

The interactive origin of iconicity: Mixed effects models

Contents

Introduction	2
Spikiness ratings	2
Load libraries	2
Load data	2
Mixed effects model	3
Mixed effects model with binarised spikiness ratings	9
Binary tree analysis	14
Iconicity of innovations	15
Load data	15
Distribution of iconicity in innovations	16
Increase in iconicity by survival	23
Accuracy	27
Mixed effects model	27
Iconicity and accuracy for innovations	29
Iconicity and accuracy for whole data	31
Iconicity and systematicity	33
Mixed effects model	34
Results	35

Introduction

This file contains an analysis of the spikiness ratings of the final output languages and the accuracy of guessing during the experiments. The spikiness ratings are not bimodally distributed, so the analysis of spikiness ratings is done using both the continuous spikiness rating values and a binarised version of the ratings.

Note that in the main text, we refer to the two conditions as “communication” and “reproduction”, while the data is coded as “communication” and “learning”.

Spikiness ratings

Load libraries

```
library(gplots)
library(lattice)
library(ggplot2)
library(lme4)
library(party)
library(sjPlot)
library(lawstat)
```

Load data

```
finalLangs = read.csv("../data/finalLanguages/FinalLanguages.csv", stringsAsFactors = F)
# convert labels to English
finalLangs$Shape[finalLangs$Shape=="Picudo"] = "Spiky"
finalLangs$Shape[finalLangs$Shape=="Redondo"] = "Round"

# load all trial data
alldatx = read.csv("../results/AllTrialData.csv", stringsAsFactors = F)
```

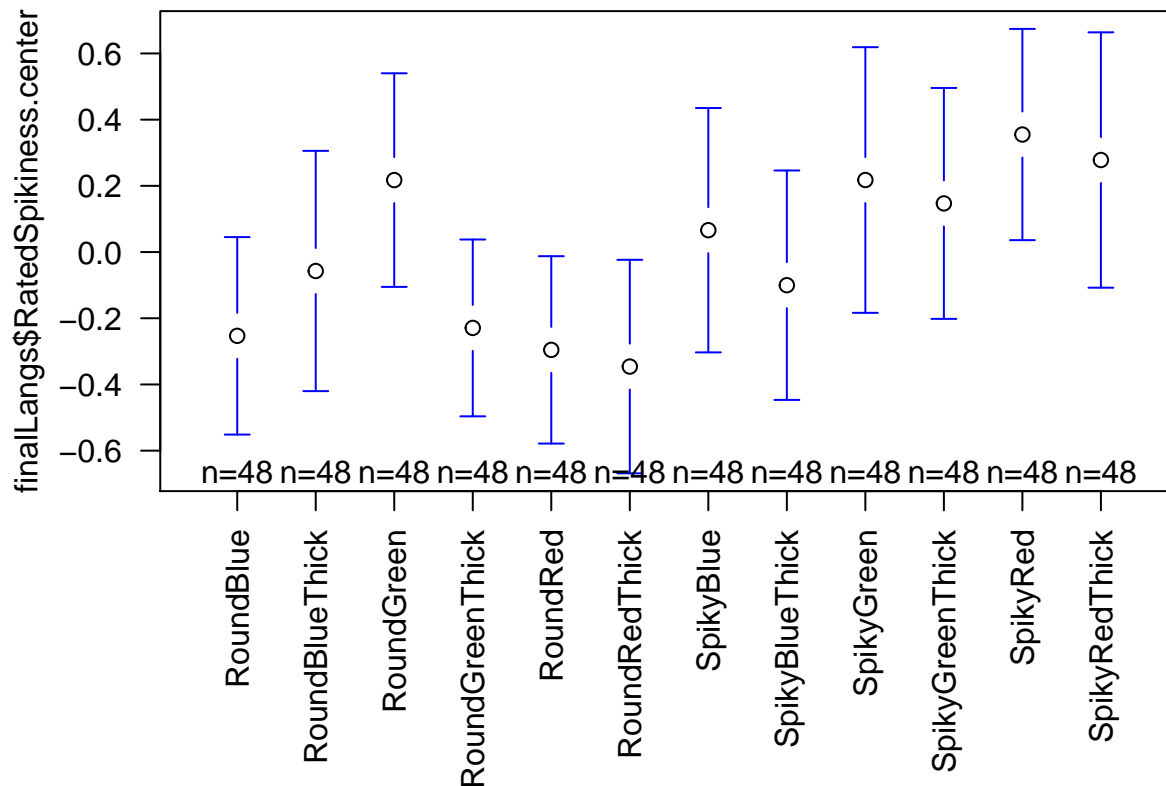
Center spikiness ratings and re-level factors.

```
finalLangs$RatedSpikiness.center =
  finalLangs$RatedSpikiness - mean(finalLangs$RatedSpikiness)

finalLangs$Cond = factor(finalLangs$Cond, levels=c("Learn", "Communication"))
finalLangs$Shape = factor(finalLangs$Shape, levels=c("Round", "Spiky"))
```

Plot the data by item (all conditions, all generations)

```
par(mar=c(8,4,2,2))
plotmeans(finalLangs$RatedSpikiness.center~finalLangs$Item, las=2, xlab="", connect=F)
```



There are differences between items

Mixed effects model

Build a series of models with random effects for Chain and Item.

```
# null model
m0 = lmer(RatedSpikiness.center ~ 1 + (1 |Chain) + (1|Item), data=finalLangs)
# + condition
m1 = lmer(RatedSpikiness.center ~ Cond + (1 |Chain) + (1|Item), data=finalLangs)
# + generation
m2 = lmer(RatedSpikiness.center ~ Cond + Gen + (1 |Chain) + (1|Item), data=finalLangs)
# + shape
m3 = lmer(RatedSpikiness.center ~ Cond + Gen + Shape + (1 |Chain)
          + (1|Item), data=finalLangs)
# + interaction between shape and generation
m4 = lmer(RatedSpikiness.center ~ Cond + (Gen * Shape) + (1 |Chain)
          + (1|Item), data=finalLangs)
# + interaction between condition and generation
m5 = lmer(RatedSpikiness.center ~ (Cond*Gen) + (Gen * Shape) + (1 |Chain)
          + (1|Item), data=finalLangs)
# + interaction between shape and condition
m6 = lmer(RatedSpikiness.center ~ (Cond*Gen) + (Gen * Shape) + (Shape:Cond)
          + (1 |Chain) + (1|Item), data=finalLangs)
# + 3-way interaction
m7 = lmer(RatedSpikiness.center ~ Cond * Gen * Shape + (1 |Chain)
          + (1|Item), data=finalLangs)
```

Results

Look inside main model

```
summary(m7)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: RatedSpikiness.center ~ Cond * Gen * Shape + (1 | Chain) + (1 |
##      Item)
##      Data: finallangs
##
## REML criterion at convergence: 1767.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.8411 -0.8370 -0.1665  0.7906  2.3066
##
## Random effects:
##      Groups      Name                Variance Std.Dev.
##      Item       (Intercept)  0.01058   0.1029
##      Chain      (Intercept)  0.18043   0.4248
##      Residual                    1.17881   1.0857
## Number of obs: 576, groups:  Item, 12; Chain, 8
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      0.022530   0.299064   0.075
## CondCommunication  0.096751   0.418750   0.231
## Gen              -0.033860   0.052978  -0.639
## ShapeSpiky       -0.003530   0.297764  -0.012
## CondCommunication:Gen -0.064573   0.074923  -0.862
## CondCommunication:ShapeSpiky -0.032181   0.412642  -0.078
## Gen:ShapeSpiky     0.002764   0.074923   0.037
## CondCommunication:Gen:ShapeSpiky 0.189234   0.105957   1.786
##
## Correlation of Fixed Effects:
##              (Intr) CndCmm Gen      ShpSpk CndC:G CnC:SS Gn:ShS
## CondCmmnctn -0.700
## Gen          -0.620  0.443
## ShapeSpiky   -0.498  0.341  0.623
## CndCmmnct:G  0.438 -0.626 -0.707 -0.440
## CndCmmnc:SS  0.345 -0.493 -0.449 -0.693  0.635
## Gen:ShpSpky  0.438 -0.313 -0.707 -0.881  0.500  0.635
## CndCmm:G:SS -0.310  0.443  0.500  0.623 -0.707 -0.899 -0.707
```

Test the differences between model fits.

```
anova(m0,m1,m2,m3,m4,m5,m6,m7)
```

```
## refitting model(s) with ML (instead of REML)
## Data: finallangs
## Models:
## m0: RatedSpikiness.center ~ 1 + (1 | Chain) + (1 | Item)
## m1: RatedSpikiness.center ~ Cond + (1 | Chain) + (1 | Item)
## m2: RatedSpikiness.center ~ Cond + Gen + (1 | Chain) + (1 | Item)
```

```
## m3: RatedSpikiness.center ~ Cond + Gen + Shape + (1 | Chain) + (1 |
## m3: Item)
## m4: RatedSpikiness.center ~ Cond + (Gen * Shape) + (1 | Chain) +
## m4: (1 | Item)
## m5: RatedSpikiness.center ~ (Cond * Gen) + (Gen * Shape) + (1 | Chain) +
## m5: (1 | Item)
## m6: RatedSpikiness.center ~ (Cond * Gen) + (Gen * Shape) + (Shape:Cond) +
## m6: (1 | Chain) + (1 | Item)
## m7: RatedSpikiness.center ~ Cond * Gen * Shape + (1 | Chain) + (1 |
## m7: Item)
##      Df      AIC      BIC logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0  4 1779.7 1797.1 -885.83  1771.7
## m1  5 1781.2 1803.0 -885.61  1771.2  0.4475      1 0.5035471
## m2  6 1782.8 1808.9 -885.40  1770.8  0.4234      1 0.5152634
## m3  7 1777.7 1808.2 -881.87  1763.7  7.0627      1 0.0078704 **
## m4  8 1776.4 1811.3 -880.21  1760.4  3.3049      1 0.0690737 .
## m5  9 1778.1 1817.3 -880.05  1760.1  0.3156      1 0.5742584
## m6 10 1768.1 1811.6 -874.04  1748.1 12.0326      1 0.0005228 ***
## m7 11 1766.9 1814.8 -872.43  1744.9  3.2087      1 0.0732495 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There was a significant main effect of shape ($\beta = -0.0035$, $\text{std.err} = 0.3$, Wald $t = -0.012$; log likelihood difference = 3.5 , $df = 1$, Chi Squared = 7.06 , $p = 0.0079$).

There was a significant interaction between shape and condition ($\beta = -0.032$, $\text{std.err} = 0.41$, Wald $t = -0.078$; log likelihood difference = 6 , $df = 1$, Chi Squared = 12.03 , $p = 0.00052$).

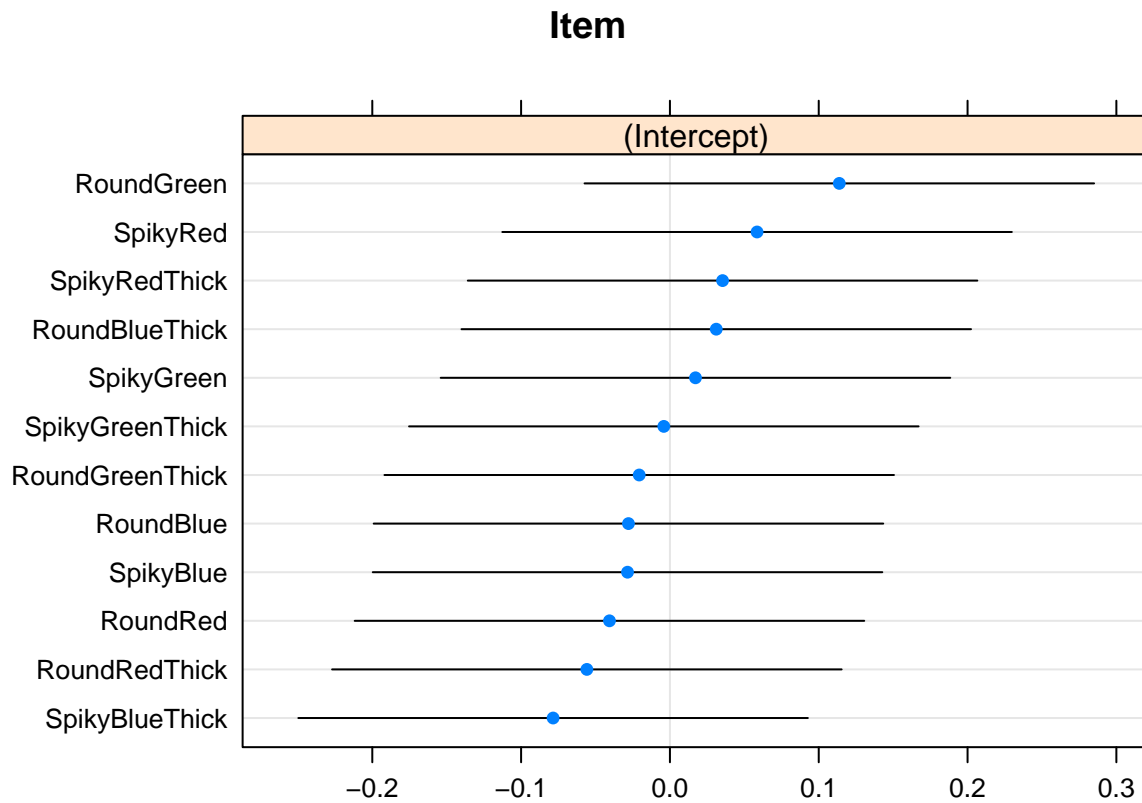
There was a marginal interaction between shape and generation ($\beta = 0.0028$, $\text{std.err} = 0.075$, Wald $t = 0.037$; log likelihood difference = 1.7 , $df = 1$, Chi Squared = 3.3 , $p = 0.069$).

There was a marginal three-way interaction between shape, condition and generation ($\beta = 0.19$, $\text{std.err} = 0.11$, Wald $t = 1.8$; log likelihood difference = 1.6 , $df = 1$, Chi Squared = 3.21 , $p = 0.073$).

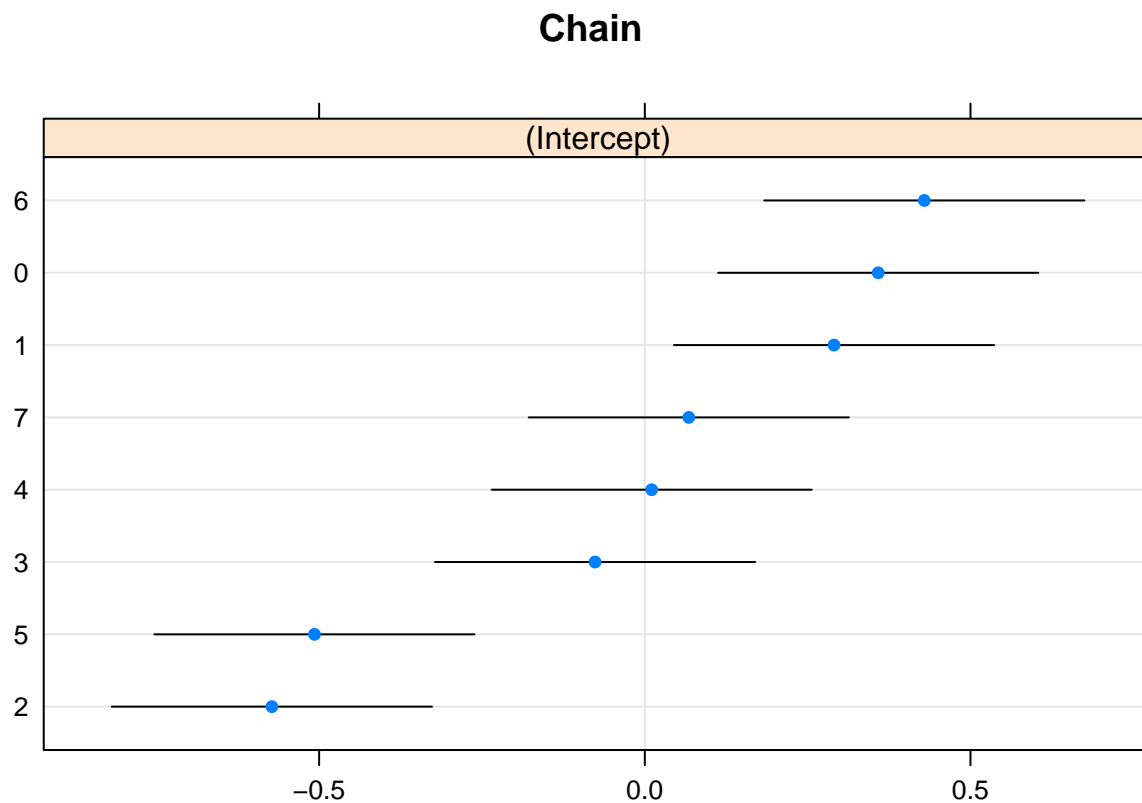
Plot the random effects.

```
dotplot(ranef(m7, condVar=T))
```

```
## $Item
```



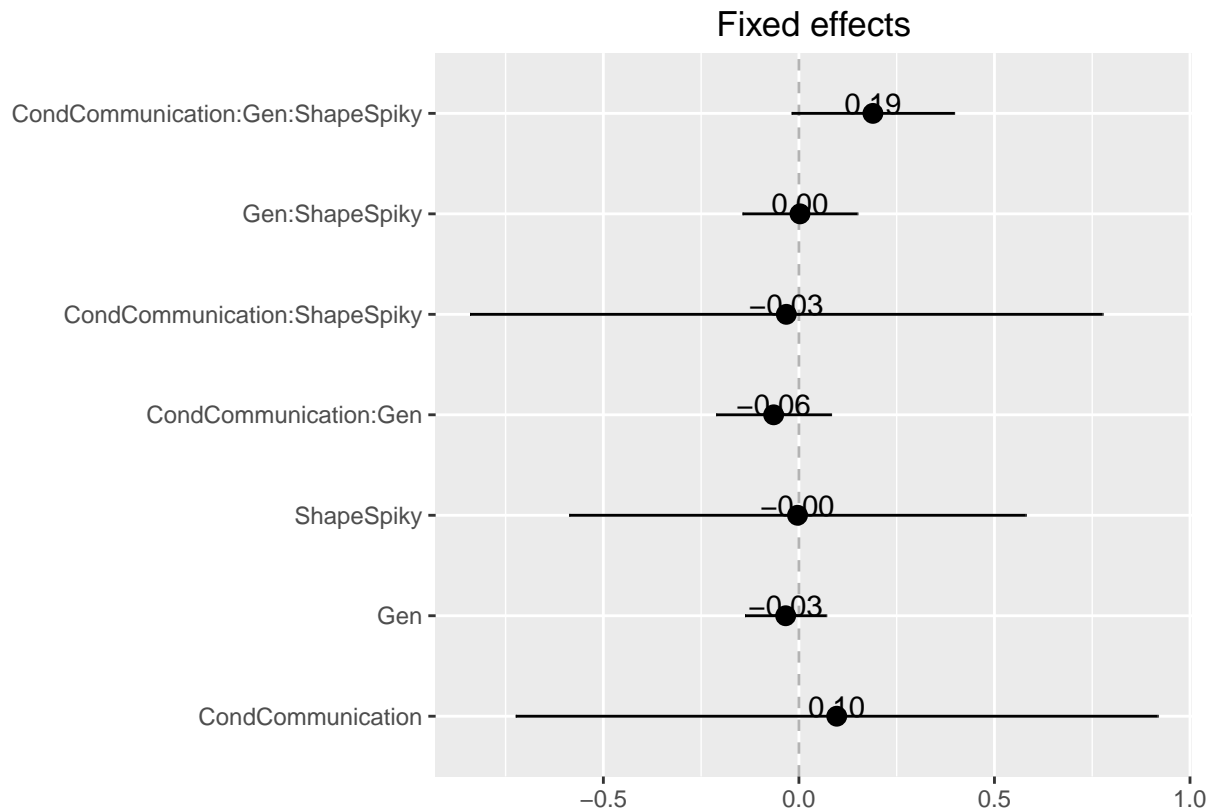
\$Chain



Plot the fixed effects with error estimates from the final model. The 3-way interaction between condition,

generation and shape is marginally significant:

```
sjp.lmer(m7, type='fe', geom.colors=c(1,1))
```



Visualise the three-way interaction. The first part of this script creates some new data and uses the model to predict values.

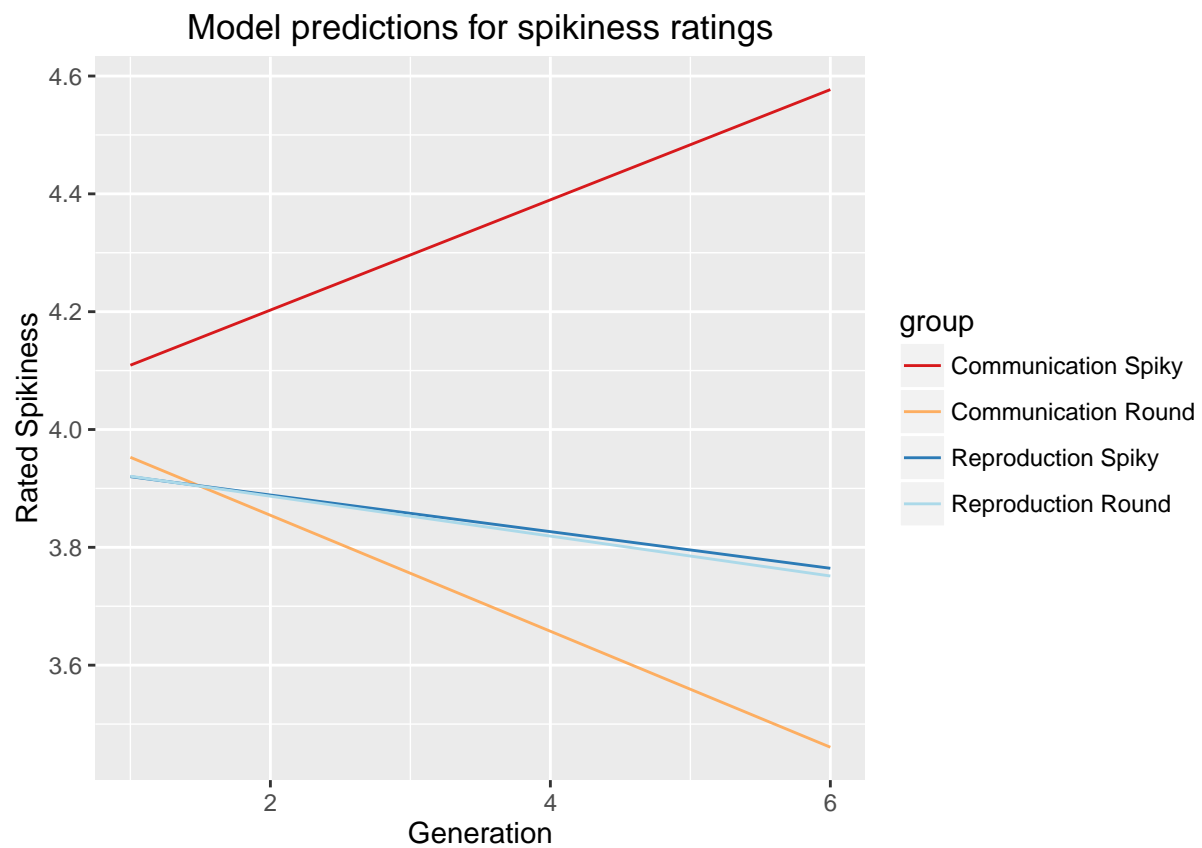
The interaction is driven by Spiky and Round shapes diverging over generations in the communication condition, but not in the reproduction condition.

```
predictedData = data.frame(
  RatedSpikiness.center = NA,
  Gen=rep(1:6,4),
  Shape= rep(rep(c("Spiky","Round"),each=6),2),
  Cond = rep(c("Communication","Learn"),each=12)
)
predictedData$RatedSpikiness.center = predict(m7,
  newdata=predictedData, re.form=NA)

#Re-scale
predictedData$RatedSpikiness =
  predictedData$RatedSpikiness.center + mean(finallangs$RatedSpikiness)

# Make groups and set order for plotting
predictedData$group = paste(predictedData$Cond, predictedData$Shape)
predictedData$group = gsub("Learn","Reproduction", predictedData$group)
predictedData$group = factor(
  predictedData$group,
  levels = c("Communication Spiky","Communication Round",
    "Reproduction Spiky", "Reproduction Round"))
```

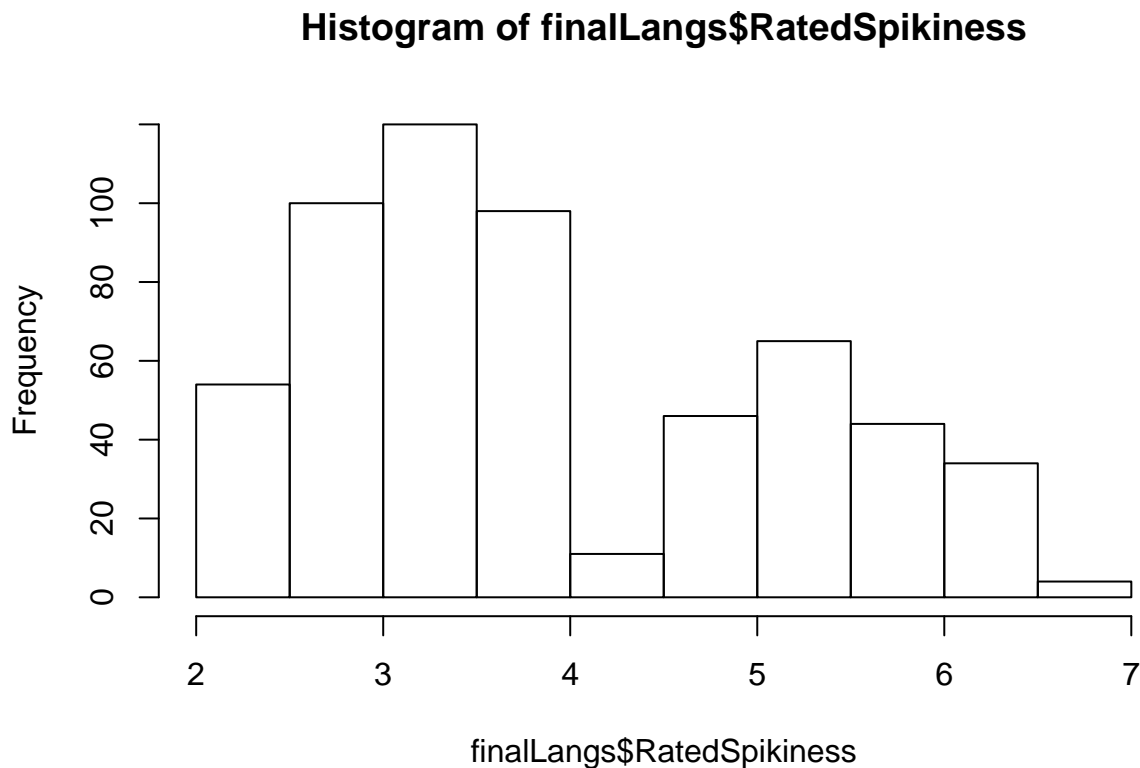
```
# Plot the predictions
qplot(Gen, RatedSpikiness,
      data=predictedData,
      color=group,
      geom=c("line")) +
scale_color_manual(values=c("#d7191c", "#fdae61", "#2c7bb6", "#abd9e9")) +
xlab("Generation") +
ylab("Rated Spikiness") +
ggtitle("Model predictions for spikiness ratings")
```



Mixed effects model with binarised spikiness ratings

The spikiness ratings are not normally distributed:

```
hist(finalLangs$RatedSpikiness)
```



So we binarise the variable into spiky/not spiky:

```
finalLangs$RatedSpikiness.bin = finalLangs$RatedSpikiness > 4
```

Run a series of models. Note that intermediate models 5 and 6 do not converge, but the final model 7 does.

```
mcontrol = glmerControl(optCtrl = list(maxfun = 500000))

mb0 = glmer(RatedSpikiness.bin ~ 1 + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
mb1 = glmer(RatedSpikiness.bin ~ Cond + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
mb2 = glmer(RatedSpikiness.bin ~ Cond + Gen + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
mb3 = glmer(RatedSpikiness.bin ~ Cond + Gen + Shape + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
mb4 = glmer(RatedSpikiness.bin ~ Cond + (Gen * Shape) + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
mb5 = glmer(RatedSpikiness.bin ~ (Cond*Gen) + (Gen * Shape) + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00112016 (tol =
## 0.001, component 1)
```

```
mb6 = glmer(RatedSpikiness.bin ~ (Cond*Gen) + (Gen * Shape) + (Shape:Cond) + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00254262 (tol =
## 0.001, component 1)

mb7 = glmer(RatedSpikiness.bin ~ Cond * Gen * Shape + (1 |Chain) + (1|Item),
            data=finalLangs, family=binomial, control = mcontrol)
```

Results

Look inside main model

```
summary(mb7)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: RatedSpikiness.bin ~ Cond * Gen * Shape + (1 | Chain) + (1 |
## Item)
## Data: finalLangs
## Control: mcontrol
##
##      AIC      BIC   logLik deviance df.resid
##    722.9    766.4   -351.4    702.9     566
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4001 -0.7152 -0.4951  0.9714  2.5752
##
## Random effects:
##  Groups Name            Variance Std.Dev.
##  Item   (Intercept) 0.06298  0.2510
##  Chain  (Intercept) 0.30153  0.5491
## Number of obs: 576, groups: Item, 12; Chain, 8
##
## Fixed effects:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.80967    0.50900  -1.591    0.112
## CondCommunication    0.11711    0.72308   0.162    0.871
## Gen                0.06152    0.10567   0.582    0.560
## ShapeSpiky         0.52479    0.60063   0.874    0.382
## CondCommunication:Gen -0.25227    0.16195  -1.558    0.119
## CondCommunication:ShapeSpiky -0.06135    0.83301  -0.074    0.941
## Gen:ShapeSpiky      -0.16967    0.15042  -1.128    0.259
## CondCommunication:Gen:ShapeSpiky 0.39112    0.21894   1.786    0.074 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) CndCmm Gen      ShpSpk CndC:G CnC:SS Gn:ShS
## CondCmmnctn -0.675
## Gen         -0.736  0.518
```

```
## ShapeSpiky -0.600 0.398 0.623
## CndCmmnct:G 0.480 -0.751 -0.653 -0.407
## CndCmmnc:SS 0.407 -0.617 -0.449 -0.679 0.652
## Gen:ShpSpky 0.518 -0.364 -0.703 -0.871 0.459 0.628
## CndCmm:G:SS -0.356 0.556 0.483 0.599 -0.740 -0.893 -0.687
```

Test model comparison:

```
anova(mb0,mb1,mb2,mb3,mb4,mb5,mb6,mb7)
```

```
## Data: finalLangs
## Models:
## mb0: RatedSpikiness.bin ~ 1 + (1 | Chain) + (1 | Item)
## mb1: RatedSpikiness.bin ~ Cond + (1 | Chain) + (1 | Item)
## mb2: RatedSpikiness.bin ~ Cond + Gen + (1 | Chain) + (1 | Item)
## mb3: RatedSpikiness.bin ~ Cond + Gen + Shape + (1 | Chain) + (1 |
## mb3: Item)
## mb4: RatedSpikiness.bin ~ Cond + (Gen * Shape) + (1 | Chain) + (1 |
## mb4: Item)
## mb5: RatedSpikiness.bin ~ (Cond * Gen) + (Gen * Shape) + (1 | Chain) +
## mb5: (1 | Item)
## mb6: RatedSpikiness.bin ~ (Cond * Gen) + (Gen * Shape) + (Shape:Cond) +
## mb6: (1 | Chain) + (1 | Item)
## mb7: RatedSpikiness.bin ~ Cond * Gen * Shape + (1 | Chain) + (1 |
## mb7: Item)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mb0  3 729.66 742.72 -361.83 723.66
## mb1  4 731.64 749.07 -361.82 723.64 0.0130      1 0.9092167
## mb2  5 733.09 754.87 -361.54 723.09 0.5560      1 0.4558874
## mb3  6 730.23 756.37 -359.12 718.23 4.8538      1 0.0275855 *
## mb4  7 732.22 762.71 -359.11 718.22 0.0115      1 0.9147795
## mb5  8 734.12 768.97 -359.06 718.12 0.1001      1 0.7517608
## mb6  9 724.09 763.29 -353.04 706.09 12.0352      1 0.0005221 ***
## mb7 10 722.88 766.44 -351.44 702.88 3.2044      1 0.0734423 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There was a significant main effect of shape ($\beta = 0.52$, $\text{std.err} = 0.6$, $\text{Wald } t = 0.87$, $\text{Wald } p = 0.38$; $\log \text{likelihood difference} = 2.4$, $\text{df} = 1$, $\text{Chi Squared} = 4.85$, $p = 0.028$).

There was a significant interaction between shape and condition ($\beta = -0.061$, $\text{std.err} = 0.83$, $\text{Wald } t = -0.074$, $\text{Wald } p = 0.94$; $\log \text{likelihood difference} = 6$, $\text{df} = 1$, $\text{Chi Squared} = 12.04$, $p = 0.00052$).

There was no significant interaction between shape and generation ($\beta = -0.17$, $\text{std.err} = 0.15$, $\text{Wald } t = -1.1$, $\text{Wald } p = 0.26$; $\log \text{likelihood difference} = 0.0057$, $\text{df} = 1$, $\text{Chi Squared} = 0.01$, $p = 0.91$).

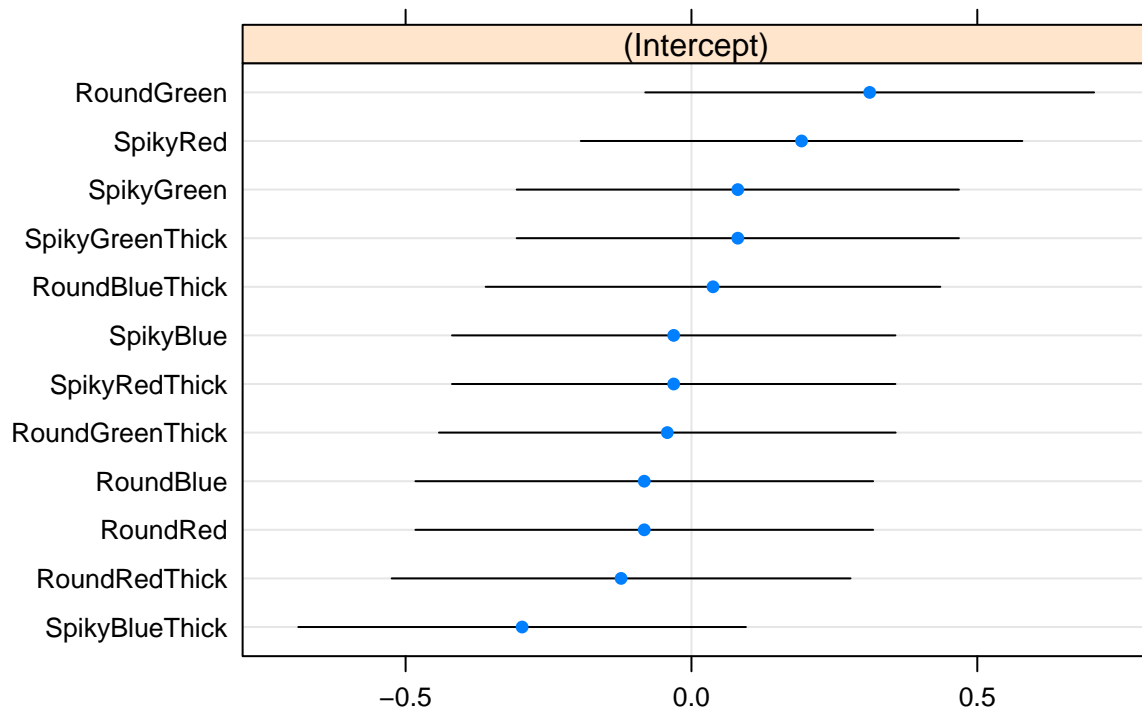
There was a marginal three-way interaction between shape, condition and generation ($\beta = 0.39$, $\text{std.err} = 0.22$, $\text{Wald } t = 1.8$, $\text{Wald } p = 0.074$; $\log \text{likelihood difference} = 1.6$, $\text{df} = 1$, $\text{Chi Squared} = 3.2$, $p = 0.073$).

Plot random effects of final model

```
dotplot(ranef(mb7, condVar=T))
```

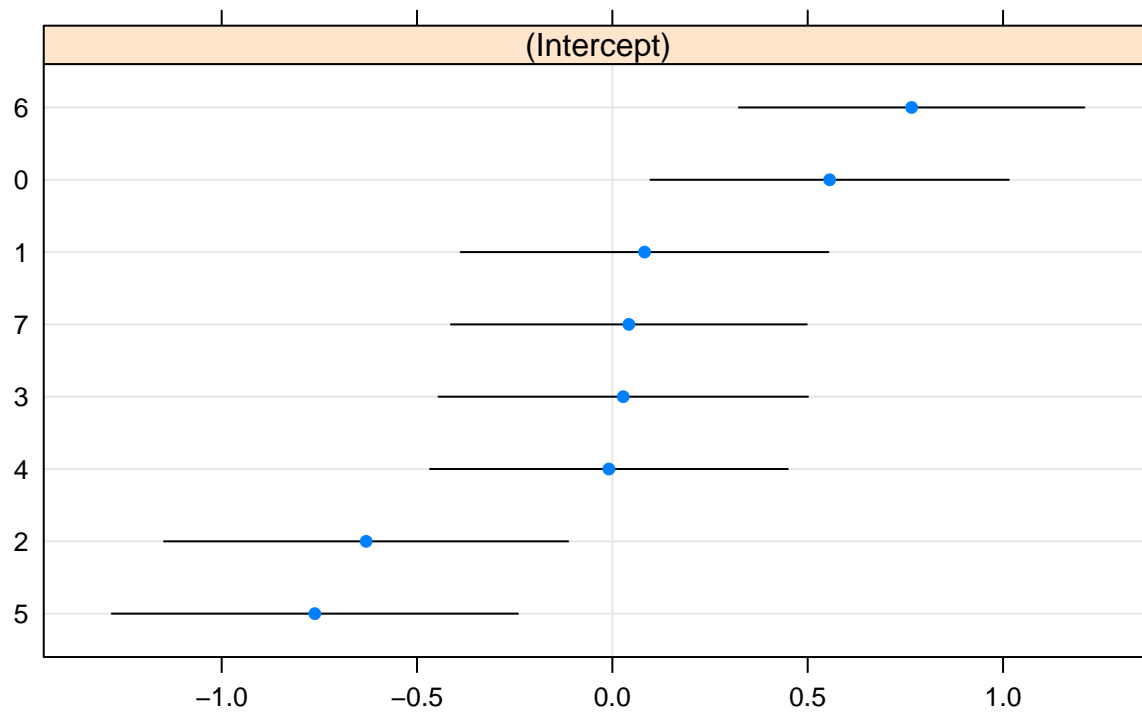
```
## $Item
```

Item



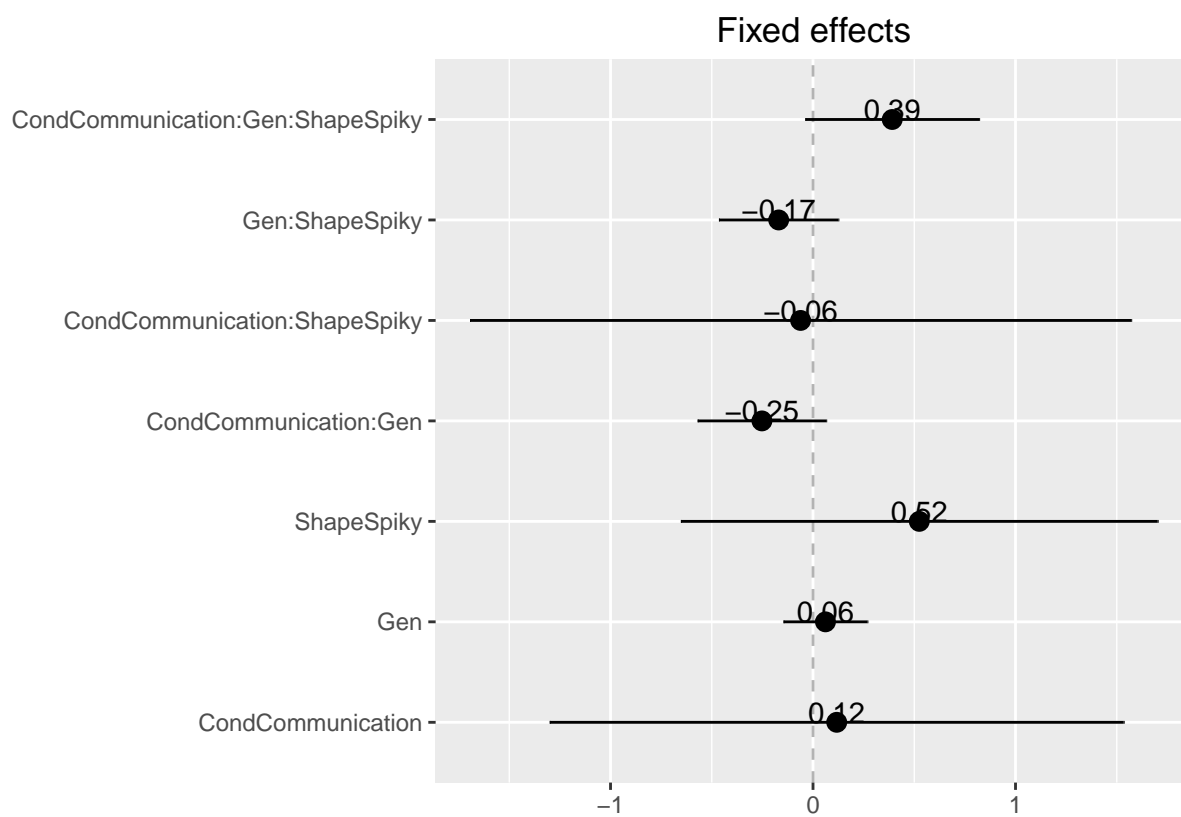
\$Chain

Chain



Plot fixed effects with standard error from final model.

```
sjp.lmer(mb7, type='fe', geom.colors=c(1,1))
```



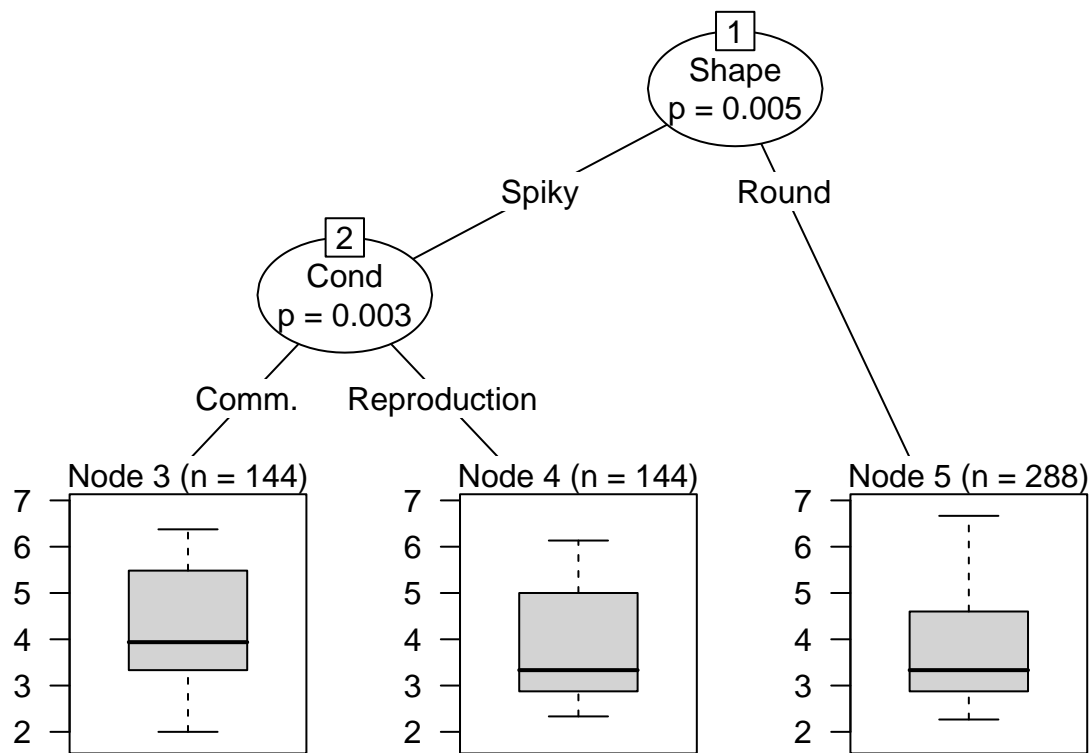
Binary tree analysis

We use a binary decision tree to predict spikiness ratings by condition, generation, item shape, item colour and item border type.

The results agree with those above, namely that the main effects are for shape, but spiky meanings are rated as more spiky in the communication condition

```
finalLangs2 = finalLangs
finalLangs2$Shape = factor(finalLangs2$Shape)
finalLangs2$Colour = factor(finalLangs2$Colour)
finalLangs2$Border = factor(finalLangs2$Border)
finalLangs2$Cond = factor(finalLangs2$Cond, labels = c("Reproduction", "Comm. "))

cx = ctree(RatedSpikiness~Cond+Gen+Shape+Colour+Border, data=finalLangs2)
plot(cx)
```



Iconicity of innovations

Load data

Note that the column *Human* in the data indicates whether the signal was sent by a human. This is always the case in the communication condition, but only true for half of the trials in the reproduction condition. In the reproduction condition, when *Human* is FALSE, the human participant is guessing meaning from the signal sent by the program.

```
datax = read.csv("../results/IncreaseInIconicity.csv", stringsAsFactors = F)
alldatax = read.csv("../results/AllTrialData.csv", stringsAsFactors = F)
```

Number of innovations in each conditon, by whether the innovation was an unseen word (versus a change in mapping):

```
tx = table(datax[datax$Human,]$innovation.mutation,
            datax[datax$Human,]$condition)
```

```
tx
```

```
##
##           Comm Learn
##   FALSE   511   206
##   TRUE    178    76
```

```
tx/rep(colSums(tx),each=2)
```

```
##
##           Comm      Learn
##   FALSE 0.7416546 0.7304965
##   TRUE  0.2583454 0.2695035
```

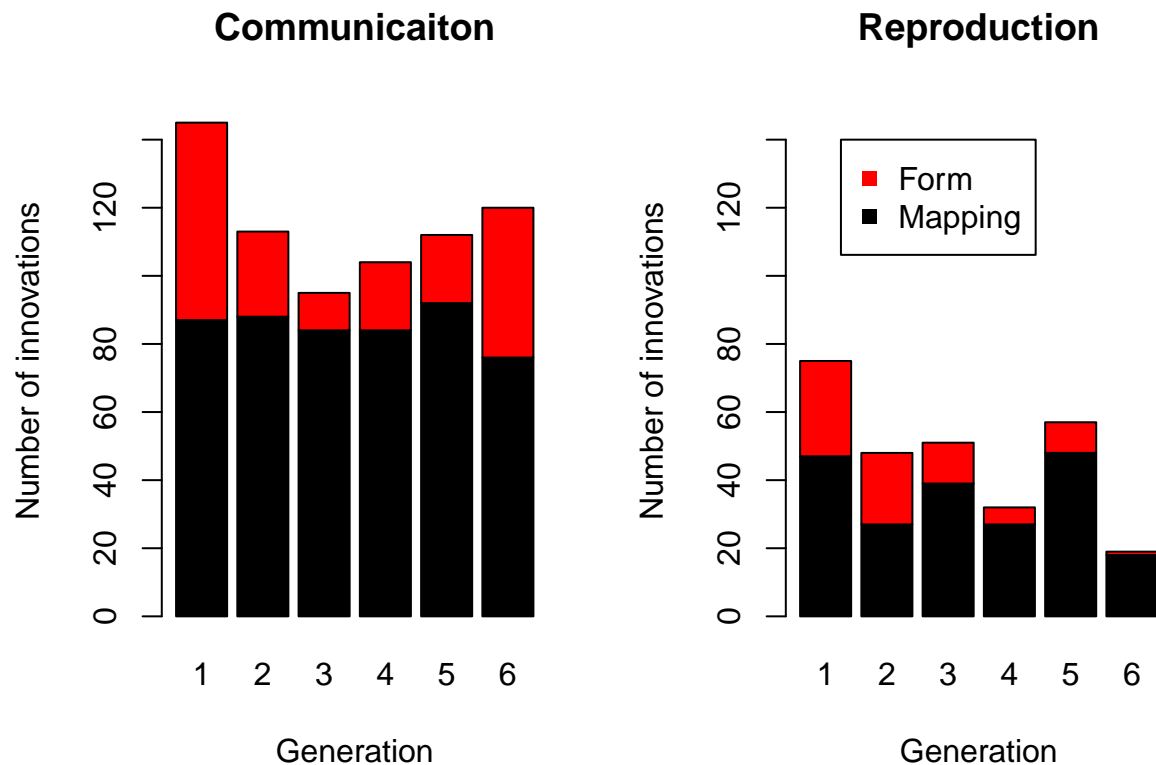
The number of innovations in each condition by generation. The number of innovations declines in the learning condition:

```
par(mfrow=c(1,2))
barplot((table(
  datax[datax$condition=="Comm",]$innovation.mutation,
  datax[datax$condition=="Comm",]$gen)), beside = F,
  ylab="Number of innovations"
  ,ylim=c(0,150),
  col=1:2,
  main="Communicaiton",
  xlab="Generation")
```

```
lx = datax$condition=="Learn" & datax$Human
```

```
barplot((table(
  datax[lx,]$innovation.mutation,
  datax[lx,]$gen)), beside = F,
  ylab="Number of innovations"
  ,ylim=c(0,150),
  col=1:2,
  main="Reproduction",
  xlab="Generation")
```

```
legend(1,140,legend=c("Form","Mapping"),
  col=2:1, pch=15)
```



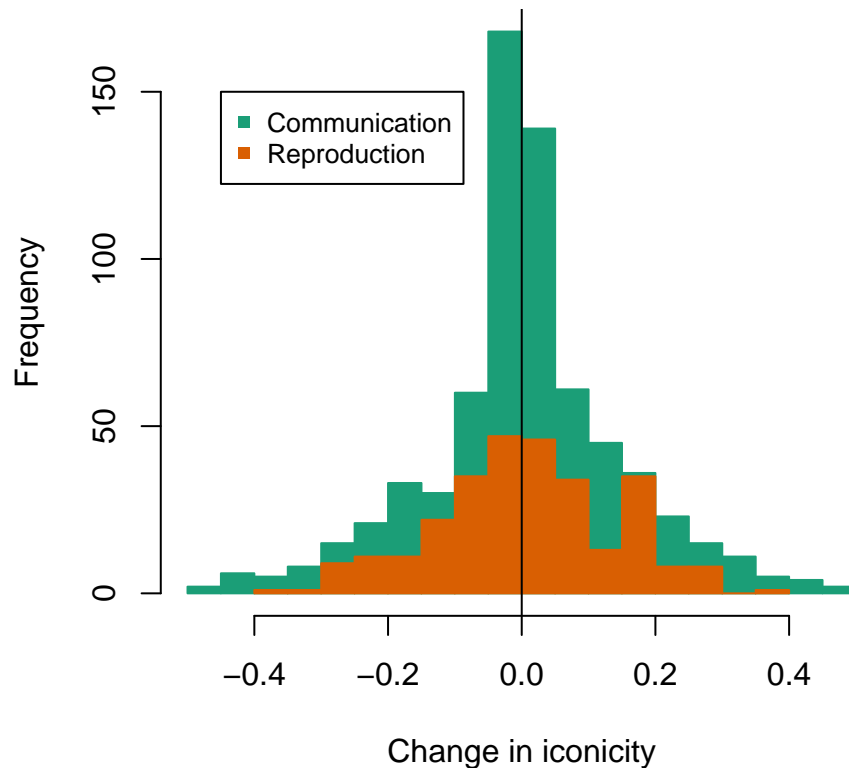
Distribution of iconicity in innovations

Below is the distribution of how the innovation increases the iconicity of the mapping compared to the word that it replaced. The increases are small compared to the full Likert scale (1-7), and centered around zero.

```
# innovations produced by participants in the communication condition
comm.innovation.iconicity.dist =
  datax[datax$condition=='Comm' & datax$Human,]$increaseIconicity
# innovations produced by the human in the reproduction condition
learn.innovation.iconicity.dist =
  datax[datax$condition=='Learn' & datax$Human,]$increaseIconicity

cols = c('#1b9e77', '#d95f02')

hist(comm.innovation.iconicity.dist, col=cols[1],
      breaks=14,
      border = cols[1],
      main='',
      xlab="Change in iconicity")
hist(learn.innovation.iconicity.dist,
      add=T,
      col=cols[2],
      breaks=14,
      border = cols[2])
abline(v=0)
legend(-0.45,150, legend = c("Communication", "Reproduction"), col=cols, pch=15, cex=0.8)
```

Are the distributions biased? The curve for the learning condition looks like it has a bump on the right. Test the symmetry with a Wilcox signed rank test and the MGG test (Miao, Gel & Gastwirth, 2006, see `symmetry.test` function in `lawstat` package).

```
wilcox.test(comm.innovation.iconicity.dist)
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: comm.innovation.iconicity.dist
## V = 117360, p-value = 0.3562
## alternative hypothesis: true location is not equal to 0
```

```
wilcox.test(learn.innovation.iconicity.dist)
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: learn.innovation.iconicity.dist
## V = 20314, p-value = 0.6351
## alternative hypothesis: true location is not equal to 0
```

```
symmetry.test(comm.innovation.iconicity.dist)
```

```
##
## m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth
## (2006)
##
## data: comm.innovation.iconicity.dist
## Test statistic = 1.1233, p-value = 0.328
## alternative hypothesis: the distribution is asymmetric.
```

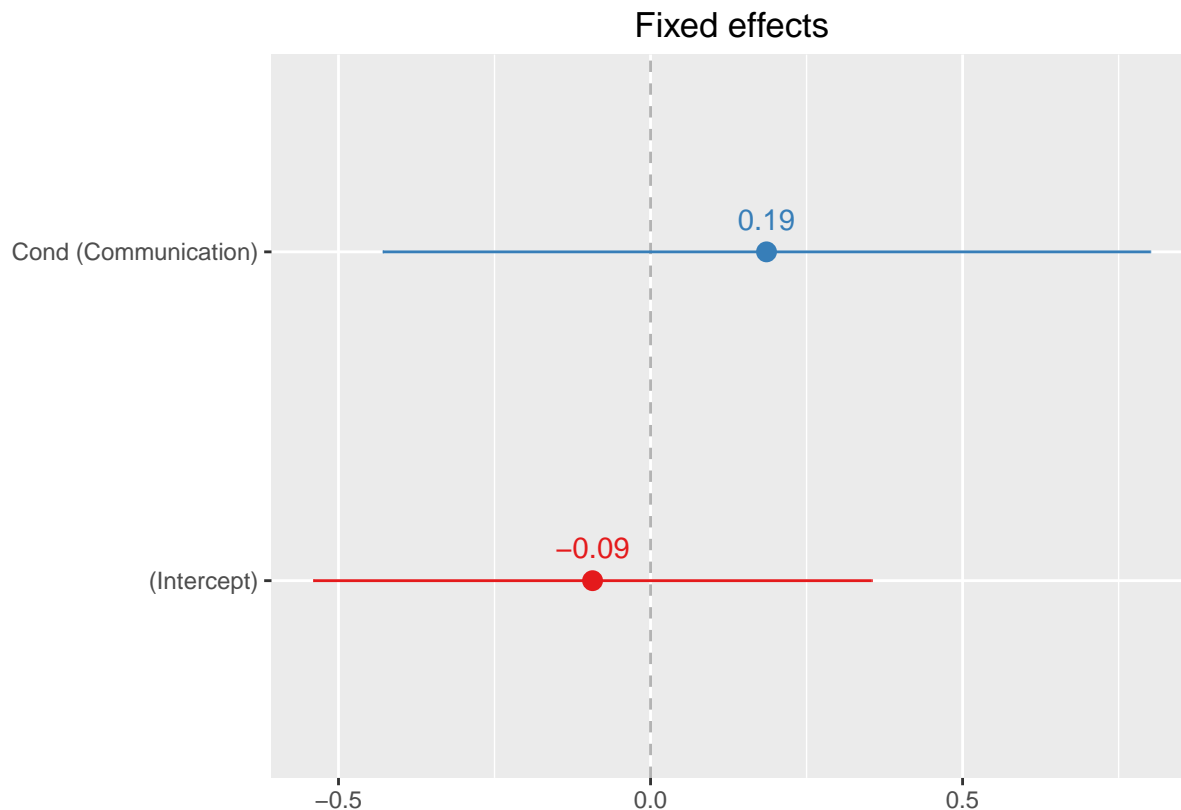
```
## sample estimates:
## bootstrap optimal m
##          118
symmetry.test(learn.innovation.iconicity.dist)

##
## m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth
## (2006)
##
## data: learn.innovation.iconicity.dist
## Test statistic = 0.31437, p-value = 0.698
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##          54
```

The tests show that the distributions are not different from zero and are not asymmetric. In other words, innovations are randomly distributed. A mixed effects model also shows that the intercept is not significantly different from zero:

```
m0 = lmer(increaseIconicity ~ condition*inFinalLang + (1|chain) + (1|gen) + (1|meaning),
          data=datax[datax$Human,])
sjp.lmer(m1, 'fe', show.intercept=T)
```

```
## Computing p-values via Kenward-Roger approximation. Use `p.kr = FALSE` if computation takes too long
## Warning in deviance.merMod(object, ...): deviance() is deprecated for REML
## fits; use REMLcrit for the REML criterion or deviance(.,REML=FALSE) for
## deviance calculated at the REML fit
```

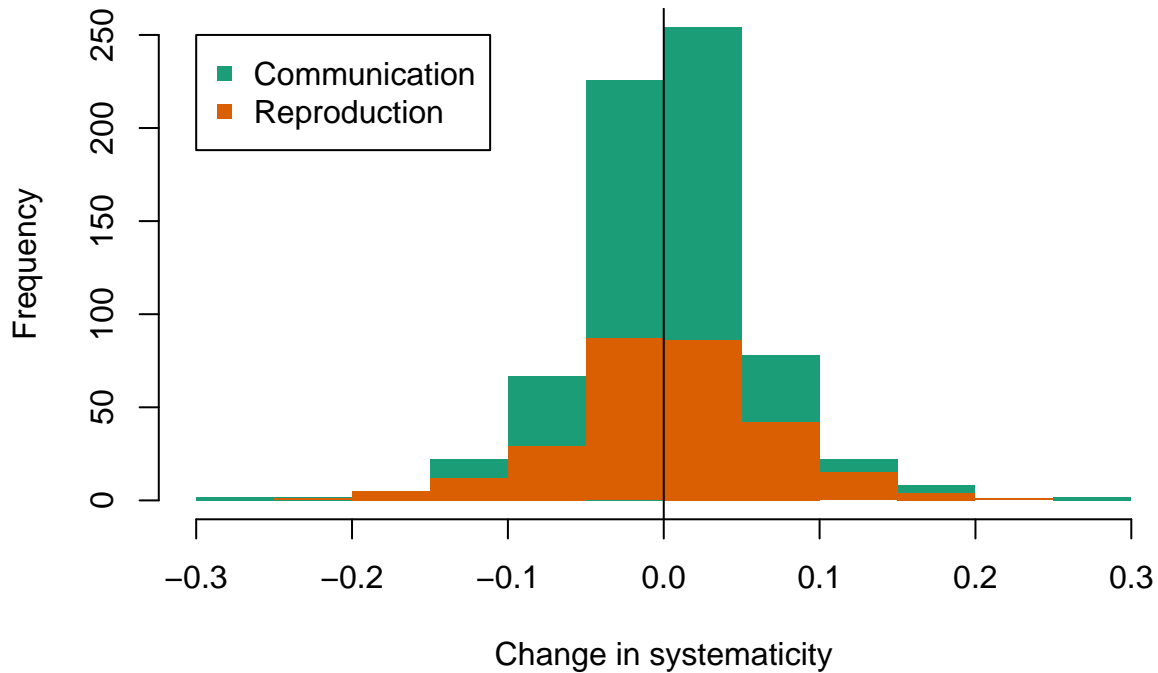


Interestingly, the results look similar for the change in systematicity.

```
# innovations produced by participants in the communication condition
comm.innovation.sys.dist =
  datax[datax$condition=='Comm' & datax$Human,]$systematicity.increase
# innovations produced by the human in the reproduction condition
learn.innovation.sys.dist =
  datax[datax$condition=='Learn' & datax$Human,]$systematicity.increase

cols = c('#1b9e77', '#d95f02')

hist(comm.innovation.sys.dist, col=cols[1],
      breaks=14,
      border = NA,
      main='',
      xlab="Change in systematicity")
hist(learn.innovation.sys.dist,
      add=T,
      col=cols[2],
      breaks=14,
      border = NA)
abline(v=0)
legend(-0.3,250, legend = c("Communication", "Reproduction"), col=cols, pch=15)
```



```
wilcox.test(comm.innovation.sys.dist)

##
## Wilcoxon signed rank test with continuity correction
##
## data:  comm.innovation.sys.dist
## V = 125380, p-value = 0.1655
## alternative hypothesis: true location is not equal to 0
```

```
wilcox.test(learn.innovation.sys.dist)
```

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: learn.innovation.sys.dist  
## V = 21640, p-value = 0.1178  
## alternative hypothesis: true location is not equal to 0
```

```
symmetry.test(comm.innovation.sys.dist)
```

```
##  
## m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth  
## (2006)  
##  
## data: comm.innovation.sys.dist  
## Test statistic = 0.22877, p-value = 0.864  
## alternative hypothesis: the distribution is asymmetric.  
## sample estimates:  
## bootstrap optimal m  
## 430
```

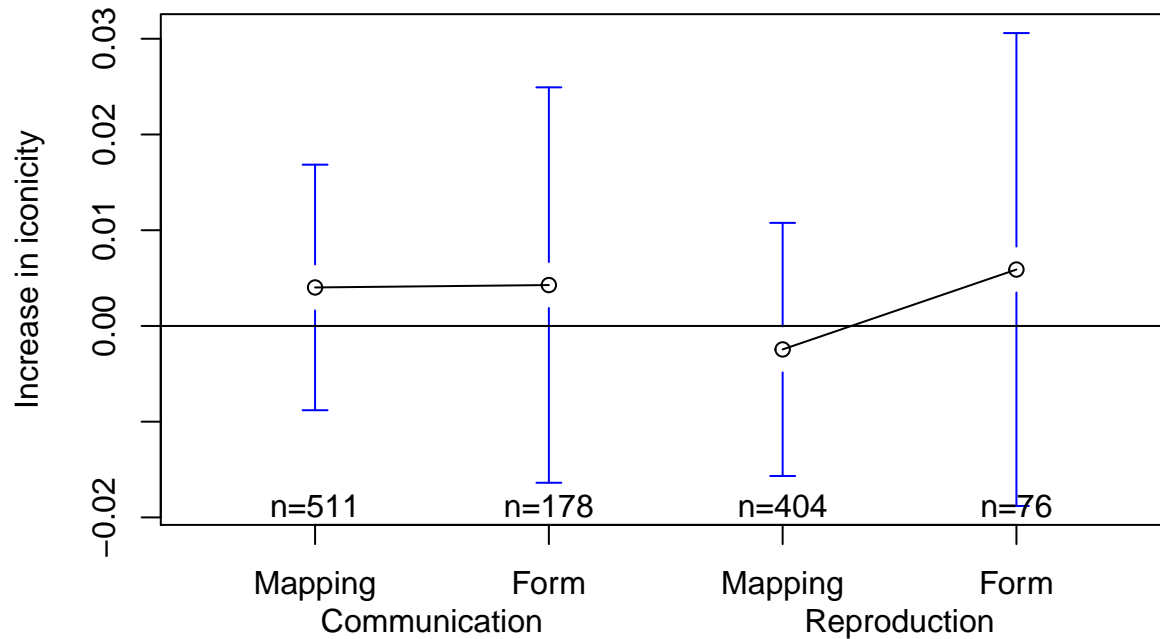
```
symmetry.test(learn.innovation.sys.dist)
```

```
##  
## m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth  
## (2006)  
##  
## data: learn.innovation.sys.dist  
## Test statistic = 0.99137, p-value = 0.324  
## alternative hypothesis: the distribution is asymmetric.  
## sample estimates:  
## bootstrap optimal m  
## 176
```

Innovations in form versus innovations in mapping

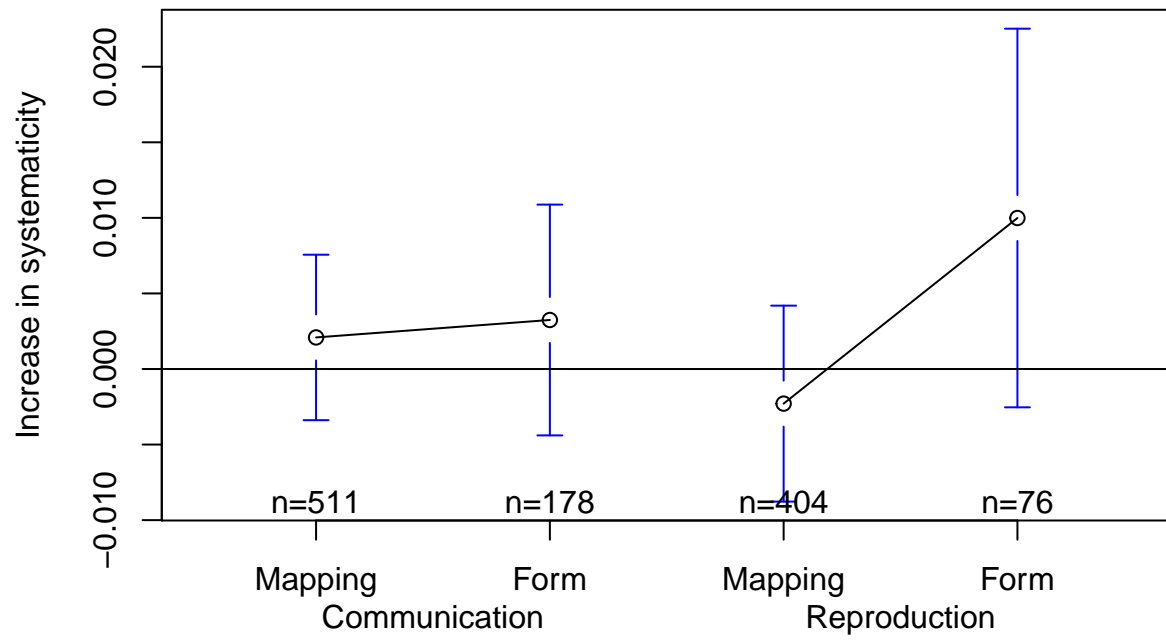
There appear to be no differences with regards to iconicity according to the category of innovation:

```
plotmeans(increaseIconicity ~ paste(condition,innovation.mutation),
  data=datax,
  legends = c("Mapping","Form","Mapping",'Form'),
  xlab='',
  ylab='Increase in iconicity',
  connect = list(1:2,3:4)
)
axis(1,at=c(1.5,3.5),c("Communication","Reproduction"),line=1, tick=F)
abline(h=0)
```



However, there seems like a weak bias for form innovations in the learning condition to lead to an increase in systematicity. Future work could explore these implications.

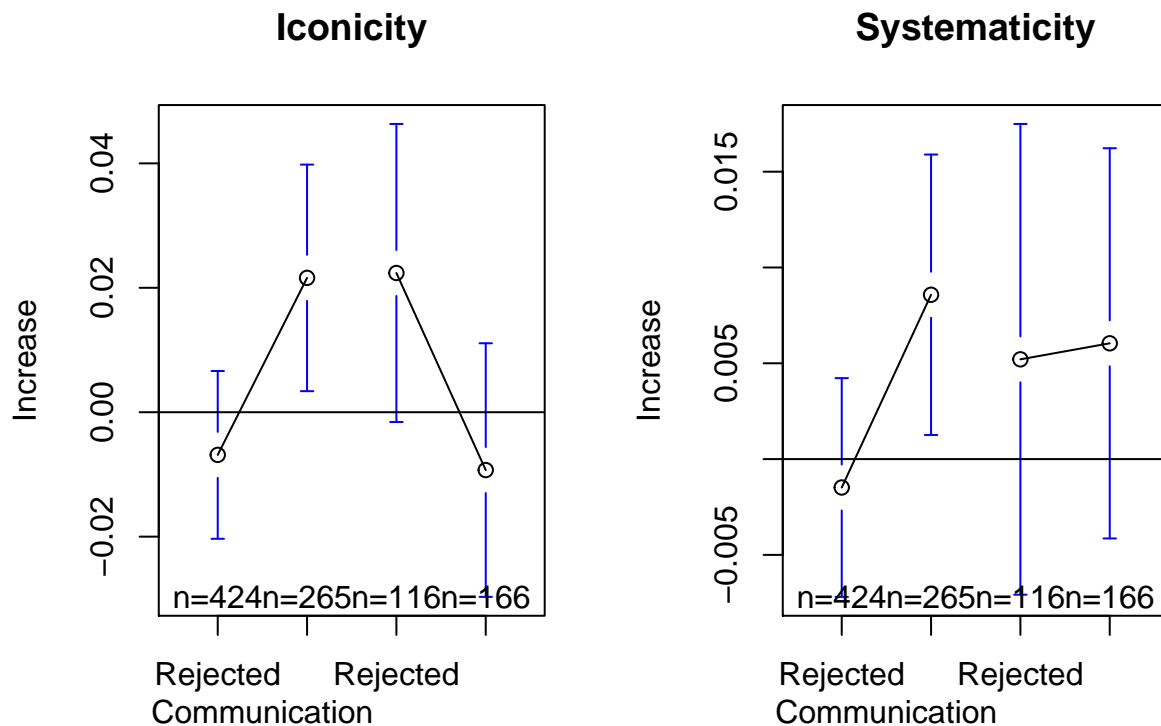
```
plotmeans(systematicity.increase ~ paste(condition,innovation.mutation),
  data=datax,
  legends = c("Mapping","Form","Mapping",'Form'),
  xlab='',
  ylab='Increase in systematicity',
  connect = list(1:2,3:4)
)
axis(1,at=c(1.5,3.5),c("Communication","Reproduction"),line=1, tick=F)
abline(h=0)
```



Increase in iconicity by survival

In the communication condition, innovations that survive tend to increase both iconicity and systematicity. However, in the learning condition, the innovations only contribute to systematicity but not to iconicity.

```
par(mfrow=c(1,2))
plotmeans(increaseIconicity ~ paste(condition,inFinalLang), data=datax[datax$Human,], connect = list(1:
title(main="Iconicity")
axis(1,at=c(1.5,3.5),c("Communication","Reproduction"),line=1, tick=F)
abline(h=0)
plotmeans(systematicity.increase ~ paste(condition, inFinalLang), data = datax[datax$Human,], connect =
title(main="Systematicity")
axis(1,at=c(1.5,3.5),c("Communication","Reproduction"),line=1, tick=F)
abline(h=0)
```



Build a mixed effects model predicting the increase in iconicity with random effects for chain, generation and item:

```
m0= lmer(increaseIconicity ~ 1 +(1|chain) + (1|meaning) + (1|gen),
data=datax[datax$Human,])
m1= lmer(increaseIconicity ~ condition +(1|chain) + (1|meaning) + (1|gen),
data=datax[datax$Human,])
m2= lmer(increaseIconicity ~ condition+inFinalLang +(1|chain) + (1|meaning) + (1|gen),
data=datax[datax$Human,])
m3= lmer(increaseIconicity ~ condition*inFinalLang +(1|chain) + (1|meaning) + (1|gen),
data=datax[datax$Human,])
```

Model comparison test:

```
anova(m0,m1,m2,m3)
```

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

```
## Data: datax[datax$Human, ]
```

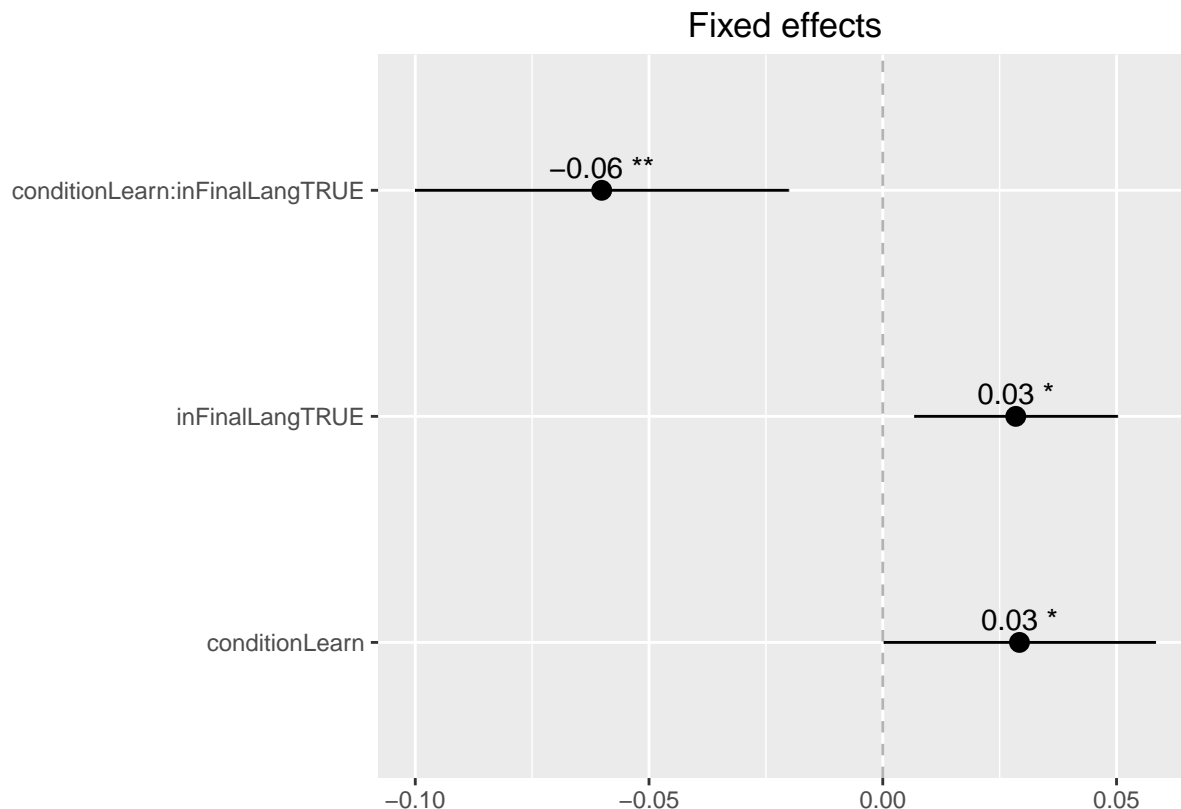
```
## Models:
## m0: increaseIconicity ~ 1 + (1 | chain) + (1 | meaning) + (1 | gen)
## m1: increaseIconicity ~ condition + (1 | chain) + (1 | meaning) +
## m1:      (1 | gen)
## m2: increaseIconicity ~ condition + inFinalLang + (1 | chain) + (1 |
## m2:      meaning) + (1 | gen)
## m3: increaseIconicity ~ condition * inFinalLang + (1 | chain) + (1 |
## m3:      meaning) + (1 | gen)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0  5 -1029.3 -1004.91 519.65 -1039.3
## m1  6 -1027.3 -998.03 519.65 -1039.3 0.0012      1 0.971891
## m2  7 -1026.6 -992.47 520.31 -1040.6 1.3218      1 0.250265
## m3  8 -1033.3 -994.31 524.67 -1049.3 8.7170      1 0.003153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There was a significant interaction between condition and survival to transmission ($\beta = -0.06$, $\text{std.err} = 0.02$, Wald $t = -3$; log likelihood difference = 4.4 , $df = 1$, Chi Squared = 8.72 , $p = 0.0032$).

Plot the fixed effects from the final model:

```
sjp.lmer(m3, 'fe', geom.colors = c(1,1), p.kr=F)
```

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



Increase in systematicity by survival

Build a mixed effects model predicting the increase in systematicity with random effects for chain, generation and item:


```

m0= lmer(systematicity.increase ~ 1 +(1|chain) + (1|meaning) + (1|gen),
         data=datax[datax$Human,])
m1= lmer(systematicity.increase ~ condition +(1|chain) + (1|meaning) + (1|gen),
         data=datax[datax$Human,])
m2= lmer(systematicity.increase ~ condition+inFinalLang +(1|chain) + (1|meaning) + (1|gen),
         data=datax[datax$Human,])
m3= lmer(systematicity.increase ~ condition*inFinalLang +(1|chain) + (1|meaning) + (1|gen),
         data=datax[datax$Human,])

```

Model comparison test:

```
anova(m0,m1,m2,m3)
```

```

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

## Warning in optwrap(optimizer, devfun, x@theta, lower = x@lower, calc.derivs
## = TRUE, : convergence code 3 from bobyqa: bobyqa -- a trust region step
## failed to reduce q

## Data: datax[datax$Human, ]
## Models:
## m0: systematicity.increase ~ 1 + (1 | chain) + (1 | meaning) + (1 |
## m0:      gen)
## m1: systematicity.increase ~ condition + (1 | chain) + (1 | meaning) +
## m1:      (1 | gen)
## m2: systematicity.increase ~ condition + inFinalLang + (1 | chain) +
## m2:      (1 | meaning) + (1 | gen)
## m3: systematicity.increase ~ condition * inFinalLang + (1 | chain) +
## m3:      (1 | meaning) + (1 | gen)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0   5 -2632.7 -2608.3 1321.3  -2642.7
## m1   6 -2631.2 -2602.0 1321.6  -2643.2 0.5701      1    0.4502
## m2   7 -2632.5 -2598.3 1323.2  -2646.5 3.2414      1    0.0718 .
## m3   8 -2631.5 -2592.5 1323.8  -2647.5 1.0682      1    0.3014
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

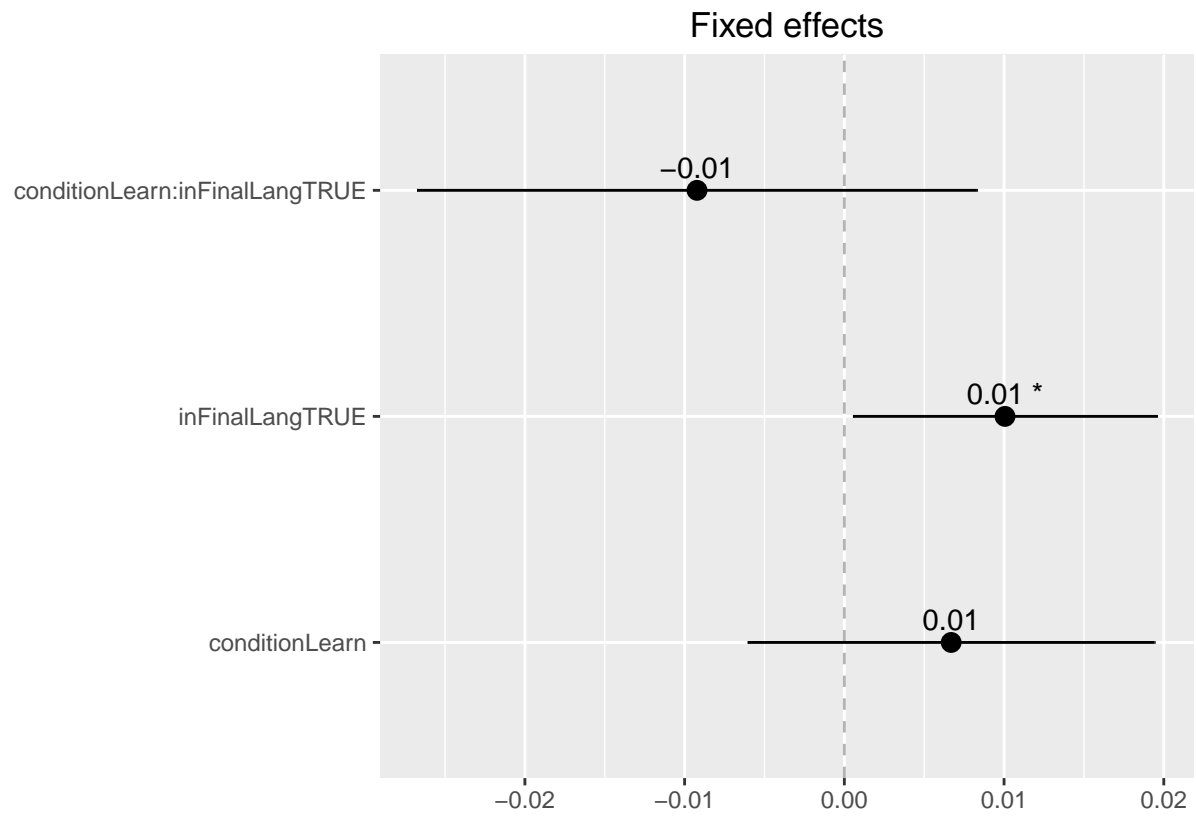
There was no significant main effect of condition ($\beta = -0.0092$, $\text{std.err} = 0.0089$, Wald $t = -1$; log likelihood difference = 0.53 , $\text{df} = 1$, Chi Squared = 1.07 , $p = 0.3$).

There was no significant interaction between condition and survival to transmission ($\beta = -0.0092$, $\text{std.err} = 0.0089$, Wald $t = -1$; log likelihood difference = 0.53 , $\text{df} = 1$, Chi Squared = 1.07 , $p = 0.3$).

Plot the fixed effects from the final model:

```
sjp.lmer(m3,'fe', geom.colors = c(1,1), p.kr = F)
```

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

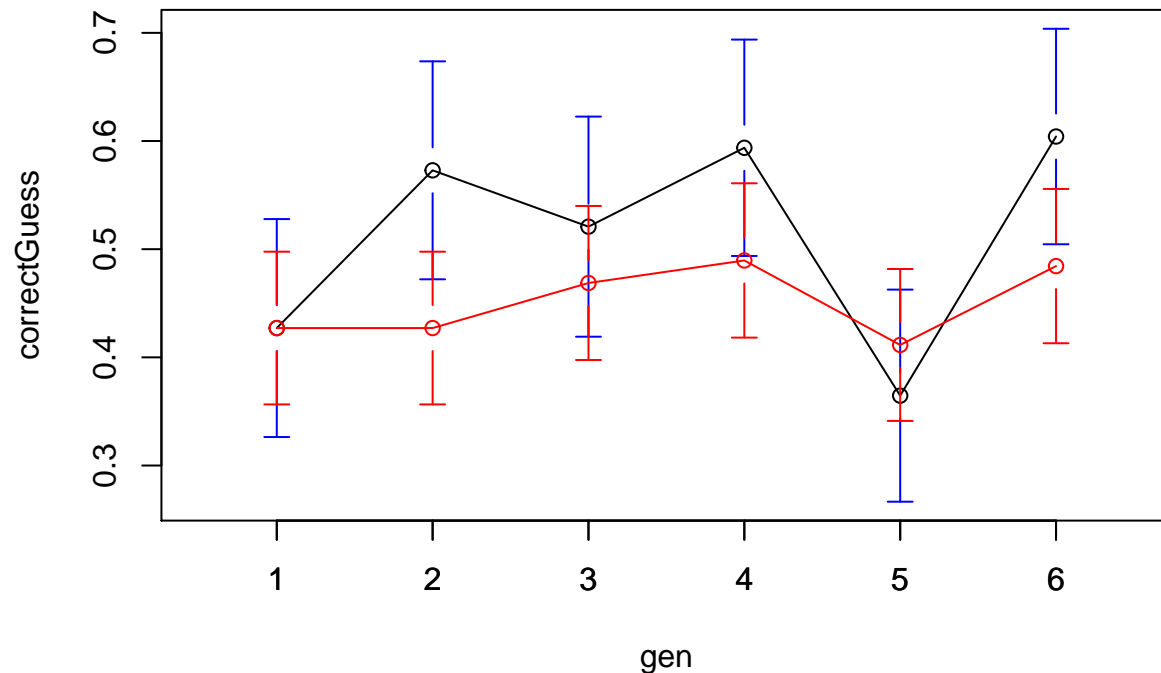


Accuracy

The mean proportion of correct guesses in the communication condition was 45.14%. The mean proportion of correct guesses by the human participant in the learning condition was 51.39%.

Plot the correct guesses by generation (means and 95% confidence intervals):

```
plotmeans(correctGuess~gen,alldatx[alldatx$condition=='Learn' & !alldatx$Human,], n.label = F)
plotmeans(correctGuess~gen,alldatx[alldatx$condition=='Comm',],add=T,col=2,barcol=2, n.label = F)
```



Mixed effects model

Binomial mixed effects model, with random effects for chain, target item. Test whether there are differences between conditions.

```
ctrl = glmerControl(optCtrl = list(maxfun=50000))
# we want to exclude trials where the computer is guessing meanings
# from the participant's signals in the reproduction condition
m0 = glmer(correctGuess ~ 1 + (1|chain) + (1|target.meaning) ,
  data=alldatx[alldatx$condition=='Comm' | (!alldatx$Human),],
  family = binomial, control= ctrl)

m1 = glmer(correctGuess ~ 1 + (1|chain) + (1|target.meaning) + (1|gen),
  data=alldatx[alldatx$condition=='Comm' | (!alldatx$Human),],
  family = binomial, control= ctrl)
m2 = glmer(correctGuess ~ condition + (1|chain) + (1|target.meaning)+ (1|gen),
  data=alldatx[alldatx$condition=='Comm' | (!alldatx$Human),],
  family = binomial, control= ctrl)
anova(m0,m1,m2)

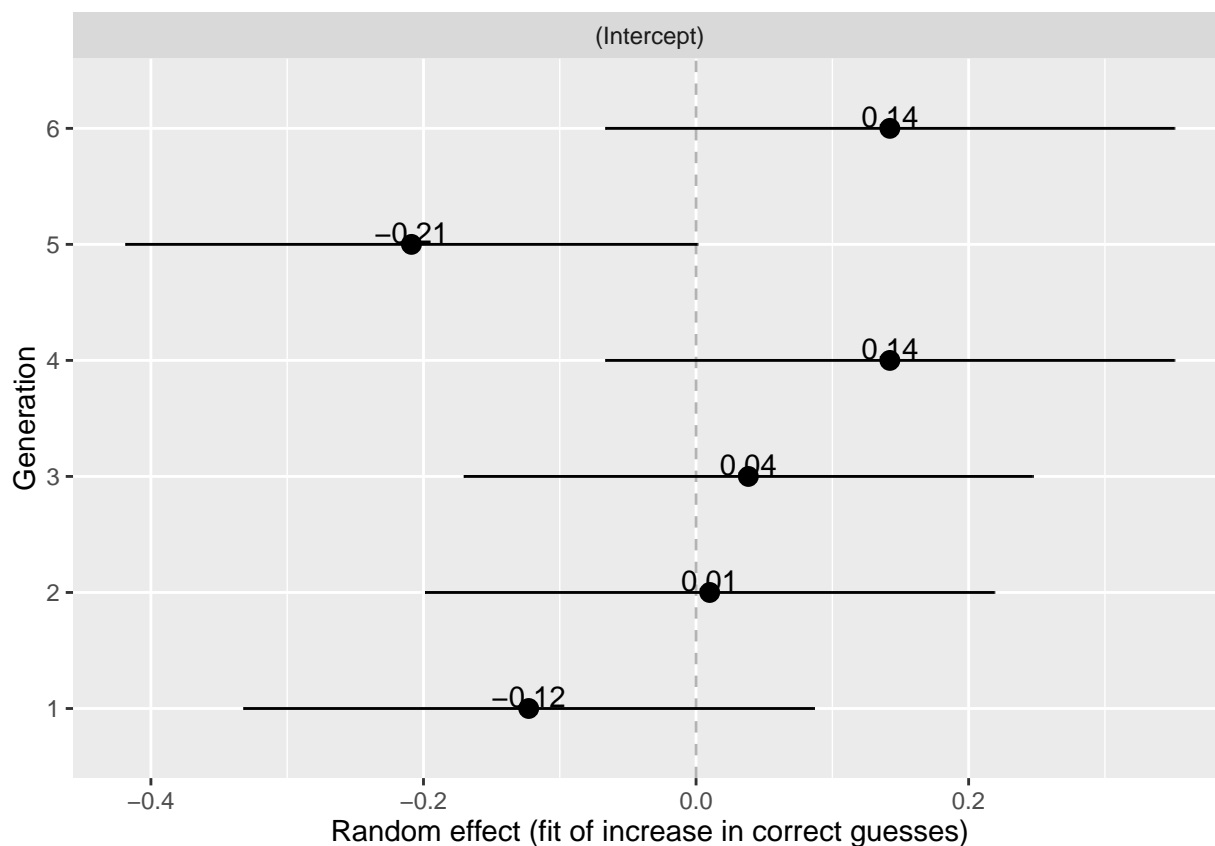
## Data: alldatx[alldatx$condition == "Comm" | (!alldatx$Human), ]
## Models:
## m0: correctGuess ~ 1 + (1 | chain) + (1 | target.meaning)
```

```
## m1: correctGuess ~ 1 + (1 | chain) + (1 | target.meaning) + (1 |
## m1:    gen)
## m2: correctGuess ~ condition + (1 | chain) + (1 | target.meaning) +
## m2:    (1 | gen)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0   3 2375.7 2392.0 -1184.8  2369.7
## m1   4 2372.8 2394.6 -1182.4  2364.8 4.9062      1 0.02676 *
## m2   5 2373.1 2400.4 -1181.5  2363.1 1.6748      1 0.19562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There was no significant main effect of condition ($\beta = 0.26$, $\text{std.err} = 0.19$, $\text{Wald } t = 1.4$, $\text{Wald } p = 0.17$; $\log \text{likelihood difference} = 0.84$, $\text{df} = 1$, $\text{Chi Squared} = 1.67$, $p = 0.2$).

There was a significant difference between generations ($\log \text{likelihood difference} = 2.5$, $\text{df} = 1$, $\text{Chi Squared} = 4.91$, $p = 0.027$). There is a weak trend for the proportion of correct guesses to increase by generation, as shown by the estimates for the random effects for generation:

```
x = sjp.lmer(m2, sort.est='sort.all', prnt.plot=F,
             geom.colors=c(1,1),
             facet.grid = T)
x$plot.list[[3]] +
  xlab("Generation") +
  ylab("Random effect (fit of increase in correct guesses)")
```

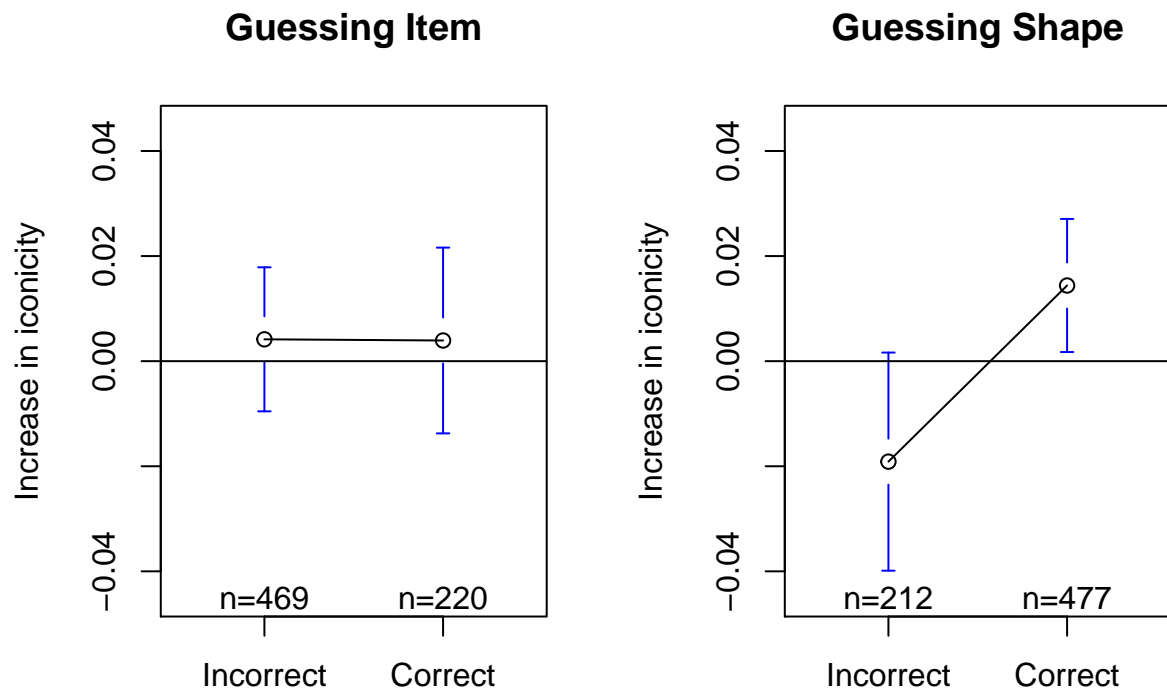


Iconicity and accuracy for innovations

Innovations are either more or less iconic than the words they replace. There is no difference in how accurate the guesses are in terms of choosing the right item (see below, left), but the innovation tends to be more iconic when the shape of a meaning is guessed correctly (spiky or round). That is, the iconicity is helping participants guess the shape of a target meaning correctly.

Note that this analysis only makes sense for the communication condition.

```
par(mfrow=c(1,2))
ylimx = c(-0.045,0.045)
plotmeans(increaseIconicity ~ paste(condition, correctGuess),
  data = datax[datax$Human & datax$condition=="Comm",],
  ylim=ylimx, legends = c("Incorrect","Correct"),
  xlab='',
  ylab="Increase in iconicity")
title("Guessing Item")
abline(h=0)
plotmeans(increaseIconicity ~ paste(condition, correctSpikiness),
  data = datax[datax$Human & datax$condition=="Comm",],
  ylim=ylimx, legends = c("Incorrect","Correct"),
  xlab='',
  ylab="Increase in iconicity")
title("Guessing Shape")
abline(h=0)
```



Mixed effects model for accuracy and iconicity

A mixed effects model predicting the increase in iconicity by whether the receiver selected the correct target item, and by whether the receiver selected an item which matched the target in the shape dimension, with random effects for chain, generation and item. Note that it would make more intuitive sense to predict accuracy by increase in iconicity, but this way we can compare the effects of item accuracy versus shape accuracy.

```
m0 = lmer(increaseIconicity ~ 1 + (1|chain) + (1|gen) + (1|meaning),
          data=datax[datax$condition=="Comm",])

m1 = lmer(increaseIconicity ~ correctGuess + (1|chain) + (1|gen)+ (1|meaning),
          data=datax[datax$condition=="Comm",])

m2 = lmer(increaseIconicity ~ correctGuess + correctSpikiness + (1|chain) + (1|gen)+ (1|meaning),
          data=datax[datax$condition=="Comm",])

anova(m0,m1,m2)

## refitting model(s) with ML (instead of REML)
## Data: datax[datax$condition == "Comm", ]
## Models:
## m0: increaseIconicity ~ 1 + (1 | chain) + (1 | gen) + (1 | meaning)
## m1: increaseIconicity ~ correctGuess + (1 | chain) + (1 | gen) +
## m1:      (1 | meaning)
## m2: increaseIconicity ~ correctGuess + correctSpikiness + (1 | chain) +
## m2:      (1 | gen) + (1 | meaning)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0   5 -692.34 -669.66 351.17  -702.34
## m1   6 -690.34 -663.13 351.17  -702.34 0.0004      1  0.984852
## m2   7 -698.34 -666.59 356.17  -712.34 9.9963      1  0.001569 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m2)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## increaseIconicity ~ correctGuess + correctSpikiness + (1 | chain) +
##      (1 | gen) + (1 | meaning)
##      Data: datax[datax$condition == "Comm", ]
##
## REML criterion at convergence: -689.9
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -3.3006 -0.3909 -0.0257  0.4642  3.2271
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
## meaning (Intercept) 0.00000  0.0000
## gen      (Intercept) 0.00000  0.0000
## chain    (Intercept) 0.00000  0.0000
## Residual                    0.02091  0.1446
## Number of obs: 689, groups: meaning, 12; gen, 6; chain, 4
```

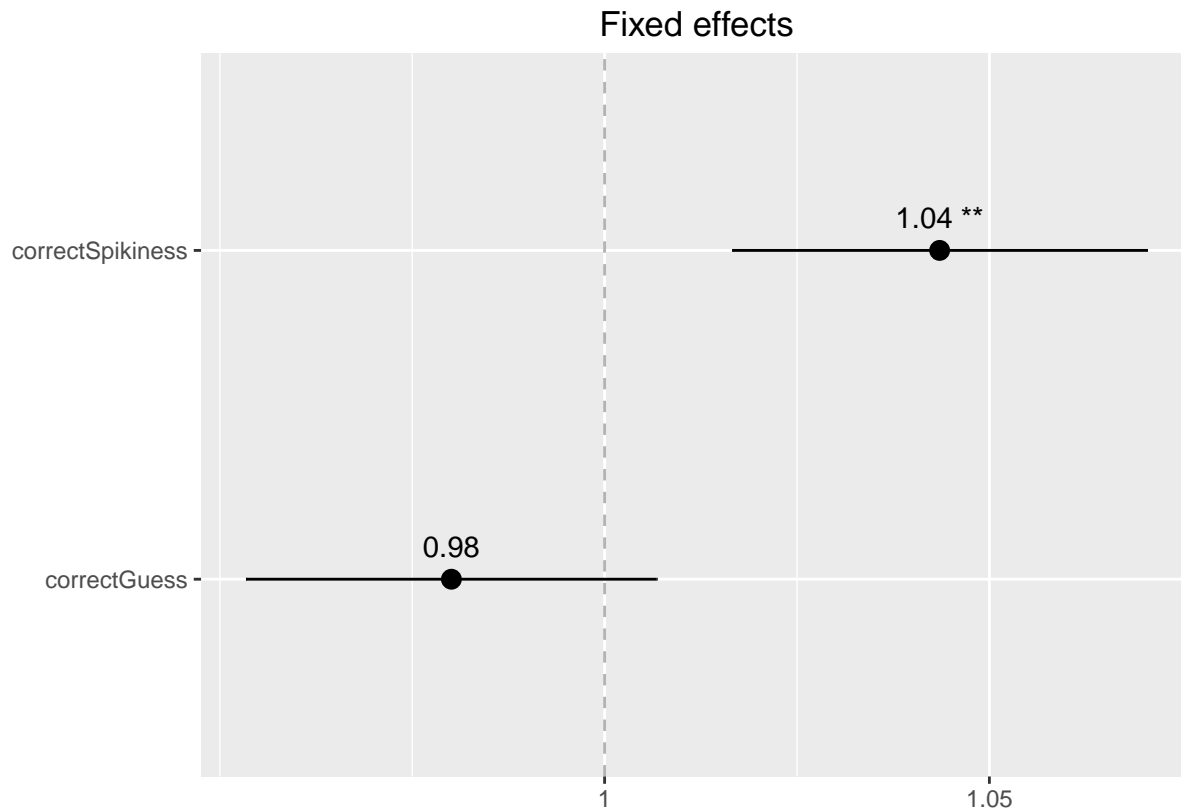
```
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    -0.019121  0.009932  -1.925
## correctGuessTRUE -0.019429  0.013283  -1.463
## correctSpikinessTRUE 0.042482  0.013417   3.166
##
## Correlation of Fixed Effects:
##           (Intr) cGTRUE
## crrctGsTRUE  0.000
## crrctSpTRUE -0.740 -0.457
```

There was no significant main effect of guessing the item correctly ($\beta = -0.019$, $\text{std.err} = 0.013$, Wald $t = -1.5$; log likelihood difference = 0.00018 , $df = 1$, Chi Squared = 0 , $p = 0.98$).

There was a significant main effect of guessing the shape correctly ($\beta = 0.042$, $\text{std.err} = 0.013$, Wald $t = 3.2$; log likelihood difference = 5 , $df = 1$, Chi Squared = 10 , $p = 0.0016$).

Plot the fixed effects:

```
sjp.glmer(m2, type='fe', geom.colors=c(1,1) )
```



Note that the model is probably overfitted, since the random effects are singular. But the effect is clear from the plot of the raw data.

Iconicity and accuracy for whole data

Make a variable that indicates the iconicity of a word according to its shape. i.e. high if it aligns with shape, low if it does not. In other words, reverse the scale for round meanings.

```

alldatx$estimatedIconicity = alldatx$estimatedSpikinessRating

alldatx$estimatedIconicity[alldatx$target.meaning>5] =
  7 - alldatx$estimatedIconicity[alldatx$target.meaning>5]

```

Plot the raw data

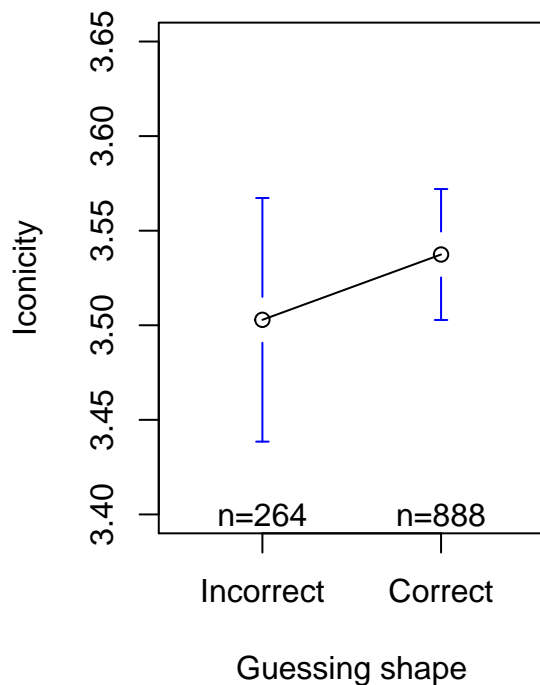
```

par(mfrow=c(1,2))
ylimx= c(3.4,3.65)
plotmeans(estimatedIconicity~correctSpikiness,
  data=alldatx[alldatx$condition=='Comm'],,
  ylim = ylimx,
  xlab="Guessing shape",
  ylab="Iconicity",
  legends = c("Incorrect","Correct"),
  main="Communication")

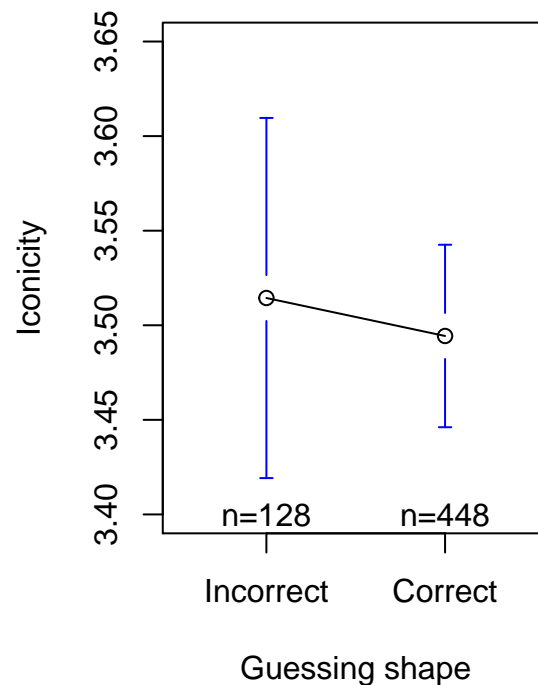
plotmeans(estimatedIconicity~correctSpikiness,
  data=alldatx[alldatx$condition=='Learn' &
    (!alldatx$Human),],
  ylim = ylimx,
  xlab="Guessing shape",
  ylab="Iconicity",
  legends = c("Incorrect","Correct"),
  main="Reproduction")

```

Communication



Reproduction



Iconicity and systematicity

We would like to test whether there is a link between the increase in iconicity and the increase in systematicity for innovations.

Overall correlation between systematicity increase and iconicity increase:

```
cor.test(datax[datax$Human,]$increaseIconicity,datax[datax$Human,]$systematicity.increase)

##
## Pearson's product-moment correlation
##
## data: datax[datax$Human,]$increaseIconicity and datax[datax$Human,]$systematicity.increase
## t = 1.6028, df = 969, p-value = 0.1093
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.01152866 0.11396482
## sample estimates:
## cor
## 0.05142107
```

For the communication condition:

```
cor.test(
  datax[datax$condition=="Comm",]$increaseIconicity,
  datax[datax$condition=="Comm",]$systematicity.increase)

##
## Pearson's product-moment correlation
##
## data: datax[datax$condition == "Comm",]$increaseIconicity and datax[datax$condition == "Comm",]$systematicity.increase
## t = 2.5225, df = 687, p-value = 0.01188
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.02125801 0.16927945
## sample estimates:
## cor
## 0.09579831
```

For the reproduction condition:

```
cor.test(
  datax[datax$condition=="Learn" & datax$Human,]$increaseIconicity,
  datax[datax$condition=="Learn" & datax$Human,]$systematicity.increase)

##
## Pearson's product-moment correlation
##
## data: datax[datax$condition == "Learn" & datax$Human,]$increaseIconicity and datax[datax$condition == "Learn" & datax$Human,]$systematicity.increase
## t = -0.9347, df = 280, p-value = 0.3508
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.17145923 0.06143279
## sample estimates:
## cor
## -0.0557718
```

Simple linear model:

```
summary(lm(increaseIconicity~systematicity.increase*condition, data=datax[datax$Human,]))
```

```
##
## Call:
## lm(formula = increaseIconicity ~ systematicity.increase * condition,
##     data = datax[datax$Human, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51356 -0.06968 -0.00254  0.07290  0.46513
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                   0.003534   0.005393   0.655
## systematicity.increase         0.231376   0.089539   2.584
## conditionLearn                 0.000833   0.010027   0.083
## systematicity.increase:conditionLearn -0.342597   0.155334  -2.206
##                                Pr(>|t|)
## (Intercept)                   0.51240
## systematicity.increase         0.00991 **
## conditionLearn                 0.93381
## systematicity.increase:conditionLearn 0.02765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1414 on 967 degrees of freedom
## Multiple R-squared:  0.007642,    Adjusted R-squared:  0.004563
## F-statistic: 2.482 on 3 and 967 DF,  p-value: 0.0596
```

Significant main effect for systematicity increase and interaction.

Mixed effects model

Build a series of mixed effects models predicting the increase in iconicity with random effects for chain and item.

```
# select data - either trails from the communication condition
# or from the reproduction condition when the human was the speaker
datax.sel = datax[datax$Human,]

# Null model, predicting increase in iconicity by condition and generation
m0 = lmer(increaseIconicity~condition*gen + (1|chain) + (1|meaning),
          data = datax.sel)

# Add main effect of systematicity increase
m1 = lmer(increaseIconicity~systematicity.increase+(condition*gen) + (1|chain) + (1|meaning),
          data = datax.sel)

# Add interaction between Sys. and condition
m2 = lmer(increaseIconicity~systematicity.increase + (systematicity.increase:condition) +
          (condition*gen) + (1|chain) + (1|meaning),
          data = datax.sel)

# Add 3-way interaction
```

```
m3 = lmer(increaseIconicity~systematicity.increase*condition*gen +
          (1|chain) + (1|meaning),
          data = datax.sel)
```

Results

Test the model fit. Note that the model converges on signuative random effect estimates.

```
anova(m0,m1,m2,m3)
```

```
## refitting model(s) with ML (instead of REML)
## Data: datax.sel
## Models:
## m0: increaseIconicity ~ condition * gen + (1 | chain) + (1 | meaning)
## m1: increaseIconicity ~ systematicity.increase + (condition * gen) +
## m1:      (1 | chain) + (1 | meaning)
## m2: increaseIconicity ~ systematicity.increase + (systematicity.increase:condition) +
## m2:      (condition * gen) + (1 | chain) + (1 | meaning)
## m3: increaseIconicity ~ systematicity.increase * condition * gen +
## m3:      (1 | chain) + (1 | meaning)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0  7 -1025.8 -991.71 519.93 -1039.8
## m1  8 -1026.5 -987.44 521.23 -1042.5 2.6088      1      0.1063
## m2  9 -1029.2 -985.33 523.62 -1047.2 4.7734      1      0.0289 *
## m3 11 -1027.6 -973.93 524.80 -1049.6 2.3605      2      0.3072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There was a significant interaction between systematicity increase and condition (log likelihood difference = 2.4 , df = 1 , Chi Squared = 4.77 , p = 0.029).

Plot the fixed effects of model 2. The relationship between systematicity and iconicity is more negative in the reproduction condition:

```
sjp.lmer(m2,type='fe', p.kr = F)
```

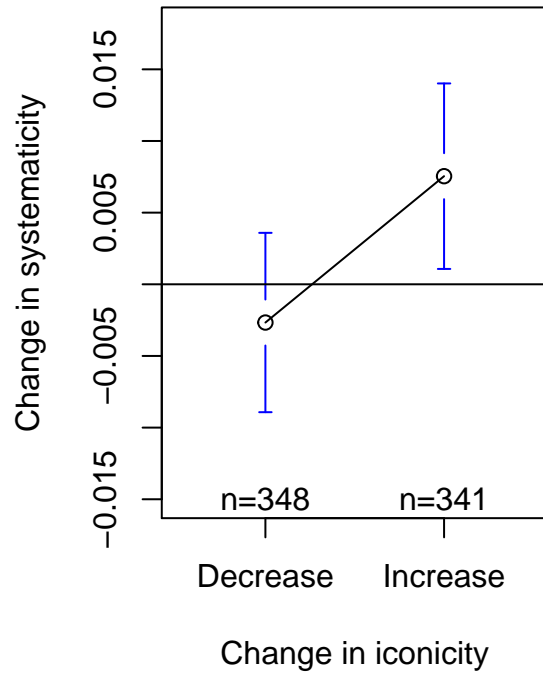
```
## Computing p-values via Wald-statistics approximation (treating t as Wald z).
```



Plot differences in raw data:

```
ylimx= c(-0.015,0.018)
par(mfrow=c(1,2))
plotmeans(systematicity.increase ~ increaseIconicity>0,
  data=datax[datax$condition=="Comm",],
  legends = c("Decrease","Increase"),
  xlab="Change in iconicity",
  ylab="Change in systematicity",
  main = "Communication",
  ylim = ylimx)
abline(h=0)
plotmeans(systematicity.increase ~ increaseIconicity>0,
  data=datax[datax$Human & datax$condition=="Learn",],
  legends = c("Decrease","Increase"),
  xlab="Change in iconicity",
  ylab="Change in systematicity",
  main = "Reproduction",
  ylim = ylimx)
abline(h=0)
```

Communication



Reproduction

