Colour experiment

Table of Contents

# Introduction

This analysis looks at the sign variants used in a colour naming game between signers of different sign languages meeting after 1 week of interaction and after 3 weeks of interaction. The data was collected by Kang Suk Byun.

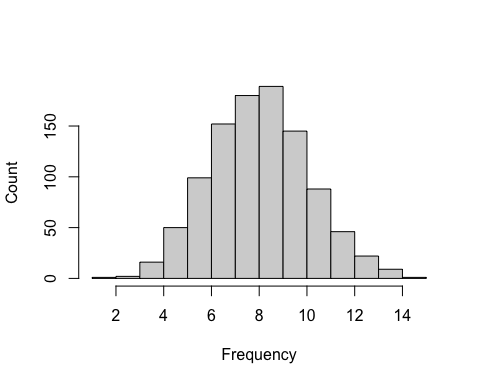
The analysis tries to predict the relative frequency of each variant within a colour category in week 3, based on measures from week 1.

## Data

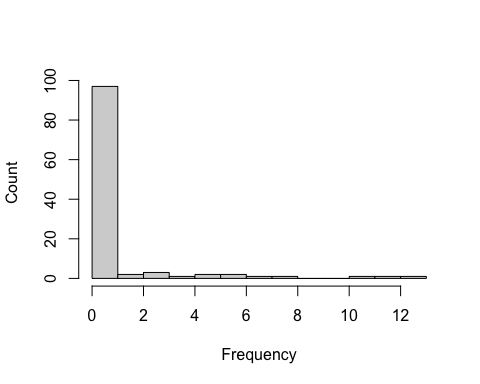
* colour: Code of the target colour
* colourName: English name of the target colour
* sign: label for the variant produced
* freq\_week\_1\_total: Total number of occurances of the variant in the first week, across all colour contexts.
* freq\_week\_4\_total: Total number of occurances of the variant in the final week, across all colour contexts.
* freq\_week\_1: Number of occurances of the variant used during the given target colour context in week 1.
* freq\_week\_4: Number of occurances of the variant used during the given target colour context in the final week.
* prop\_week\_1: Same as freq\_week\_1, but as a proportion of all variants used in the given colour context.
* prop\_week\_4: Same as freq\_week\_4, but as a proportion of all variants used in the given colour context.
* origin: The origin language of the sign. For many, identifying an origin is not possible, so is labelled “None”
* iconic: Old variable
* check: The number of times this variable was used in a checking turn.
* indexical: Is the variant non-indexical, indexical or indexical of the body?
* inventedBy: The name of the first signer to use this variant in the experiment.
* TryMarked: The number of times this sign was used in try-marking.
* Teach: The number of times this sign was explicitly taught.
* averageLength\_week\_1: Average time to produce the variant in milliseconds
* averageTrialLength\_week\_1: Average time for completing the trial for the given target colour.
* BodyAnchor: Is the variant body-anchored (redundant with ‘indexical’)

## Poisson regression

This study uses a mixed effects regression model with poisson distributions. Most standard regression analyses assume that the values they are trying to model come from a normal distribution, like this:



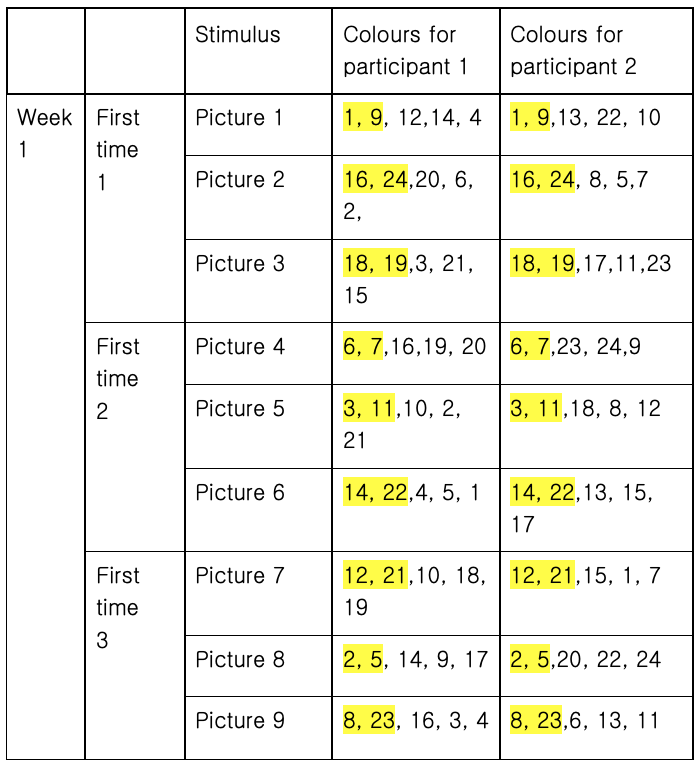
However, the main variable for this study is the frequency of sign variants, with a strong skew and many zero values:



Instead of using a normal distribution as the basis for the statistical model or transforming the data (which is difficult anyway because of the large number of zero counts), we can use a poisson distribution. This also has the advantage of only predicting whole, non-negative numbers, which makes sense for this data because a variant can’t be used half a time or a negative number of times.

## Stimuli

The table below summarises the colours used in each round, which we identified by giving each one a number. The numbers highlighted in yellow were the colours that were the same across both pictures. The table above gives an indication of the kind of planning that was carried out prior to the experiments to ensure robust organisation and reliable use of the materials. The actual table we used is much longer, encompassing all of the dates and rounds, but for the sake of brevity only the part related to the first week of data-gathering is shown here. The numbers in yellow, which indicate colours that were the same across both pictures, are also not repeated after that one use in the week; i.e. 1 and 9 do not appear after the first row. This was done to maximise discussion about as large a range of colours as possible.



Stimuli design

# Load libraries

library(ggplot2)  
library(lme4)  
library(Rmisc)  
library(dplyr)  
library(sjPlot)  
library(gridExtra)  
library(REEMtree)  
library(ggpubr)

# Load data

variants = read.csv('../data/processedData/variants\_summary.csv', stringsAsFactors = F)

There is only 1 variant for `white’. Therefore, we remove it from this statistical analysis.

variants = variants[variants$colourName!='white',]

Some variants need to be removed:

variants = variants[!variants$sign %in%  
 c("SAME",  
 "DIFFERENT",  
 "DO NOT UNDERSTAND",  
 "UNDERSTAND",  
 "ME",  
 "YOU",  
 "YES",  
 "NOT",  
 "SIGNING"),]

Transform some variables.

# The range of values for 'Teach' is very small:  
table(variants$Teach)

##   
## 0 1 2 3   
## 88 6 1 2

# So we'll turn it into a binary category:   
# variants that were never taught and variants that were  
variants$Teach = as.factor(variants$Teach >0)  
  
# Similar for checking   
variants$check.any = as.factor(variants$check>0)  
  
variants$candidateUnderstanding.any = as.factor(variants$candidateUnderstanding>0)  
  
variants$T0 = c("No","Check")[as.numeric(variants$check.any=="TRUE")+1]  
variants$T0[variants$candidateUnderstanding.any=="TRUE"] = "Candidate Understanding"  
  
# Collapse checks and candidate understandings  
variants$T0[variants$T0 %in% c("Check","Candidate Understanding")] = "Yes"  
  
variants$T0 = factor(variants$T0, levels = c("No","Yes"))  
  
# ... and T-1  
  
variants$T\_minus\_1.any = as.factor(variants$T\_minus\_1>0)  
  
# Transform total frequency  
variants$freq\_week\_1\_total.logcenter =   
 log(variants$freq\_week\_1\_total + 1)  
variants$freq\_week\_1\_total.logcenter = scale(variants$freq\_week\_1\_total.logcenter)  
  
# cut TryMarking into two categories  
variants$TryMarked.any = as.factor(variants$TryMarked>0)  
  
# transform length  
  
# One extreme value is scaled down to the next highest number  
variants$averageLength\_week\_1[  
 variants$averageLength\_week\_1>30000] =  
 max(variants$averageLength\_week\_1[  
 variants$averageLength\_week\_1<20000])  
  
variants$averageLength\_week\_1.logcenter = log(variants$averageLength\_week\_1)  
variants$averageLength\_week\_1.logcenter =  
 scale(variants$averageLength\_week\_1.logcenter)  
  
variants$indexical = as.factor(variants$indexical)  
  
# Make inventedBy deviation coding   
# (deviation from the grand mean)  
variants$inventedBy = as.factor(variants$inventedBy)  
#contrasts(variants$inventedBy) = contr.sum(length(levels(variants$inventedBy)))  
  
  
listofsigns = variants[,c("colourName","sign")]  
write.csv(listofsigns,"../results/descriptive/ListOfVariants.csv", fileEncoding = 'utf-8')

# LMER models

Each model predicts the frequency of a variant in week 4, with a random intercept by colourName. The random intercept allows some colours to have higher variant frequencies than others. This is useful because we know that signs for some colours are converged on quickly, making their frequencies within those colours potentially higher. In other words, the use of a particular variant to refer to a given colour is not entierly independent of the use of another variant to refer to the same colour.

Random slopes allow the strength of the effect of a factor to be different for each colour concept. Only indexicality is theoretically relevant here: it is possible that an indexical strategy, particularly body-indexical signs, would be more effective for some colours than others. For example, the body affords indexicality for black (hair) and red (tongue), but not green. We can check whether this is true by comparing a baseline model to one with a random slope for indexicality.

# Optimiser adjustments  
gcontrol = control=glmerControl(  
 optimizer = 'bobyqa',  
 optCtrl = list(maxfun = 1000000))  
  
m0 = glmer(freq\_week\_4 ~   
 1  
 + (1 | colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)  
  
m1 = glmer(freq\_week\_4 ~   
 1  
 + (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

anova(m0,m1)

## Data: variants  
## Models:  
## m0: freq\_week\_4 ~ 1 + (1 | colourName)  
## m1: freq\_week\_4 ~ 1 + (1 + indexical || colourName)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## m0 2 375.66 380.81 -185.83 371.66   
## m1 8 308.28 328.88 -146.14 292.28 79.383 6 4.791e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Indeed, a random slope for indexicality significantly improves the model. It is difficult to think of a theoretical reason why different colours would be affected differently for other variables. Given that random slopes make the model more complex, these are left out.

We begin with a null model and gradually add predictor variables, using model comparison to judge the significance of each variable.

# Null model  
m0 = glmer(freq\_week\_4 ~   
 1  
 + (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mFrq = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mLen = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=glmerControl(  
 optimizer = 'bobyqa',  
 optCtrl = list(maxfun = 10000000)))

## boundary (singular) fit: see ?isSingular

# add indexicality  
mIndx = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=glmerControl(  
 optimizer = c('Nelder\_Mead'),  
 optCtrl = list(maxfun = 100000000)))

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.0149663 (tol = 0.002, component 1)

# add whether the variant is explicitly taught  
mTeach = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=glmerControl(  
 optimizer = c('Nelder\_Mead','bobyqa'),  
 optCtrl = list(maxfun = 10000000)))

## boundary (singular) fit: see ?isSingular

mTry = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mChk = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mTm1 = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 T\_minus\_1.any +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=glmerControl(  
 optimizer = c('Nelder\_Mead','bobyqa'),  
 optCtrl = list(maxfun = 10000000)))

## boundary (singular) fit: see ?isSingular

mInv = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 T\_minus\_1.any +  
 inventedBy +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

Note that the model mTm1 throws a convergence warning. It suggests that the estimates were still changing to some extent when the estimation process finished running. However, a more accurate estimation of the model graident shows that the model is actually converged suitably:

relgrad <- with(mTm1@optinfo$derivs,   
 solve(Hessian,gradient))  
max(abs(relgrad))

## [1] 5.614077e-06

Test some interactions:

mT0xF = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 T\_minus\_1.any +  
 inventedBy +  
 T0:freq\_week\_1\_total.logcenter +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mTchxF = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 T\_minus\_1.any +  
 inventedBy +  
 T0:freq\_week\_1\_total.logcenter +  
 Teach:freq\_week\_1\_total.logcenter +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

mTryXF = glmer(freq\_week\_4 ~   
 1 +   
 freq\_week\_1\_total.logcenter +  
 averageLength\_week\_1.logcenter +  
 indexical +   
 Teach +  
 TryMarked.any +  
 T0 +  
 T\_minus\_1.any +  
 inventedBy +  
 T0:freq\_week\_1\_total.logcenter +  
 Teach:freq\_week\_1\_total.logcenter +  
 TryMarked.any:freq\_week\_1\_total.logcenter +  
 (1 + indexical || colourName) ,   
 data=variants, family=poisson,  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

# Results

Model comparison test:

Results of the model comparison tests. In the output below, there is a summary of the structure of each model, then a table with titles “Df”, “AIC”, “BIC” etc. Each row represents an increasingly complex model, with each fixed effect being added one at a time. The statistics describe how much better the model fits the data compared to the model above. A significant result (low p-value) suggests that the fixed effect significantly improves the fit of the model, and therefore is a good predictor of frequency of variants in the first week.

modelComparisonResults =   
 anova(m0,mFrq,mLen,mIndx,mTeach,mTry,mChk,  
 mTm1,mInv,mT0xF,mTchxF, mTryXF)  
modelComparisonResults

## Data: variants  
## Models:  
## m0: freq\_week\_4 ~ 1 + (1 + indexical || colourName)  
## mFrq: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + (1 + indexical || colourName)  
## mLen: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + (1 + indexical || colourName)  
## mIndx: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + (1 + indexical || colourName)  
## mTeach: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + (1 + indexical || colourName)  
## mTry: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + (1 + indexical || colourName)  
## mChk: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + (1 + indexical || colourName)  
## mTm1: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any + (1 + indexical || colourName)  
## mInv: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any + inventedBy + (1 + indexical || colourName)  
## mT0xF: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any + inventedBy + T0:freq\_week\_1\_total.logcenter + (1 + indexical || colourName)  
## mTchxF: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any + inventedBy + T0:freq\_week\_1\_total.logcenter + Teach:freq\_week\_1\_total.logcenter + (1 + indexical || colourName)  
## mTryXF: freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter + indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any + inventedBy + T0:freq\_week\_1\_total.logcenter + Teach:freq\_week\_1\_total.logcenter + TryMarked.any:freq\_week\_1\_total.logcenter + (1 + indexical || colourName)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## m0 8 308.28 328.88 -146.141 292.28   
## mFrq 9 252.04 275.21 -117.020 234.04 58.2425 1 2.317e-14 \*\*\*  
## mLen 10 249.09 274.84 -114.546 229.09 4.9466 1 0.0261426 \*   
## mIndx 12 248.88 279.77 -112.439 224.88 4.2145 2 0.1215733   
## mTeach 13 237.71 271.18 -105.853 211.71 13.1713 1 0.0002843 \*\*\*  
## mTry 14 234.27 270.32 -103.136 206.27 5.4340 1 0.0197488 \*   
## mChk 15 214.50 253.12 -92.248 184.50 21.7768 1 3.063e-06 \*\*\*  
## mTm1 16 211.88 253.07 -89.939 179.88 4.6172 1 0.0316528 \*   
## mInv 19 209.07 257.99 -85.535 171.07 8.8086 3 0.0319475 \*   
## mT0xF 20 189.28 240.78 -74.641 149.28 21.7882 1 3.045e-06 \*\*\*  
## mTchxF 21 173.15 227.22 -65.577 131.15 18.1280 1 2.065e-05 \*\*\*  
## mTryXF 22 152.79 209.43 -54.394 108.79 22.3668 1 2.252e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Choose a final model for the beta values.

finalModel = mTryXF

Write model comparison results to file:

mcr = as.data.frame(modelComparisonResults)  
mcr$AIC = round(mcr$AIC,2)  
mcr$BIC = round(mcr$BIC,2)  
mcr$logLik = round(mcr$logLik,2)  
mcr$deviance = round(mcr$deviance,2)  
mcr$Chisq = round(mcr$Chisq,2)  
mcr$Significance = ""  
mcr$Significance[mcr$`Pr(>Chisq)`<0.05] = "\*"  
mcr$p = mcr$`Pr(>Chisq)`  
mcr$p = round(mcr$p,2)  
mcr$p[mcr$p<0.001] = "< 0.001"  
  
mcr = mcr[,c("AIC","BIC","logLik","Chisq",'Df','p',"Significance")]  
names(mcr) = c("AIC","BIC","Log likelihood","χ2","Df",'p','')  
mcr[is.na(mcr)] = ""  
  
rownames(mcr) = c("Null model","Frequency","Length",  
 "Indexicality","Teach","Try-marking",  
 "T0","T-1",'Identity of first user',  
 "T0 x Frequency","T-1 x Frequency",  
 'Try-marking x Frequency')  
write.csv(mcr,   
 "../results/inferential/ModelComparisonResults.csv",  
 row.names = T, fileEncoding = "utf-8")

## Final model parameter summary

This show the estimates for all parameters in the model.

summary(finalModel)

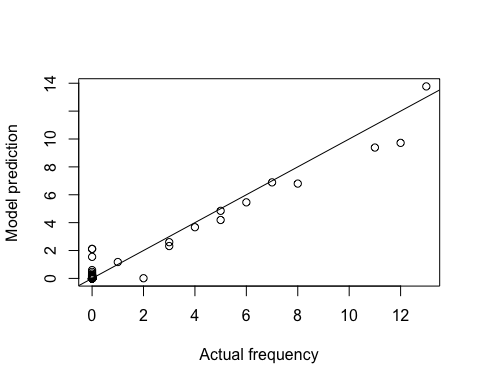
## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: poisson ( log )  
## Formula:   
## freq\_week\_4 ~ 1 + freq\_week\_1\_total.logcenter + averageLength\_week\_1.logcenter +   
## indexical + Teach + TryMarked.any + T0 + T\_minus\_1.any +   
## inventedBy + T0:freq\_week\_1\_total.logcenter + Teach:freq\_week\_1\_total.logcenter +   
## TryMarked.any:freq\_week\_1\_total.logcenter + (1 + indexical ||   
## colourName)  
## Data: variants  
## Control: gcontrol  
##   
## AIC BIC logLik deviance df.resid   
## 152.8 209.4 -54.4 108.8 75   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.4544 -0.0890 -0.0014 0.0000 18.2088   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## colourName (Intercept) 0.00000 0.000   
## colourName.1 indexicalno 14.90386 3.861   
## indexicalyes 3.31863 1.822 -1.00   
## indexicalYes-body 0.09983 0.316 1.00 -1.00  
## Number of obs: 97, groups: colourName, 6  
##   
## Fixed effects:  
## Estimate Std. Error z value  
## (Intercept) -13.7285 4.1553 -3.304  
## freq\_week\_1\_total.logcenter 7.1689 2.5202 2.845  
## averageLength\_week\_1.logcenter -1.0595 0.2555 -4.146  
## indexicalyes -1.7894 2.6629 -0.672  
## indexicalYes-body 0.7043 2.2258 0.316  
## TeachTRUE 11.3216 2.6302 4.305  
## TryMarked.anyTRUE 1.3326 1.4854 0.897  
## T0Yes 10.4168 2.8247 3.688  
## T\_minus\_1.anyTRUE -2.2656 0.7723 -2.934  
## inventedByIndonesia -3.0290 2.3089 -1.312  
## inventedByJordan 2.2129 0.7426 2.980  
## inventedByNepal 1.8396 0.7256 2.535  
## freq\_week\_1\_total.logcenter:T0Yes -10.4990 3.0042 -3.495  
## freq\_week\_1\_total.logcenter:TeachTRUE -9.6475 2.2786 -4.234  
## freq\_week\_1\_total.logcenter:TryMarked.anyTRUE 5.7483 1.7613 3.264  
## Pr(>|z|)   
## (Intercept) 0.000954 \*\*\*  
## freq\_week\_1\_total.logcenter 0.004446 \*\*   
## averageLength\_week\_1.logcenter 3.38e-05 \*\*\*  
## indexicalyes 0.501601   
## indexicalYes-body 0.751665   
## TeachTRUE 1.67e-05 \*\*\*  
## TryMarked.anyTRUE 0.369667   
## T0Yes 0.000226 \*\*\*  
## T\_minus\_1.anyTRUE 0.003349 \*\*   
## inventedByIndonesia 0.189549   
## inventedByJordan 0.002881 \*\*   
## inventedByNepal 0.011233 \*   
## freq\_week\_1\_total.logcenter:T0Yes 0.000474 \*\*\*  
## freq\_week\_1\_total.logcenter:TeachTRUE 2.30e-05 \*\*\*  
## freq\_week\_1\_total.logcenter:TryMarked.anyTRUE 0.001100 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##   
## Correlation matrix not shown by default, as p = 15 > 12.  
## Use print(x, correlation=TRUE) or  
## vcov(x) if you need it

## optimizer (bobyqa) convergence code: 0 (OK)  
## boundary (singular) fit: see ?isSingular

Check the predictions. Each point in the graph below is a variant. The actual frequency in the 3rd week is on the horizontal axis. Many have a frequency of 0: they don’t appear. The model’s predictions are shown on the vertical axis. The diagonal line is just a guide to show a perfect fit. Most of the points lie on this line, suggesting that the model has fit the data well.

plot(variants$freq\_week\_4, exp(predict(finalModel)),  
 xlab="Actual frequency",  
 ylab="Model prediction")  
abline(0,1)



Here is a summary of whether a variant appeared or did not appear in the final week, and the model’s predictions of whether they appeared or not:

##   
## Not observed Observed  
## Predicted: Not observed 81 1  
## Predicted: Observed 3 12

These are the variants used in the final week:

variants[variants$freq\_week\_4>0,c("colourName","sign", 'freq\_week\_4')]

## colourName sign freq\_week\_4  
## 1 red RED(FRONTLET) 3  
## 2 red RED(MOUTH) 11  
## 6 brown IX:'BACK OF ONE'S HAND' 8  
## 10 brown BOOGER 2  
## 15 brown IX:WATCH 5  
## 28 black BLACK 13  
## 39 green GREEN-CHEST 12  
## 47 green CORRECT 5  
## 48 yellow YELLOW-CHEST 1  
## 72 yellow SUN 4  
## 74 yellow YELLOW(ISL) 6  
## 87 pink FLOWER 7  
## 92 pink IX:TONGUE 3

There were 13 variants that had a frequency of greater than 0 in the final week.

The model predicts 12 of these correctly, missing 1 variants (BOOGER).

The model predicts a further 3 variants should have been observed in the final week but were not observed in reality:

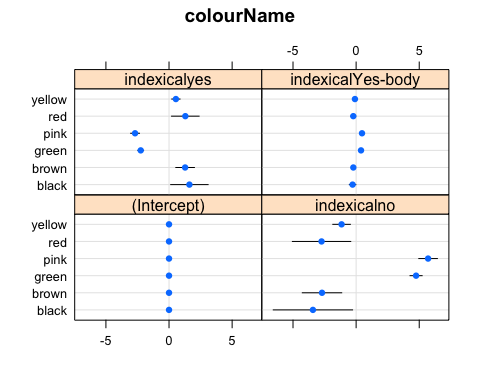
## colourName sign  
## 5 red RED-BLOOD  
## 7 brown DARK  
## 36 green GREEN-NSL

## Random slopes results

We can plot the random slopes below:

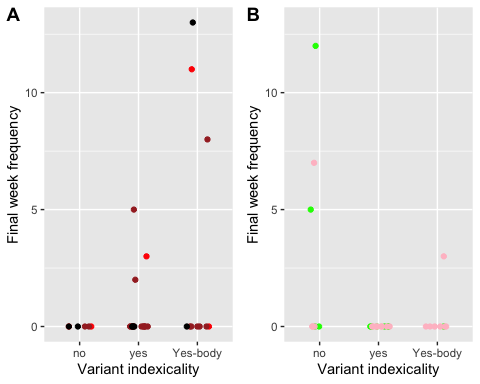
dotplot(ranef(finalModel))

## $colourName



The effect of indexicality is stronger for red, brown, and black. It is weaker for pink and green.

indxRedBrownBlack = ggplot(variants[variants$colourName %in% c("red","brown","black"),],  
 aes(x=indexical, y = freq\_week\_4,colour = colourName)) +   
 geom\_point(position = position\_jitter(height = 0, width=0.2)) +  
 scale\_colour\_manual(breaks=c("red","brown","black"), values = c("red","brown","black")) +  
 theme(legend.position = 'none') +  
 xlab("Variant indexicality") +  
 ylab("Final week frequency")+  
 coord\_cartesian(ylim=c(0,13))  
  
indxPinkGreen = ggplot(variants[variants$colourName %in% c("pink","green"),],  
 aes(x=indexical, y = freq\_week\_4,colour = colourName)) +   
 geom\_point(position = position\_jitter(height = 0, width=0.2)) +  
 scale\_colour\_manual(breaks=c("pink","green"), values = c("pink","green")) +  
 theme(legend.position = 'none') +  
 xlab("Variant indexicality") +  
 ylab("Final week frequency") +  
 coord\_cartesian(ylim=c(0,13))  
  
indx = ggarrange(indxRedBrownBlack, indxPinkGreen, ncol=2, labels=c("A","B"))  
indx



pdf("../results/descriptive/graphs/Indexicality\_RedBrownBlack\_GreenPink.pdf",  
 width = 6,height=3)  
indx  
dev.off()

## quartz\_off\_screen   
## 2

Check whether the coefficients are radically different with a full random effects structure:

finalModelFull = update(finalModel, ~. +   
 (0 + Teach + check.any +  
 T\_minus\_1.any + inventedBy || colourName),  
 control=gcontrol)

## boundary (singular) fit: see ?isSingular

cor.test(fixef(finalModel), fixef(finalModelFull))

##   
## Pearson's product-moment correlation  
##   
## data: fixef(finalModel) and fixef(finalModelFull)  
## t = 5.625, df = 13, p-value = 8.269e-05  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5795946 0.9461222  
## sample estimates:  
## cor   
## 0.8418948

The correlation is high, suggesting that the extra random slopes would not have a big effect on the results.

# Summary

Here is a summary of the main results:

There was a significant main effect of frequency in week 1 ( beta = 7.2 , log likelihood difference = 29 , df = , Chi Squared = 58.24 , p = 2.3e-14 ). More frequent forms in the first week were more likely to be more frequent in the final week.

There was a significant main effect of sign length ( beta = -1.1 , log likelihood difference = 2.5 , df = , Chi Squared = 4.95 , p = 0.026 ). Shorter signs were more frequent in the final week.

There was no significant main effect of indexicality ( beta = -1.8 , log likelihood difference = 2.1 , df = , Chi Squared = 4.21 , p = 0.12 ). (beta for body-indexical = 0.853).

There was a significant main effect of teaching ( beta = 11 , log likelihood difference = 6.6 , df = , Chi Squared = 13.17 , p = 0.00028 ). Teaching increased the frequency of variants in the final week.

There was a significant main effect of try marking ( beta = 1.3 , log likelihood difference = 2.7 , df = , Chi Squared = 5.43 , p = 0.02 ). Variants that were try marked more often were more frequent in the final week.

There was a significant main effect of of repair initiation (T0) ( beta = 10 , log likelihood difference = 11 , df = , Chi Squared = 21.78 , p = 3.1e-06 ). Variants used in a repair initiation were more frequent in the final week.

There was a significant main effect of use in a problem source turn (T-1) ( beta = -2.3 , log likelihood difference = 2.3 , df = , Chi Squared = 4.62 , p = 0.032 ).

There was a significant main effect of first user (inventedBy) ( log likelihood difference = 4.4 , df = , Chi Squared = 8.81 , p = 0.032 ). Signs invented by the signer from Jordan and Nepal were more frequent in the final week.

There was a significant interaction between T0 and frequency ( beta = -10 , log likelihood difference = 11 , df = , Chi Squared = 21.79 , p = 3e-06 ). The effect of appearing in a T0 sequence is stronger if the variant was more frequent in the first week.

There was a significant interaction between Teaching and frequency ( log likelihood difference = 9.1 , df = , Chi Squared = 18.13 , p = 2.1e-05 ).. The effect of teaching was stronger if the variant was more frequent in the first week.

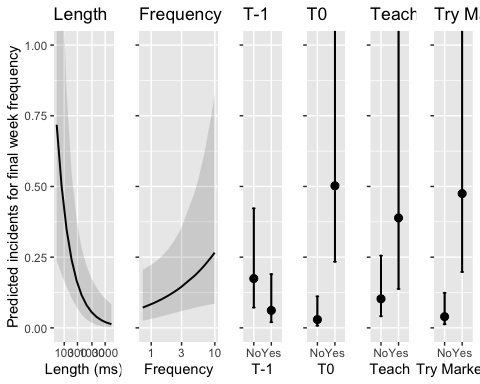
There was a significant interaction between Try marking and frequency ( log likelihood difference = 11 , df = , Chi Squared = 22.37 , p = 2.3e-06 ).. The effect of try marking is stronger if the variant was more frequent in the first week.

We also found some evidence that the effect of indexicality is more important for black and red, and less important for green and pink.

# Graphs

Plot of the marginal effects of the predicted probabilities for each fixed effect:

x = plot\_model(mInv, 'eff',   
 vars=c('Teach',  
 'TryMarked.any',  
 'T0',  
 'freq\_week\_1\_total.logcenter',  
 'averageLength\_week\_1.logcenter',  
 'T\_minus\_1.any'),   
 facet.grid = F,  
 prnt.plot = F,  
 show.ci = T)  
  
freqi = 1  
leni = 2  
t0i = 6  
teachi = 4  
tryi = 5  
tm1i = 7  
  
# Rescale frequency back to real values  
freq.m = x[[freqi]]$data$x  
x[[freqi]]$data$x =   
 exp(freq.m\*   
 attr(variants$freq\_week\_1\_total.logcenter, 'scaled:scale') +   
 attr(variants$freq\_week\_1\_total.logcenter, 'scaled:center') -1)  
  
x[[freqi]]$labels$x = "Frequency"  
x[[freqi]]$labels$title = "Frequency"  
  
# Rescale length back to real values  
len.m = x[[leni]]$data$x  
x[[leni]]$data$x =   
 exp(len.m\*   
 attr(variants$averageLength\_week\_1.logcenter, 'scaled:scale') +   
 attr(variants$averageLength\_week\_1.logcenter, 'scaled:center') -1)  
  
#x$plot.list[[1]]$labels$title = "Indexicality"  
#x$plot.list[[1]]$scales$scales[[1]]$labels = c("No","Yes","Body")  
  
x[[teachi]]$labels$title = "Teaching"  
x[[teachi]]$scales$scales[[1]]$labels = c("No","Yes")  
x[[teachi]]$coordinates$limits$x = c(0.5,2.5)  
  
x[[tryi]]$labels$title = "Try Marked"  
x[[tryi]]$scales$scales[[1]]$labels = c("No","Yes")  
x[[tryi]]$coordinates$limits$x = c(0.5,2.5)  
  
x[[t0i]]$labels$title = "T0"  
x[[t0i]]$scales$scales[[1]]$labels = c("No","Yes")  
x[[t0i]]$coordinates$limits$x = c(0.5,2.5)  
  
x[[tm1i]]$labels$title = "T-1"  
x[[tm1i]]$scales$scales[[1]]$labels = c("No","Yes")  
x[[tm1i]]$coordinates$limits$x = c(0.5,2.5)  
  
x[[leni]]$labels$title = "Length"  
for(i in 1:7){  
 x[[i]]$coordinates$limits$y = c(0,1)  
 if(i!=leni){  
 x[[i]]$theme =   
 list(  
 panel.grid.minor.x = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.line.y=element\_blank())  
 } else{  
 x[[i]]$theme = list(panel.grid.minor.x = element\_blank())  
 }  
 x[[i]]$labels$y = element\_blank()  
}  
  
x[[leni]]$labels$y = "Predicted incidents for final week frequency"  
x[[leni]]$labels$x = "Length (ms)"  
x[[tm1i]]$labels$x = "T-1"  
x[[tryi]]$labels$x = "Try Marked"  
#x[[tm1i]]$labels$y = "Predicted incidents for final week frequency"  
  
  
grx = grid.arrange(  
 x[[leni]] + scale\_x\_log10(),  
 x[[freqi]] + scale\_x\_log10(),  
 x[[tm1i]]+   
 geom\_point() +  
 geom\_errorbar(aes(ymin=conf.low,ymax=conf.high), width=0.2),  
 x[[t0i]]+   
 geom\_point() +  
 geom\_errorbar(aes(ymin=conf.low,ymax=conf.high), width=0.2),  
 x[[teachi]]+ geom\_point()+  
 geom\_errorbar(aes(ymin=conf.low,ymax=conf.high), width=0.2),  
 x[[tryi]]+ geom\_point()+  
 geom\_errorbar(aes(ymin=conf.low,ymax=conf.high), width=0.2),   
 widths=c(2.1,1.8,1.1,1.1,1.1,1.1))

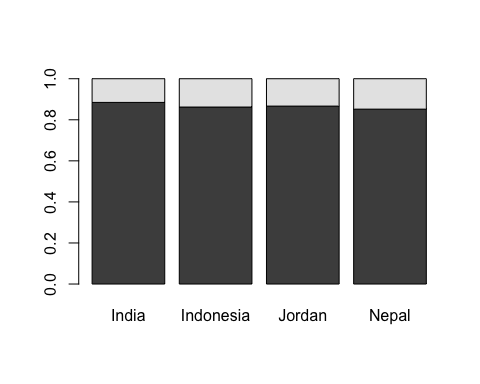


pdf("../results/descriptive/graphs/EffectsPlot.pdf", width=8.5, height=3.5)  
plot(grx)  
dev.off()

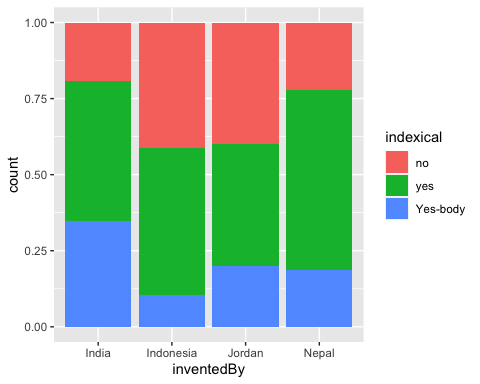
## quartz\_off\_screen   
## 2

Plot the average frequency in the final week for signs invented by each signer:

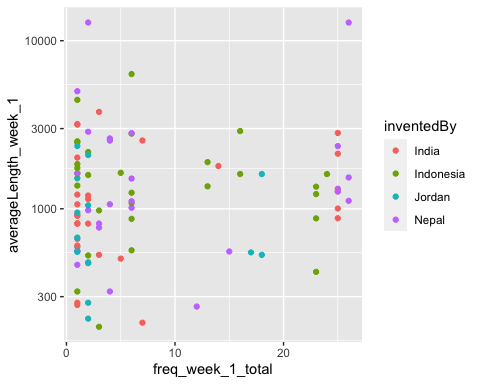
itx = prop.table(table(  
 variants$inventedBy,  
 variants$freq\_week\_4>0), margin=1)  
barplot(t(itx))



ggplot(variants,   
 aes(x=inventedBy, fill=indexical)) +  
 geom\_bar(position="fill")



ggplot(variants,   
 aes(x=freq\_week\_1\_total,  
 y=averageLength\_week\_1,  
 colour=inventedBy)) +  
 geom\_point() + scale\_y\_log10()



Plot interactions between frequency and sequential variables. Here we’re just visualising the rough relationship in the raw data in order to understand the direction of the interaction.

variants$freq\_week\_1\_total.cut = cut(variants$freq\_week\_1\_total,  
 breaks = quantile(variants$freq\_week\_1\_total,  
 probs = c(0,1/3,2/3,1)),include.lowest = T,  
 labels = c("low","medium","high"))  
xi = variants %>% group\_by(Teach, freq\_week\_1\_total.cut) %>%  
 summarise(freq\_week\_4 = mean(freq\_week\_4))

## `summarise()` has grouped output by 'Teach'. You can override using the `.groups` argument.

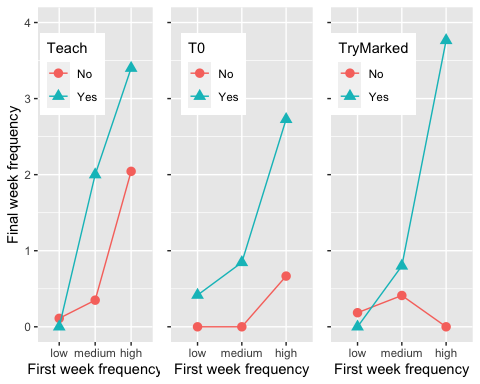
int1 = ggplot(xi,aes(x=freq\_week\_1\_total.cut,y=freq\_week\_4,group=Teach,colour=Teach,shape=Teach)) +  
 geom\_point(size=3) + geom\_line() +  
 ylab("Final week frequency") +  
 xlab("First week frequency") +  
 theme(legend.position = c(0.3,0.8)) +  
 coord\_cartesian(ylim=c(0,4)) +  
 scale\_colour\_discrete(name = "Teach", labels = c("No", "Yes")) +  
 scale\_shape\_discrete(name = "Teach", labels = c("No", "Yes"))   
  
xi = variants %>% group\_by(T0, freq\_week\_1\_total.cut) %>%  
 summarise(freq\_week\_4 = mean(freq\_week\_4))

## `summarise()` has grouped output by 'T0'. You can override using the `.groups` argument.

int2 = ggplot(xi,aes(x=freq\_week\_1\_total.cut,y=freq\_week\_4,group=T0,colour=T0,shape=T0)) +  
 geom\_point(size=3) + geom\_line() +  
 xlab("First week frequency") +  
 theme(legend.position = c(0.3,0.8),   
 axis.title.y = element\_blank(),  
 axis.text.y = element\_blank()) +  
 coord\_cartesian(ylim=c(0,4))  
  
xi = variants %>% group\_by(TryMarked.any,freq\_week\_1\_total.cut) %>%  
 summarise(freq\_week\_4 = mean(freq\_week\_4))

## `summarise()` has grouped output by 'TryMarked.any'. You can override using the `.groups` argument.

int3 = ggplot(xi,aes(x=freq\_week\_1\_total.cut,y=freq\_week\_4,group=TryMarked.any,  
 colour=TryMarked.any,shape=TryMarked.any)) +  
 geom\_point(size=3) + geom\_line() +  
 xlab("First week frequency") +  
 theme(legend.position = c(0.3,0.8),   
 axis.title.y = element\_blank(),  
 axis.text.y = element\_blank()) +  
 coord\_cartesian(ylim=c(0,4)) +  
 #labs(group="TryMarked",colour="TryMarked",shape="TryMarked")  
 scale\_colour\_discrete(name = "TryMarked", labels = c("No", "Yes")) +  
 scale\_shape\_discrete(name = "TryMarked", labels = c("No", "Yes"))   
  
gx = grid.arrange(int1,int2,int3, layout\_matrix=t(matrix(c(1,2,3))))



pdf("../results/descriptive/graphs/Interactions.pdf",  
 width=6,height=3)  
plot(gx)  
dev.off()

## quartz\_off\_screen   
## 2

gx

## TableGrob (1 x 3) "arrange": 3 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]

Check model assumptions. The model is not good at predicting higher values and less common cases.

plot\_model(finalModel, 'resid')

