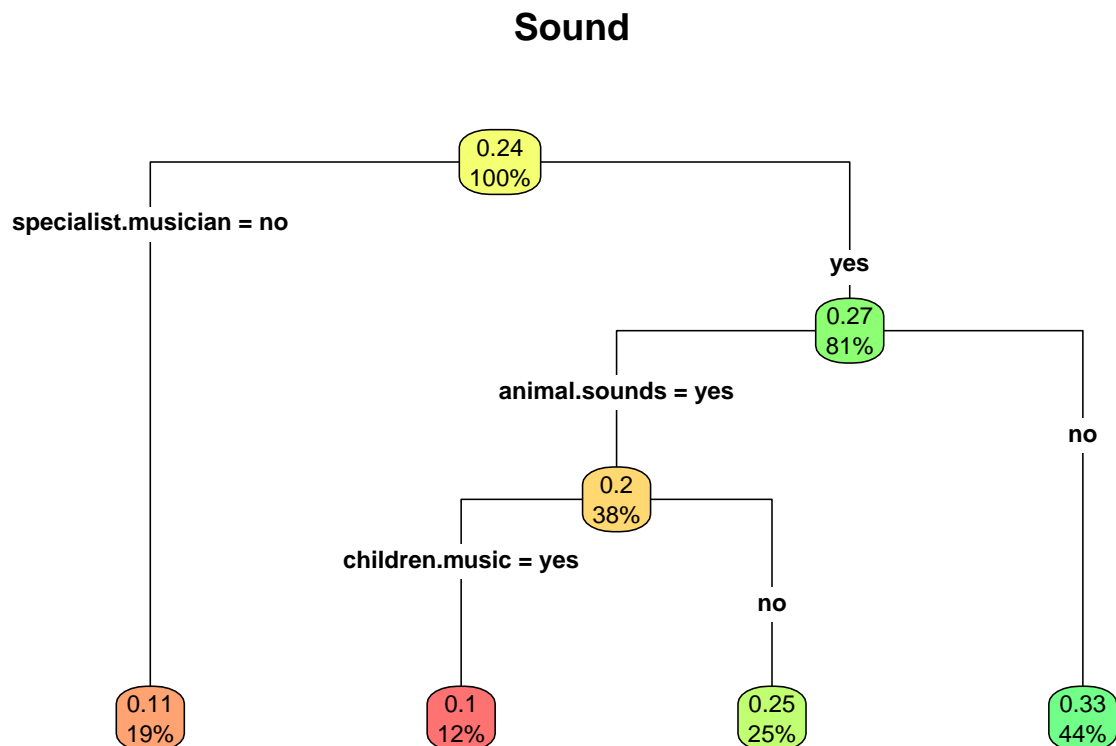
Proportion of variance explained:

```
# R squared  
cor(predict.REEMtree(rt.sound,s.sound,id=s.sound$Language, EstimateRandomEffects = T),  
  s.sound$simpson.diversityIndex)^2
```

```
## [1] 0.4052415
```

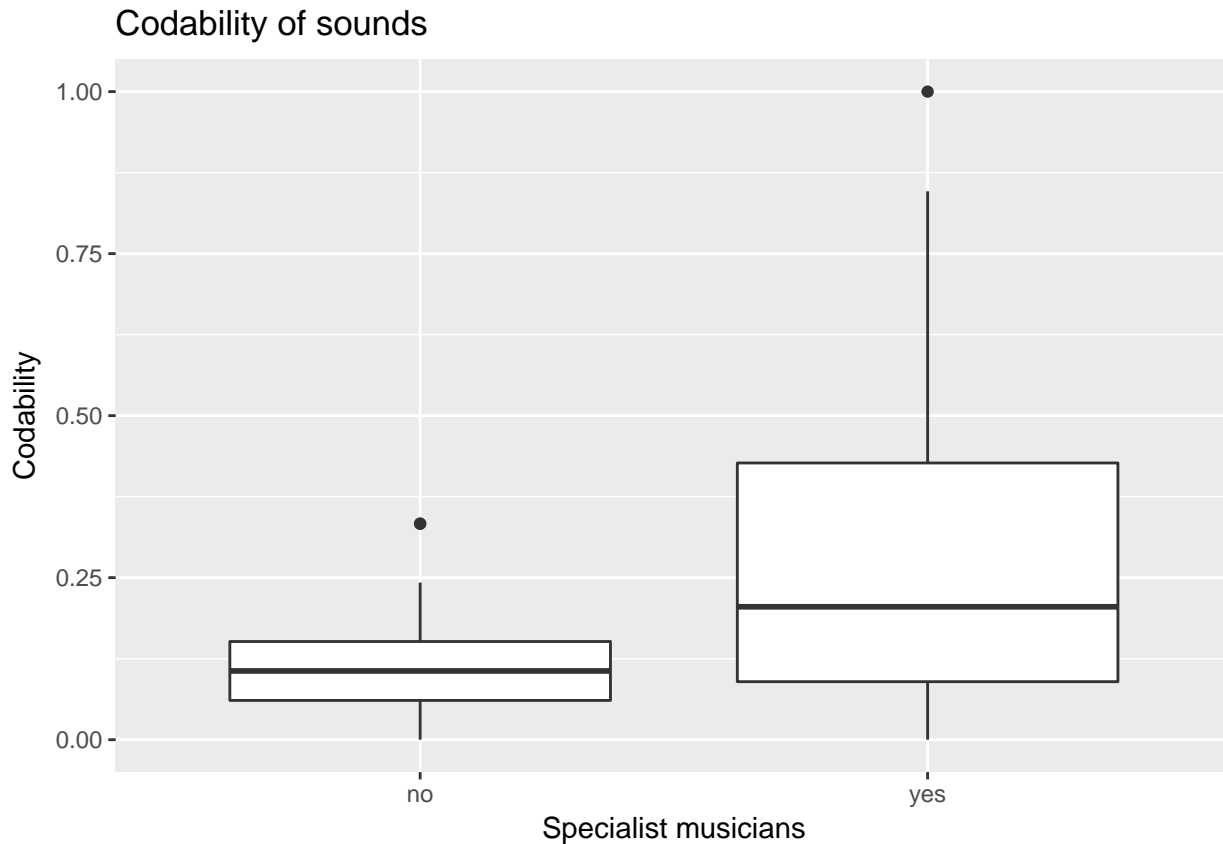
Plot the tree and calculate variable importance:

```
rpart.plot(tree(rt.sound), type=4,extra=100,  
  branch.lty=1, box.palette="RdYlGn", main="Sound")
```



This suggests that the training of musicians and having specialist musicians is important. Training of musicians depends on having specialist musicians, so we concentrate on the latter.

```
gx.sound = ggplot(s.sound, aes(specialist.musician, simpson.diversityIndex))+  
  geom_boxplot() +  
  xlab("Specialist musicians") +  
  ylab("Codability") +  
  ggtitle("Codability of sounds")  
gx.sound
```



Test with mixed effects modelling:

```
mSd.null = lmer(simpson.diversityIndex.log ~  
  1 +  
  (1 | Language) +  
  (1 | Stimulus.code),  
  data = s.sound)  
  
mSd.music = lmer(simpson.diversityIndex.log ~  
  1 +  
  specialist.musician +  
  (1 | Language) +  
  (1 | Stimulus.code),  
  data = s.sound)  
  
anova(mSd.null, mSd.music)
```

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

```
## Data: s.sound
## Models:
## mSd.null: simpson.diversityIndex.log ~ 1 + (1 | Language) + (1 | Stimulus.code)
## mSd.music: simpson.diversityIndex.log ~ 1 + specialist.musician + (1 | Language) +
## mSd.music:      (1 | Stimulus.code)
##           Df      AIC      BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mSd.null    4 540.57 555.64 -266.29   532.57
## mSd.music    5 538.47 557.31 -264.24   528.47 4.1008      1   0.04286 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##M
tapply(s.sound$simpson.diversityIndex,s.sound$specialist.musician,mean)

##           no           yes
## 0.1101010 0.2709464
```


Touch

Calculate the optimal decision tree, given variables related to shape:

```
s.touch = s[s$domain=='touch',]  
  
rt.touch = REEMtree(simpson.diversityIndex~  
  pulverise.spices +  
  fine-surfaces.on.houses +  
  professional.textures,  
  random = ~1|Language,  
  data = s.touch,  
  MaxIterations=100000)
```

Touch Results

Proportion of variance explained:

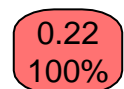
```
# R squared  
cor(predict.REEMtree(rt.touch,s.touch,id=s.touch$Language, EstimateRandomEffects = T),  
  s.touch$simpson.diversityIndex)^2
```

```
## [1] 0.4448271
```

Plot the tree and calculate variable importance:

```
rpart.plot(tree(rt.touch), type=4,extra=100, branch.lty=1, box.palette="RdYlGn", main="Touch")
```

Touch



```
#varimp = rt.touch$Tree$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)
```

There are no significant partitions for touch.

Taste

Calculate the optimal decision tree, given variables related to taste:

(some variables are co-linear, so we take out sweet.additive and sour.additive)

```
s.taste = s[s$domain=='taste',]

rt.taste = REEMtree(simpson.diversityIndex~
  pulverise.spices+
  spices.herbs+
  num.additives.scaled +
  fragrant.food,
  random = ~1|Language,
  data = s.taste,
  MaxIterations=100000)
```

Taste Results

Proportion of variance explained:

```
# R squared
cor(predict.REEMtree(rt.taste,s.taste,id=s.taste$Language, EstimateRandomEffects = T),
  s.taste$simpson.diversityIndex)^2
```

```
## [1] 0.5397018
```

Plot the tree and calculate variable importance:

```
if(nrow(tree(rt.taste)$splits)>0){
  rpart.plot(tree(rt.taste), type=4,extra=100, branch.lty=1, box.palette="RdYlGn", main="Taste")
}

if(nrow(tree(rt.taste)$splits)>0){
  varimp = rt.taste$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)
}
```

No strong effect of number of additives in a full model:

```
mT.null = lmer(simpson.diversityIndex.log ~
  1 +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.taste)
mT.nA = lmer(simpson.diversityIndex.log ~
  1 + num.additives.scaled +
  (1 | Language) +
  (1 | Stimulus.code),
  data = s.taste)
anova(mT.null, mT.nA)
```

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

```
## Data: s.taste
## Models:
## mT.null: simpson.diversityIndex.log ~ 1 + (1 | Language) + (1 | Stimulus.code)
## mT.nA: simpson.diversityIndex.log ~ 1 + num.additives.scaled + (1 |
## mT.nA: Language) + (1 | Stimulus.code)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mT.null  4 219.36 229.36 -105.68  211.36
## mT.nA    5 220.67 233.17 -105.33  210.67 0.6901      1      0.4061
```

Specific taste hypotheses

Does the codability of a specific taste correlate with having an additive for that taste?

```
tx = s.taste[s.taste$Stimulus.code=="taste.citric acid monohydrate (sour)",]
t.test(tx$simpson.diversityIndex ~ tx$sour.additive)
```

```
##
## Welch Two Sample t-test
##
## data: tx$simpson.diversityIndex by tx$sour.additive
## t = -1.3496, df = 15.841, p-value = 0.1961
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.30612907 0.06808546
## sample estimates:
## mean in group no mean in group yes
##      0.3831169      0.5021387
```

```
tx = s.taste[s.taste$Stimulus.code=="taste.glutamate (umami)",]
t.test(tx$simpson.diversityIndex ~ tx$umami.additive)
```

```
##
## Welch Two Sample t-test
##
## data: tx$simpson.diversityIndex by tx$umami.additive
## t = -0.31073, df = 8.0573, p-value = 0.7639
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.4177380 0.3184188
## sample estimates:
## mean in group no mean in group yes
##      0.3132727      0.3629323
```

```
tx = s.taste[s.taste$Stimulus.code=="taste.quinine hydrochloride (bitter)",]
t.test(tx$simpson.diversityIndex ~ tx$bitter.additive)
```

```
##
## Welch Two Sample t-test
##
## data: tx$simpson.diversityIndex by tx$bitter.additive
## t = -0.12993, df = 10.88, p-value = 0.899
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.4251373 0.3778017
## sample estimates:
```

```

## mean in group no mean in group yes
##      0.5184787      0.5421466

tx = s.taste[s.taste$Stimulus.code=="taste.sodium chloride (salt)",]
t.test(tx$simpson.diversityIndex ~ tx$salt.additive)

##
## Welch Two Sample t-test
##
## data: tx$simpson.diversityIndex by tx$salt.additive
## t = -1.9499, df = 7.4425, p-value = 0.08972
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.45775223  0.04127946
## sample estimates:
## mean in group no mean in group yes
##      0.3787879      0.5870242

tx = s.taste[s.taste$Stimulus.code=="taste.sucrose (sweet)",]
t.test(tx$simpson.diversityIndex ~ tx$sweet.additive)

##
## Welch Two Sample t-test
##
## data: tx$simpson.diversityIndex by tx$sweet.additive
## t = -1.3499, df = 1.7914, p-value = 0.3224
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.110774  0.624763
## sample estimates:
## mean in group no mean in group yes
##      0.4053030      0.6483085

```

There are no specific taste effects.

Smell

```
s.smell = s[s$domain=='smell',]  
  
rt.smell = REEMtree(simpson.diversityIndex~  
  pulverise.spices+  
  spices.herbs+  
  fragrant.food +  
  subsistence,  
  random = ~1|Language,  
  data = s.smell,  
  MaxIterations=100000)
```

Smell Results

Proportion of variance explained:

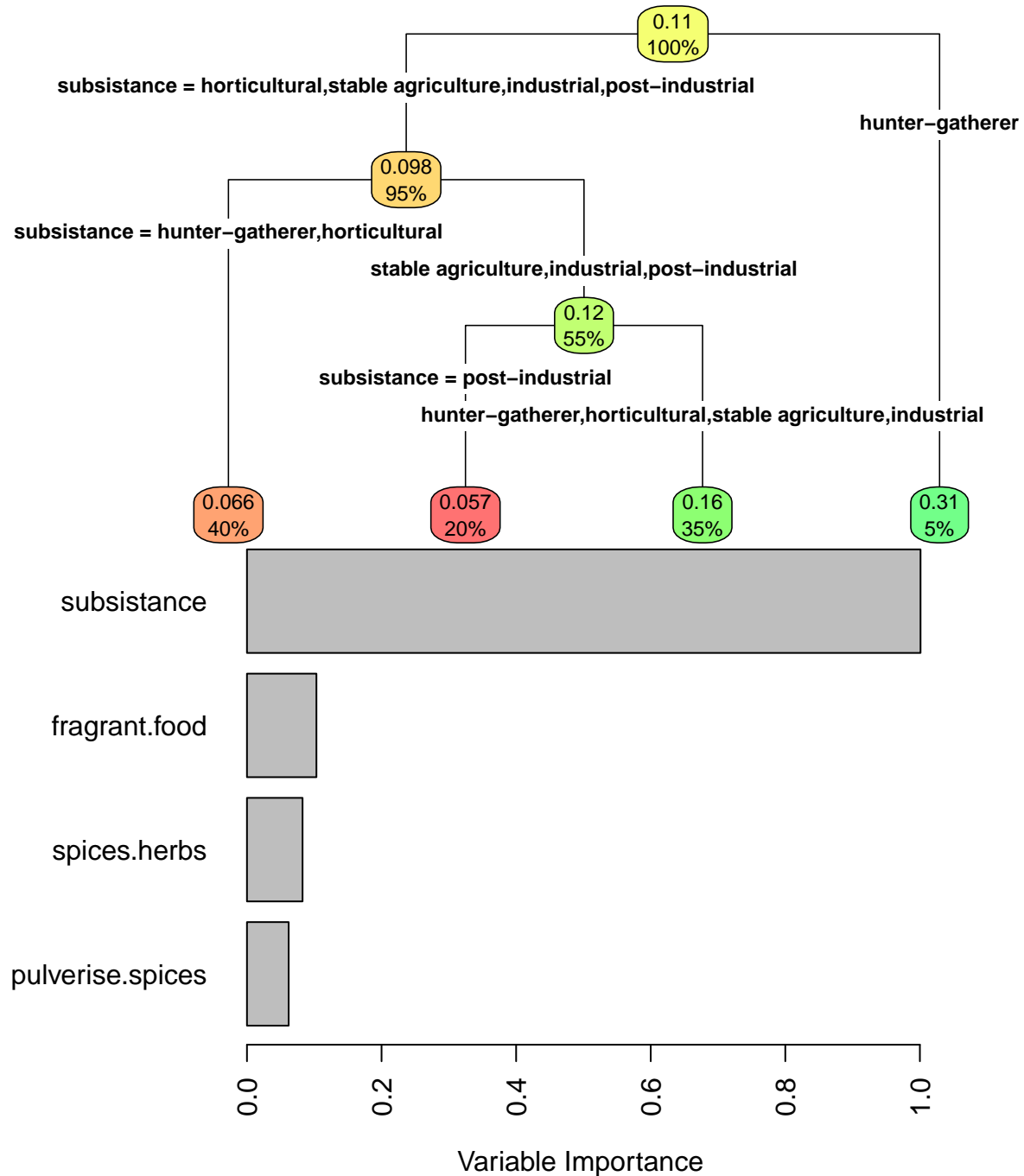
```
# R squared  
cor(predict.REEMtree(rt.smell,s.smell,id=s.smell$Language, EstimateRandomEffects = T),  
  s.smell$simpson.diversityIndex)^2
```

```
## [1] 0.3861878
```

Plot the tree and calculate variable importance:

```
if(nrow(tree(rt.smell)$cptable)>1){  
  rpart.plot(tree(rt.smell), type=4,extra=100, branch.lty=1, box.palette="RdYlGn", main="Smell")  
  
  varimp = rt.smell$Tree$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)  
}
```

Smell



Substance predicts codability. Particularly the hunter-gatherer language.

```
s.smell$substance = factor(s.smell$substance, levels=rev(levels(s.smell$substance)))
m0.smell = lmer(simpson.diversityIndex.log ~
  1 +
  (1 | Language) +
  (1 + substance | Stimulus.code),
```

```

data = s.smell)

m1.smell = update(m0.smell, ~. + subsistance)
anova(m0.smell,m1.smell)

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

## Data: s.smell
## Models:
## m0.smell: simpson.diversityIndex.log ~ 1 + (1 | Language) + (1 + subsistance |
## m0.smell:      Stimulus.code)
## m1.smell: simpson.diversityIndex.log ~ (1 | Language) + (1 + subsistance |
## m1.smell:      Stimulus.code) + subsistance
##           Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0.smell 18 432.62 495.27 -198.31  396.62
## m1.smell 22 416.96 493.53 -186.48  372.96 23.659      4 9.348e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

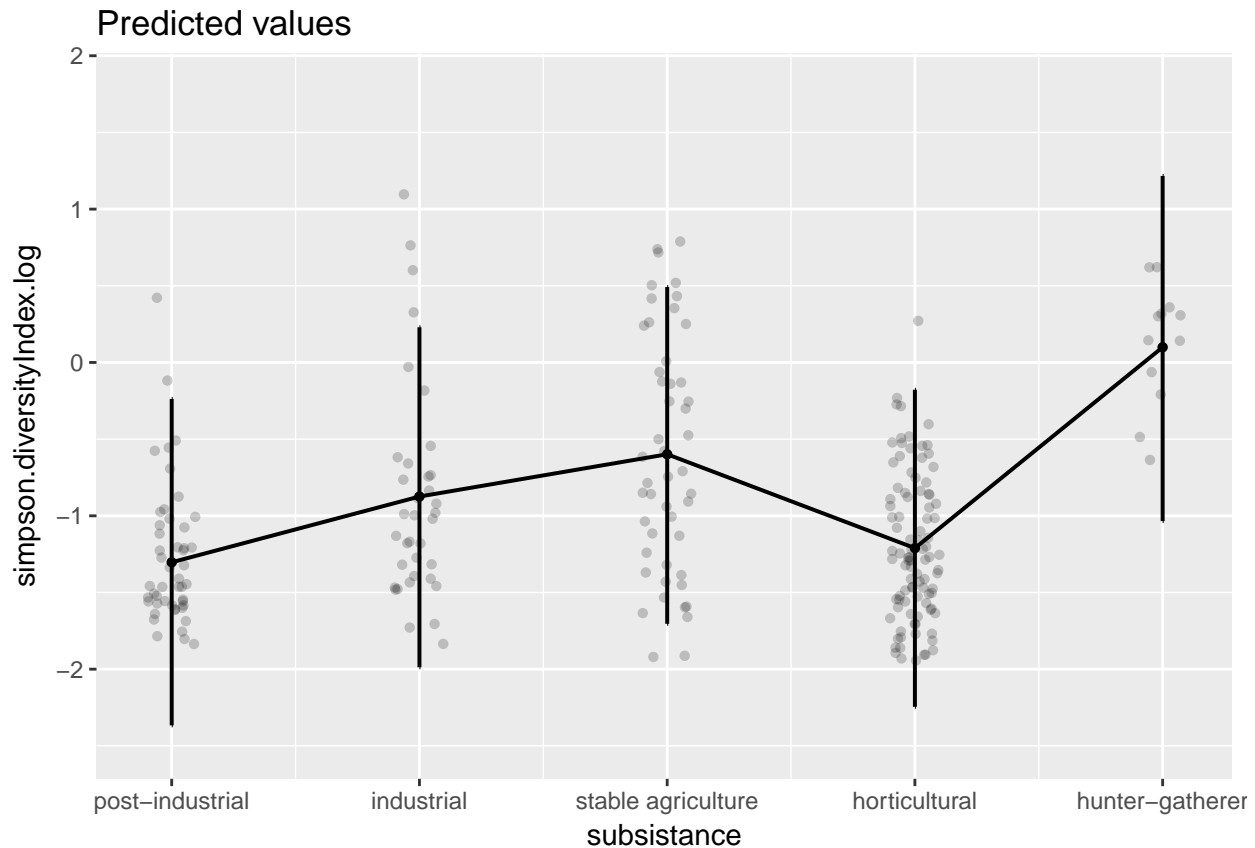
summary(m1.smell)

## Linear mixed model fit by REML ['lmerMod']
## Formula: simpson.diversityIndex.log ~ (1 | Language) + (1 + subsistance |
##      Stimulus.code) + subsistance
##      Data: s.smell
##
## REML criterion at convergence: 384.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.3109 -0.6086 -0.1843  0.5665  2.7232
##
## Random effects:
##      Groups      Name      Variance Std.Dev. Corr
##      Language      (Intercept)  0.04893  0.2212
##      Stimulus.code (Intercept)  0.05062  0.2250
##                  subsistance.L 0.02622  0.1619  -0.73
##                  subsistance.Q 0.01212  0.1101   0.20 -0.82
##                  subsistance.C 0.01019  0.1009   0.99 -0.81  0.32
##                  subsistance^4 0.07803  0.2793  -0.57  0.98 -0.92 -0.67
##      Residual                0.22559  0.4750
## Number of obs: 240, groups:  Language, 20; Stimulus.code, 12
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  -0.77520    0.09759  -7.943
## subsistance.L  0.78019    0.19787   3.943
## subsistance.Q  0.25163    0.17959   1.401
## subsistance.C  0.65474    0.14742   4.441
## subsistance^4  0.42002    0.15335   2.739
##
## Correlation of Fixed Effects:
##              (Intr) sbss.L sbss.Q sbss.C
## subsistnc.L  0.172
## subsistnc.Q  0.342  0.483

```

```
## subsistnc.C 0.478 0.330 0.246
## subsistnc^4 -0.101 0.318 0.019 -0.174
```

```
sjp.lmer(m1.smell, 'pred', "subsistence", show.ci = T)
```

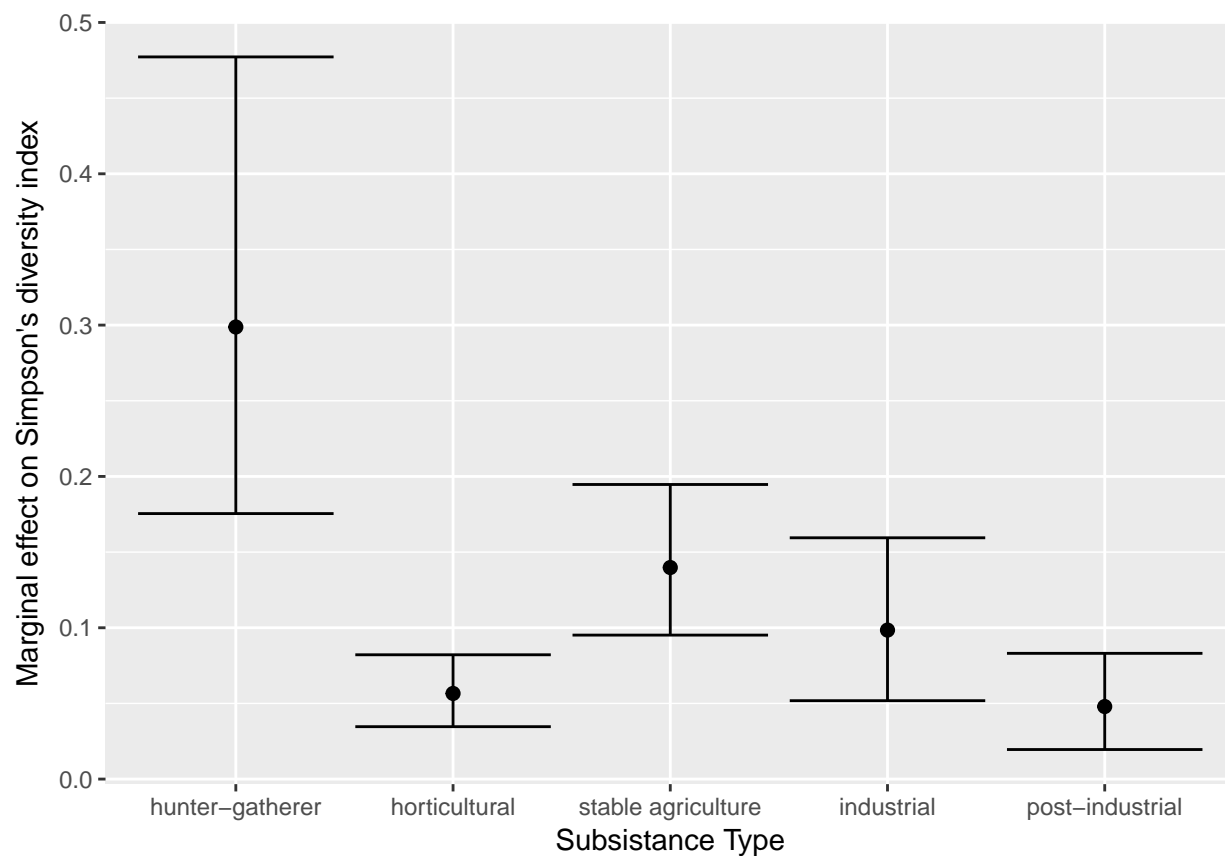


```
gx = sjp.lmer(m1.smell, 'eff', show.ci = T, prnt.plot = F)
subsd = gx$plot$data
subsd$`Subsistence Type` = factor(subsd$label,
  levels = c("hunter-gatherer", "horticultural",
    'stable agriculture', 'industrial',
    "post-industrial"), ordered = T)

convertSimps = function(X){
  exp(X * attr(s$simpson.diversityIndex.log, 'scaled:scale') +
    attr(s$simpson.diversityIndex.log, 'scaled:center')) - 0.1
}

subsd$y = convertSimps(subsd$y)
subsd$lower = convertSimps(subsd$lower)
subsd$upper = convertSimps(subsd$upper)

ggplot(subsd,
  aes(`Subsistence Type`, y)) +
  geom_point(size=2) +
  geom_errorbar(aes(ymin = lower, ymax = upper)) +
  ylab("Marginal effect on Simpson's diversity index")
```

Raw data

```
gx.smell = ggplot(s[s$domain=="smell",],
  aes(substance=="hunter-gatherer", simpson.diversityIndex)) +
  geom_boxplot() +
  xlab("Substance Type") +
  ylab("Codability") +
  scale_x_discrete(labels=c("Other", "Hunter-gatherer")) +
  coord_cartesian(ylim=c(0,1))
```

M

```
tapply(s.smell$simpson.diversityIndex,s.smell$subsistence,mean)
```

```
##      post-industrial      industrial stable agriculture
##      0.05675500      0.12930588      0.17779720
##      horticultural      hunter-gatherer
##      0.06604194      0.31296296
```

```
tapply(s.smell$simpson.diversityIndex,s.smell$subsistence=='hunter-gatherer',mean)
```

```
##      FALSE      TRUE
## 0.09760326 0.31296296
```

Relationship with latitude:

```
m0.smell = lmer(simpson.diversityIndex.log ~
  1 +
  (1 | Language) +
  (1 + I(abs(latitude))) | Stimulus.code),
  data = s.smell)
```

```

m1.smell = update(m0.smell, ~. +I(abs(latitude)))
m2.smell = update(m1.smell, ~. +I(abs(latitude)^2))

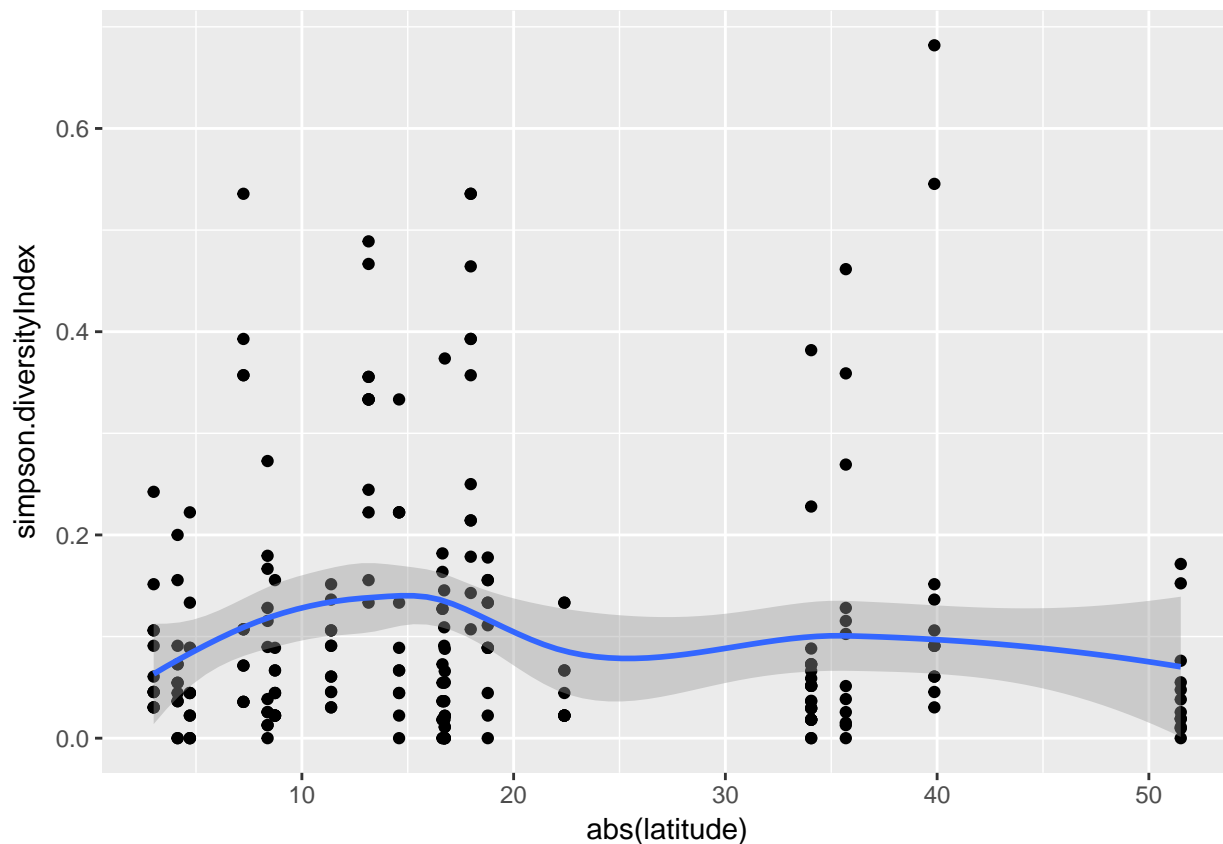
anova(m0.smell, m1.smell, m2.smell)

## refitting model(s) with ML (instead of REML)
## Data: s.smell
## Models:
## m0.smell: simpson.diversityIndex.log ~ 1 + (1 | Language) + (1 + I(abs(latitude)) |
## m0.smell: Stimulus.code)
## m1.smell: simpson.diversityIndex.log ~ (1 | Language) + (1 + I(abs(latitude)) |
## m1.smell: Stimulus.code) + I(abs(latitude))
## m2.smell: simpson.diversityIndex.log ~ (1 | Language) + (1 + I(abs(latitude)) |
## m2.smell: Stimulus.code) + I(abs(latitude)) + I(abs(latitude)^2)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0.smell 6 408.97 429.85 -198.48   396.97
## m1.smell 7 410.83 435.20 -198.42   396.83 0.1329     1    0.7155
## m2.smell 8 412.20 440.05 -198.10   396.20 0.6325     1    0.4264

ggplot(s.smell, aes(abs(latitude), simpson.diversityIndex)) +
  geom_point() + geom_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```



No significant relationship with latitude.

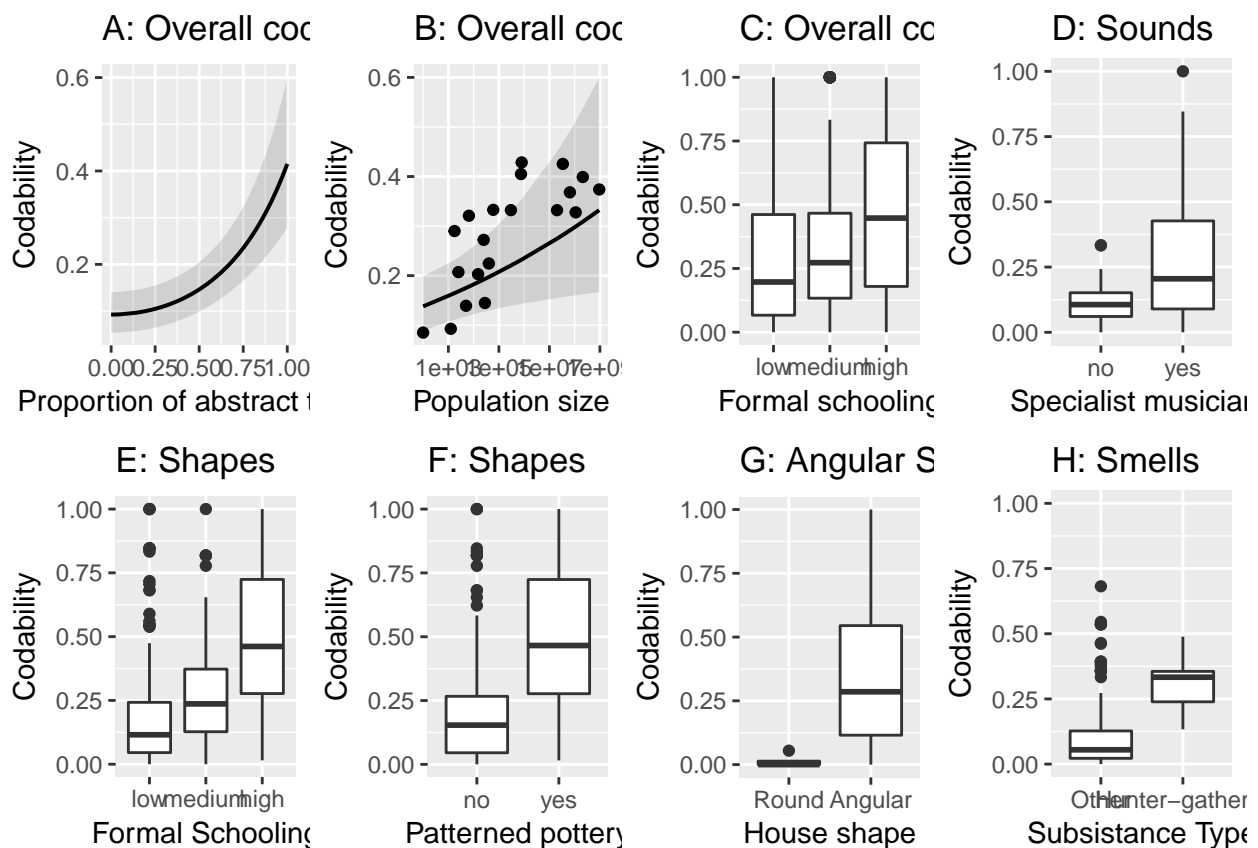
Extra Graphs

```
pdf("../results/graphs/Diversity_by_population.pdf", width=6, height=6)
gx.pop
dev.off()
```

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

```
load("../results/graphs/Codability_by_AbstractUse_ggplot.RDat")
gx.abstractTerms = px
gx.abstractTerms = gx.abstractTerms+ theme(strip.text = element_blank())
gx.abstractTerms$theme$plot.title = gx.pop$theme$plot.title

gx2b = grid.arrange(
  gx.abstractTerms + ggtitle("A: Overall codability"),
  gx.pop + ggtitle("B: Overall codability"),
  gg.formalschooling + ggtitle("C: Overall codability"),
  gx.sound + ggtitle("D: Sounds"),
  gx.shape2 + ggtitle("E: Shapes"),
  gx.shape + ggtitle("F: Shapes"),
  gx.shape.house + ggtitle("G: Angular Shapes"),
  gx.smell + ggtitle("H: Smells"),
  layout_matrix = rbind(c(1,2,3,4),c(5,6,7,8)))
```



```
pdf("../results/graphs/Diversity_plots.pdf", width=10, height=6)
plot(gx2b)
```

```

dev.off()

## pdf
## 2

setEPS()
postscript("../results/graphs/Diversity_plots.eps", width=10, height=6)
gx2b

## TableGrob (2 x 4) "arrange": 8 grobs
##   z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (1-1,3-3) arrange gtable[layout]
## 4 4 (1-1,4-4) arrange gtable[layout]
## 5 5 (2-2,1-1) arrange gtable[layout]
## 6 6 (2-2,2-2) arrange gtable[layout]
## 7 7 (2-2,3-3) arrange gtable[layout]
## 8 8 (2-2,4-4) arrange gtable[layout]

dev.off()

## pdf
## 2

```