

Colour experiment

Contents

Introduction	2
Data	2
Poisson regression	3
Load libraries	4
Load data	4
LMER models	5
Results	9
Random slopes results	13
Summary	15
Graphs	16

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 (Kang-Suk.Byun@mpi.nl).

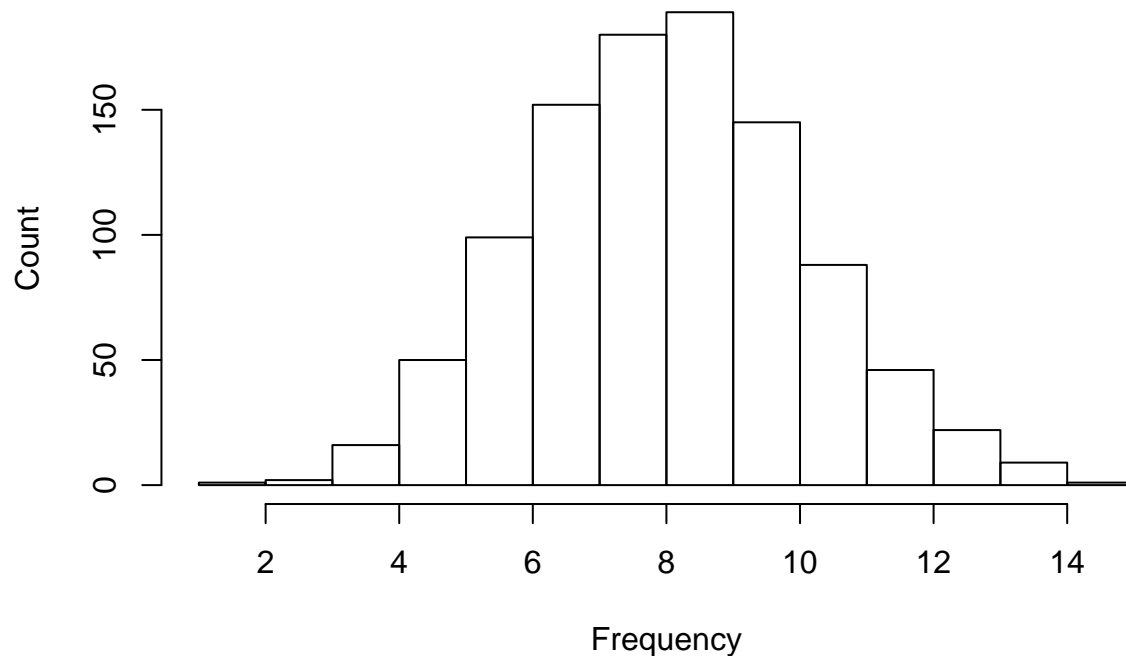
The analysis tries to predict the relative frequency of each variant within a colour category in week 3, based on measures from week 1.

Data

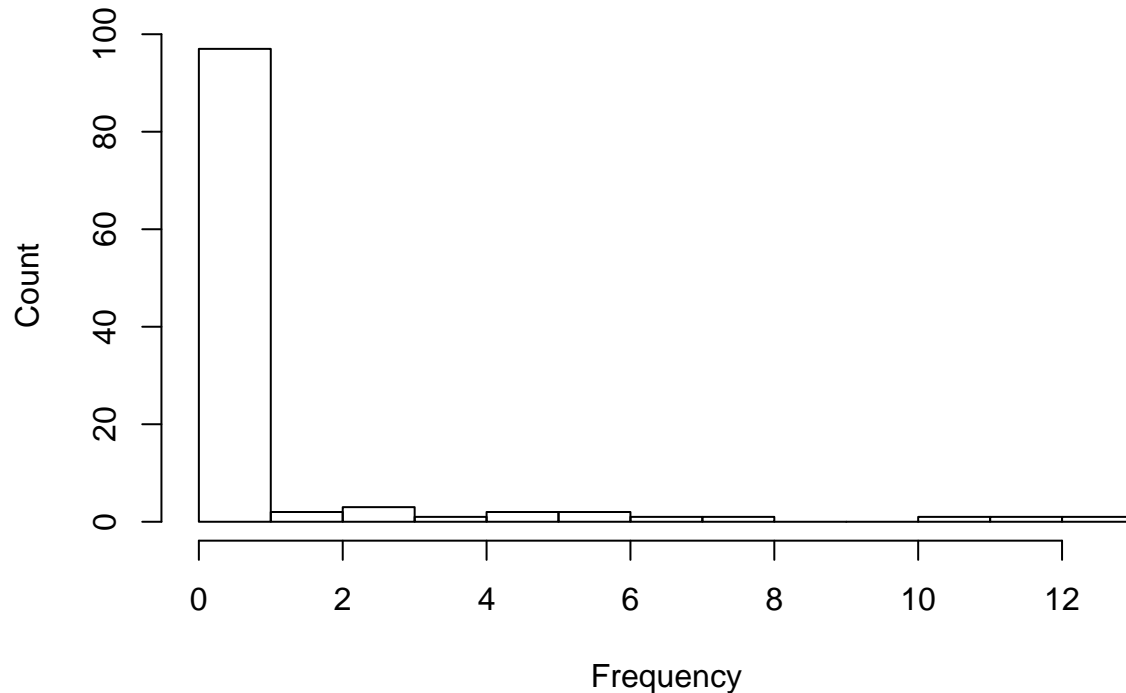
- colour: Code of the target colour
- colourName: English name of the target colour
- sign: label for the variant produced
- freq_week_1_total: Total number of occurrences of the variant in the first week, across all colour contexts.
- freq_week_4_total: Total number of occurrences of the variant in the final week, across all colour contexts.
- freq_week_1: Number of occurrences of the variant used during the given target colour context in week 1.
- freq_week_4: Number of occurrences 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.

Load libraries

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

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 and candidate understandings
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"
```

```
variants$T0 = factor(variants$T0, levels = c("No", "Check", "Candidate Understanding"))
```

```

# ... 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$aaverageLength_week_1[
  variants$aaverageLength_week_1>30000] =
  max(variants$aaverageLength_week_1[
    variants$aaverageLength_week_1<20000])

variants$aaverageLength_week_1.logcenter = log(variants$aaverageLength_week_1)
variants$aaverageLength_week_1.logcenter =
  scale(variants$aaverageLength_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 entirely 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)
anova(m0,m1)

## Data: variants
## Models:
## m0: freq_week_4 ~ 1 + (1 | colourName)
## m1: freq_week_4 ~ 1 + (1 + indexical || colourName)
##      Df      AIC      BIC logLik deviance Chisq Chi 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)

mFrq = glmer(freq_week_4 ~
  1 +
  freq_week_1_total.logcenter +
  (1 + indexical || colourName) ,
  data=variants, family=poisson,
  control=gcontrol)

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 = 1000000)))

# 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','bobyqa'),
    optCtrl = list(maxfun = 1000000)))

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.0149221 (tol =
## 0.001, 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 = 1000000)))

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)

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)

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=gcontrol)

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)

```

Test some interactions:

```

mTOxF = 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)

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)

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)

```

Results

Model comparison test:

```

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

```

```

## Data: variants
## Models:
## m0: freq_week_4 ~ 1 + (1 + indexical || colourName)
## mFrq: freq_week_4 ~ 1 + freq_week_1_total.logcenter + (1 + indexical ||
## mFrq: colourName)
## mLen: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mLen: (1 + indexical || colourName)
## mIndx: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mIndx: indexical + (1 + indexical || colourName)
## mTeach: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mTeach: indexical + Teach + (1 + indexical || colourName)
## mTry: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mTry: indexical + Teach + TryMarked.any + (1 + indexical || colourName)
## mChk: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mChk: indexical + Teach + TryMarked.any + T0 + (1 + indexical ||
## mChk: colourName)
## mTm1: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mTm1: indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
## mTm1: (1 + indexical || colourName)
## mInv: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mInv: indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
## mInv: inventedBy + (1 + indexical || colourName)
## mT0xF: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mT0xF: indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
## mT0xF: inventedBy + T0:freq_week_1_total.logcenter + (1 + indexical ||

```

```

## mT0xF:      colourName)
## mTchxF: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mTchxF:      indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
## mTchxF:      inventedBy + T0:freq_week_1_total.logcenter + Teach:freq_week_1_total.logcenter +
## mTchxF:      (1 + indexical || colourName)
## mTryXF: freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
## mTryXF:      indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
## mTryXF:      inventedBy + T0:freq_week_1_total.logcenter + Teach:freq_week_1_total.logcenter +
## mTryXF:      TryMarked.any:freq_week_1_total.logcenter + (1 + indexical ||
## mTryXF:      colourName)
##          Df      AIC      BIC    logLik deviance   Chisq Chi 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 0.1215713
## 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       16 216.30 257.49  -92.148   184.30 21.9771      2 1.689e-05 ***
## mTm1       17 213.86 257.63  -89.929   179.86  4.4384      1 0.0351398 *
## mInv       20 210.34 261.83  -85.169   170.34  9.5187      3 0.0231328 *
## mT0xF      22 183.61 240.25  -69.806   139.61 30.7276      2 2.126e-07 ***
## mTchxF     23 172.08 231.30  -63.041   126.08 13.5283      1 0.0002350 ***
## mTryXF     24 155.88 217.68  -53.941   107.88 18.1999      1 1.989e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Choose a final model for the beta values.

```
finalModel = mTryXF
```

Full summary:

```
summary(finalModel)
```

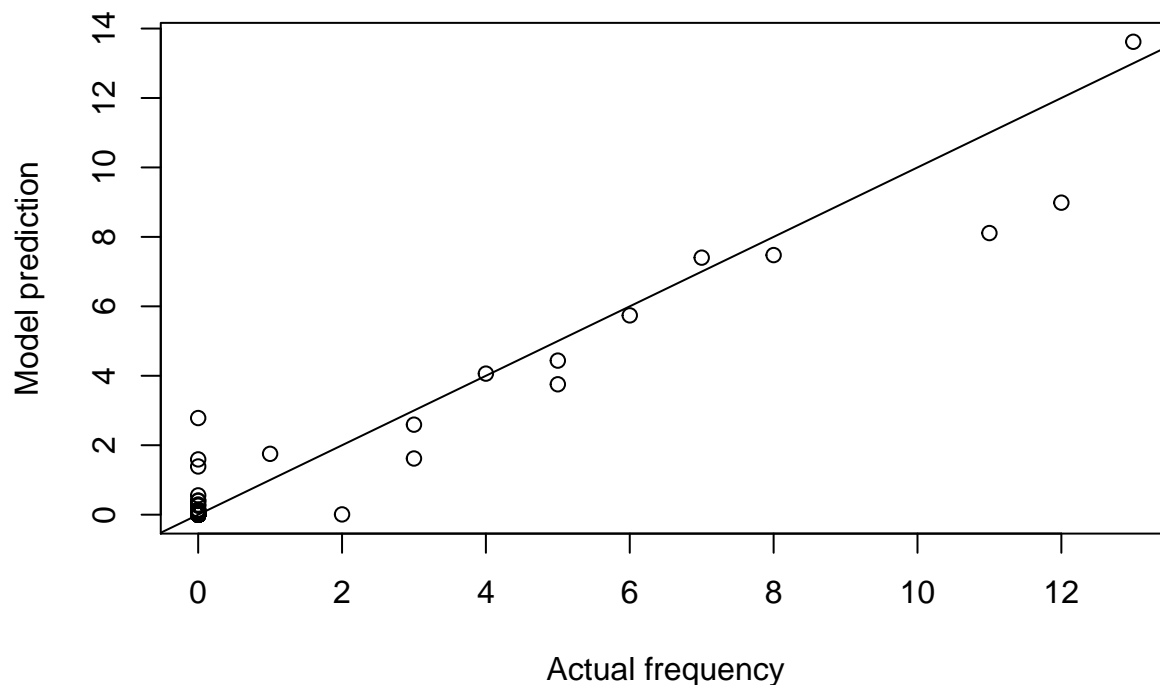
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##   Family: poisson ( log )
## Formula:
## freq_week_4 ~ 1 + freq_week_1_total.logcenter + averageLength_week_1.logcenter +
##   indexical + Teach + TryMarked.any + T0 + T_minus_1.any +
##   inventedBy + 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
##    155.9    217.7    -53.9   107.9      73
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.6683 -0.1241  0.0000   0.0000  22.0078
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   colourName  (Intercept)          0.000000 0.00000
##   colourName.1 indexicalno          0.002736 0.05231
##                  indexicalyes      25.020396 5.00204 -1.00
##                  indexicalYes-body  1.120140 1.05837  1.00 -1.00
## Number of obs: 97, groups: colourName, 6
##
## Fixed effects:
##
##                                     Estimate Std. Error
## (Intercept)                        -17.2755     7.0813
## freq_week_1_total.logcenter          11.3820     5.8878
## averageLength_week_1.logcenter      -1.8517     0.4270
## indexicalyes                        -5.3325     2.9463
## indexicalYes-body                   1.0149     0.9331
## TeachTRUE                          15.9367     3.9007
## TryMarked.anyTRUE                  -1.5702     2.1671
## T0Check                           13.7495     6.4807
## T0Candidate Understanding           15.1447     6.5828
## T_minus_1.anyTRUE                  -2.2868     0.7201
## inventedBy1                        -1.0700     0.6699
## inventedBy2                         0.4462     0.6735
## inventedBy3                        -0.2012     0.7432
## freq_week_1_total.logcenter:T0Check -13.6627     6.1836
## freq_week_1_total.logcenter:T0Candidate Understanding -13.4570     6.1207
## freq_week_1_total.logcenter:TeachTRUE -13.7939     3.3531
## freq_week_1_total.logcenter:TryMarked.anyTRUE  7.1293     2.1440
##
##                                     z value Pr(>|z|)
## (Intercept)                        -2.440 0.014703 *
## freq_week_1_total.logcenter          1.933 0.053216 .
## averageLength_week_1.logcenter      -4.337 1.44e-05 ***
## indexicalyes                        -1.810 0.070310 .
```

```
## indexicalYes-body 1.088 0.276724
## TeachTRUE 4.086 4.40e-05 ***
## TryMarked.anyTRUE -0.725 0.468716
## T0Check 2.122 0.033870 *
## T0Candidate Understanding 2.301 0.021411 *
## T_minus_1.anyTRUE -3.176 0.001494 **
## inventedBy1 -1.597 0.110193
## inventedBy2 0.662 0.507662
## inventedBy3 -0.271 0.786556
## freq_week_1_total.logcenter:T0Check -2.210 0.027138 *
## freq_week_1_total.logcenter:T0Candidate Understanding -2.199 0.027907 *
## freq_week_1_total.logcenter:TeachTRUE -4.114 3.89e-05 ***
## freq_week_1_total.logcenter:TryMarked.anyTRUE 3.325 0.000884 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 17 > 12.
## Use print(x, correlation=TRUE) or
## vcov(x) if you need it
```

Check the predictions:

```
plot(variants$freq_week_4, exp(predict(finalModel)),
     xlab="Actual frequency",
     ylab="Model prediction")
abline(0,1)
```



```
##
##               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:

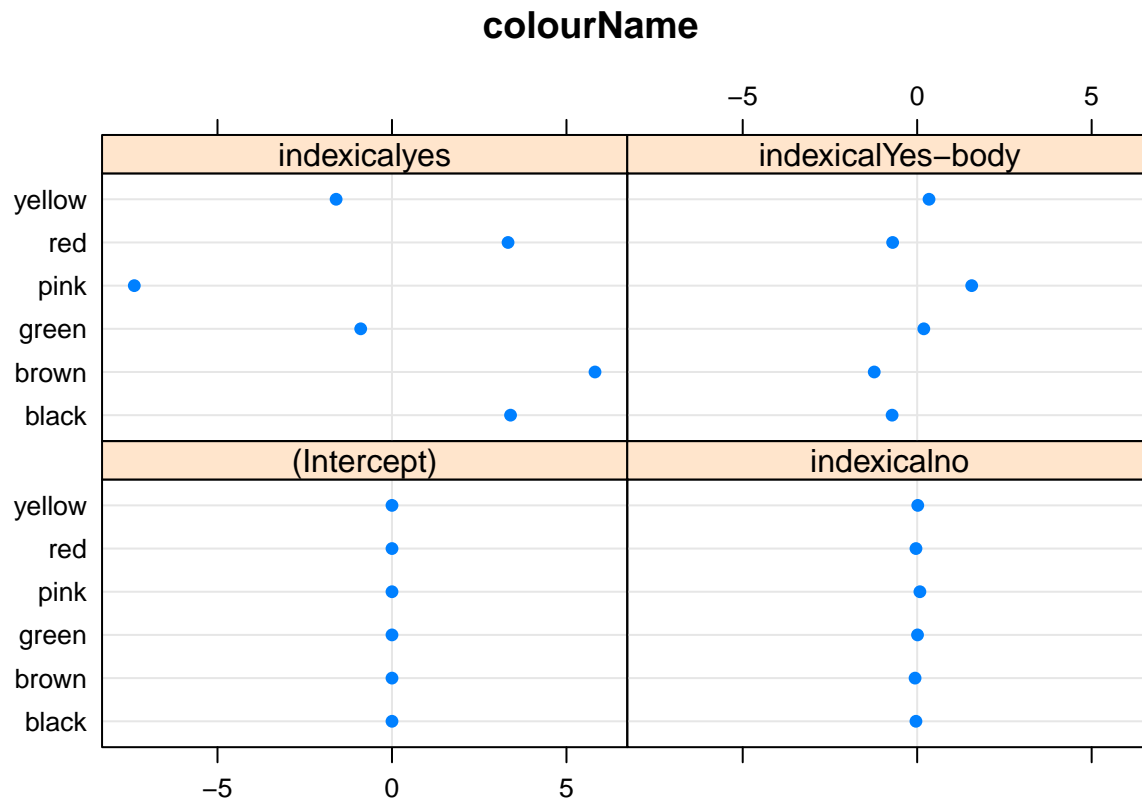
##	colourName	sign
## 5	red	RED-BLOOD
## 7	brown	DARK
## 36	green	GREEN-NSL

Random slopes results

We can plot the random slopes below:

```
dotplot(ranef(finalModel))
```

```
## $colourName
```



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

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

```
##
## Pearson's product-moment correlation
##
## data: fixef(finalModel) and fixef(finalModelFull)
## t = 3.6749, df = 15, p-value = 0.002252
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3103457 0.8783665
## sample estimates:
##      cor
## 0.688317
```

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 ($\beta = 11$, log likelihood difference = 29 , $df = 1$, 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 ($\beta = -1.9$, log likelihood difference = 2.5 , $df = 1$, Chi Squared = 4.95 , $p = 0.026$). Shorter signs were more frequent in the final week.

There was no significant main effect of indexicality ($\beta = -5.3$, log likelihood difference = 2.1 , $df = 2$, Chi Squared = 4.21 , $p = 0.12$). (β for body-indexical = 0.853). Indexical variants were more frequent in the final week.

There was a significant main effect of teaching ($\beta = 16$, log likelihood difference = 6.6 , $df = 1$, 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 ($\beta = -1.6$, log likelihood difference = 2.7 , $df = 1$, 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 checking ($\beta = 14$, log likelihood difference = 11 , $df = 2$, Chi Squared = 21.98 , $p = 1.7e-05$). Variants used in a checking context were more frequent in the final week.

There was a significant main effect of use in repair initiation (T-1) ($\beta = -2.3$, log likelihood difference = 2.2 , $df = 1$, Chi Squared = 4.44 , $p = 0.035$).

There was a significant main effect of first user (inventedBy) (log likelihood difference = 4.8 , $df = 3$, Chi Squared = 9.52 , $p = 0.023$). Signs invented by the signer from Nepal were less frequent in the final week.

There was a significant interaction between T0 and frequency ($\beta = -14$, log likelihood difference = 15 , $df = 2$, Chi Squared = 30.73 , $p = 2.1e-07$). The effect of frequency only applies to variants not appearing in T0 turns.

There was a significant interaction between Teaching and frequency (log likelihood difference = 6.8 , $df = 1$, Chi Squared = 13.53 , $p = 0.00023$).

There was a significant interaction between Try marking and frequency (log likelihood difference = 9.1 , $df = 1$, Chi Squared = 18.2 , $p = 2e-05$).

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 = sjp.glmer(mInv, 'eff',
             vars=c('Teach',
                    'TryMarked.any',
                    'TO',
                    '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 = 5
teachi = 3
tryi = 4
tml1i = 6

# Rescale frequency back to real values
freq.m = x$plot.list[[freqi]]$data$x
x$plot.list[[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$plot.list[[freqi]]$labels$x = "Frequency"
x$plot.list[[freqi]]$labels$title = "Frequency"

# Rescale length back to real values
len.m = x$plot.list[[leni]]$data$x
x$plot.list[[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$plot.list[[teachi]]$labels$title = "Teaching"
x$plot.list[[teachi]]$scales$scales[[1]]$labels = c("No", "Yes\n")
x$plot.list[[teachi]]$coordinates$limits$x = c(0.5, 2.5)

x$plot.list[[tryi]]$labels$title = "Try Marking"
x$plot.list[[tryi]]$scales$scales[[1]]$labels = c("No", "Yes\n")
x$plot.list[[tryi]]$coordinates$limits$x = c(0.5, 2.5)

x$plot.list[[t0i]]$labels$title = "TO"
x$plot.list[[t0i]]$scales$scales[[1]]$labels = c("No", "Check", "Candidate\nUnderstanding")
x$plot.list[[t0i]]$coordinates$limits$x = c(0.5, 3.5)
```



```

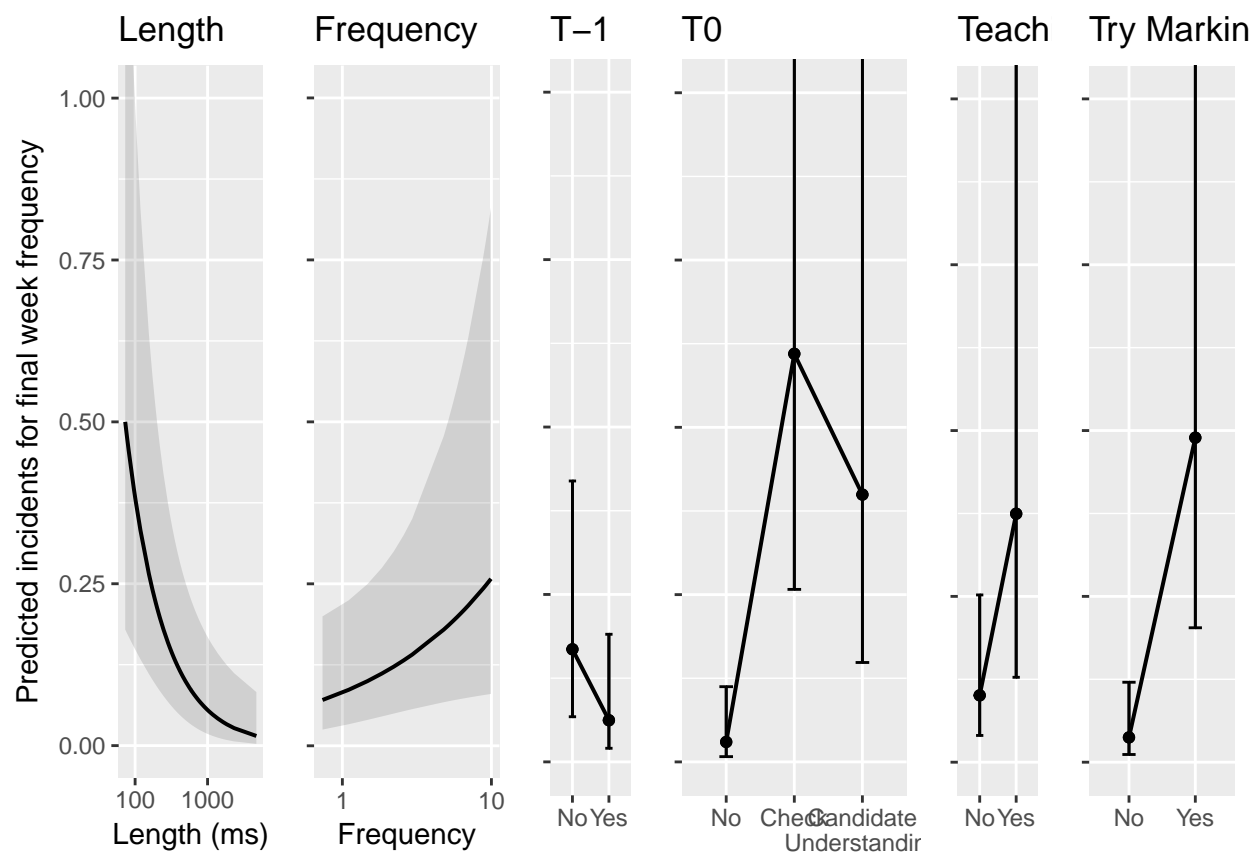
x$plot.list[[tm1i]]$labels$title = "T-1"
x$plot.list[[tm1i]]$scales$scales[[1]]$labels = c("No", "Yes\n")
x$plot.list[[tm1i]]$coordinates$limits$x = c(0.5, 2.5)

x$plot.list[[leni]]$labels$title = "Length"
for(i in 1:6){
  x$plot.list[[i]]$coordinates$limits$y = c(0, 1)
  if(i!=leni){
    x$plot.list[[i]]$theme =
      list(
        panel.grid.minor.x = element_blank(),
        axis.text.y = element_blank(),
        axis.line.y=element_blank())
  } else{
    x$plot.list[[i]]$theme = list(panel.grid.minor.x = element_blank())
  }
  x$plot.list[[i]]$labels$y = element_blank()
}

x$plot.list[[leni]]$labels$y = "Predicted incidents for final week frequency"
x$plot.list[[leni]]$labels$x = "Length (ms)"

grx = grid.arrange(
  x$plot.list[[leni]] + scale_x_log10(),
  x$plot.list[[frequi]] + scale_x_log10(),
  x$plot.list[[tm1i]]+
    geom_point() +
    geom_errorbar(aes(ymin=lower,ymax=upper), width=0.2),
  x$plot.list[[t0i]]+
    geom_point() +
    geom_errorbar(aes(ymin=lower,ymax=upper), width=0.2),
  x$plot.list[[teachi]]+ geom_point()+
    geom_errorbar(aes(ymin=lower,ymax=upper), width=0.2),
  x$plot.list[[tryi]]+ geom_point()+
    geom_errorbar(aes(ymin=lower,ymax=upper), width=0.2),
  widths=c(2.1, 1.8, 1, 2.1, 1, 1.5))

```

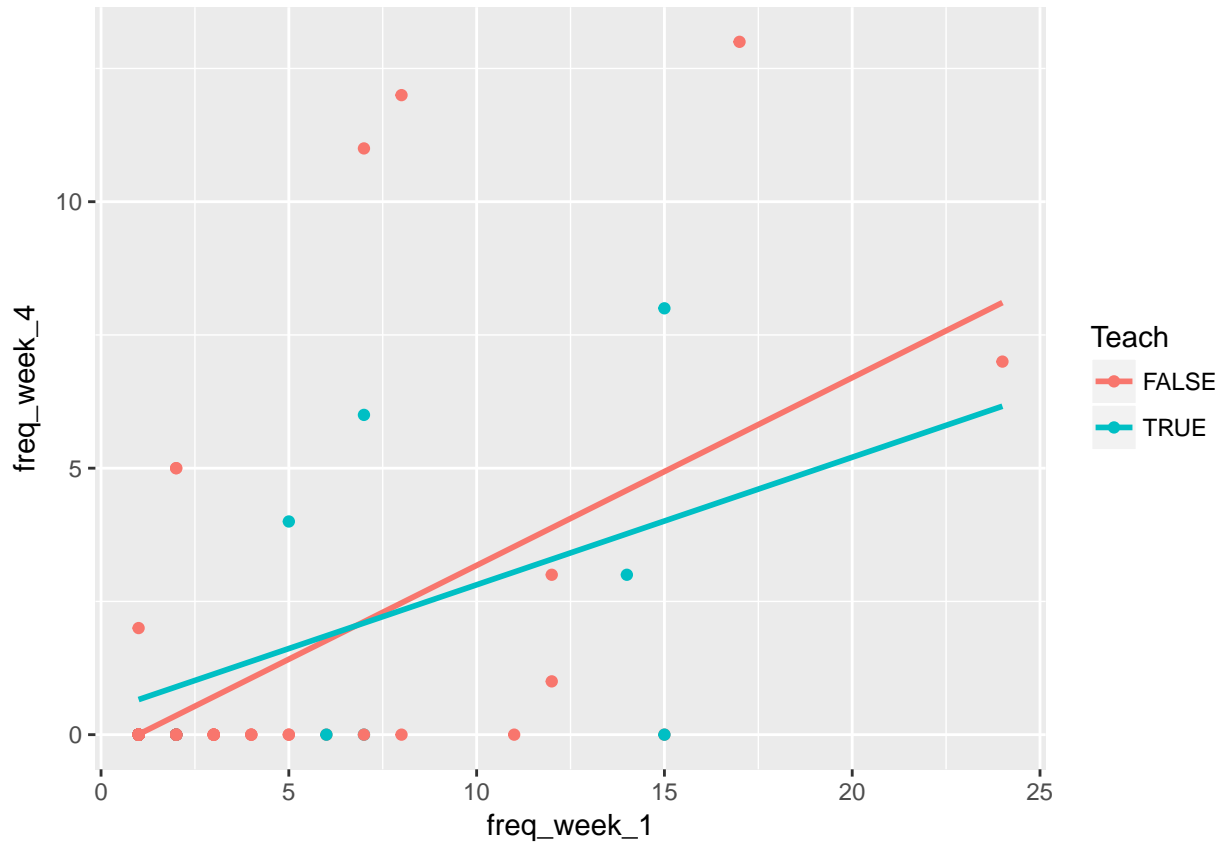


```
pdf("../results/descriptive/graphs/EffectsPlot.pdf", width=8.5, height=3.5)
plot(grx)
dev.off()
```

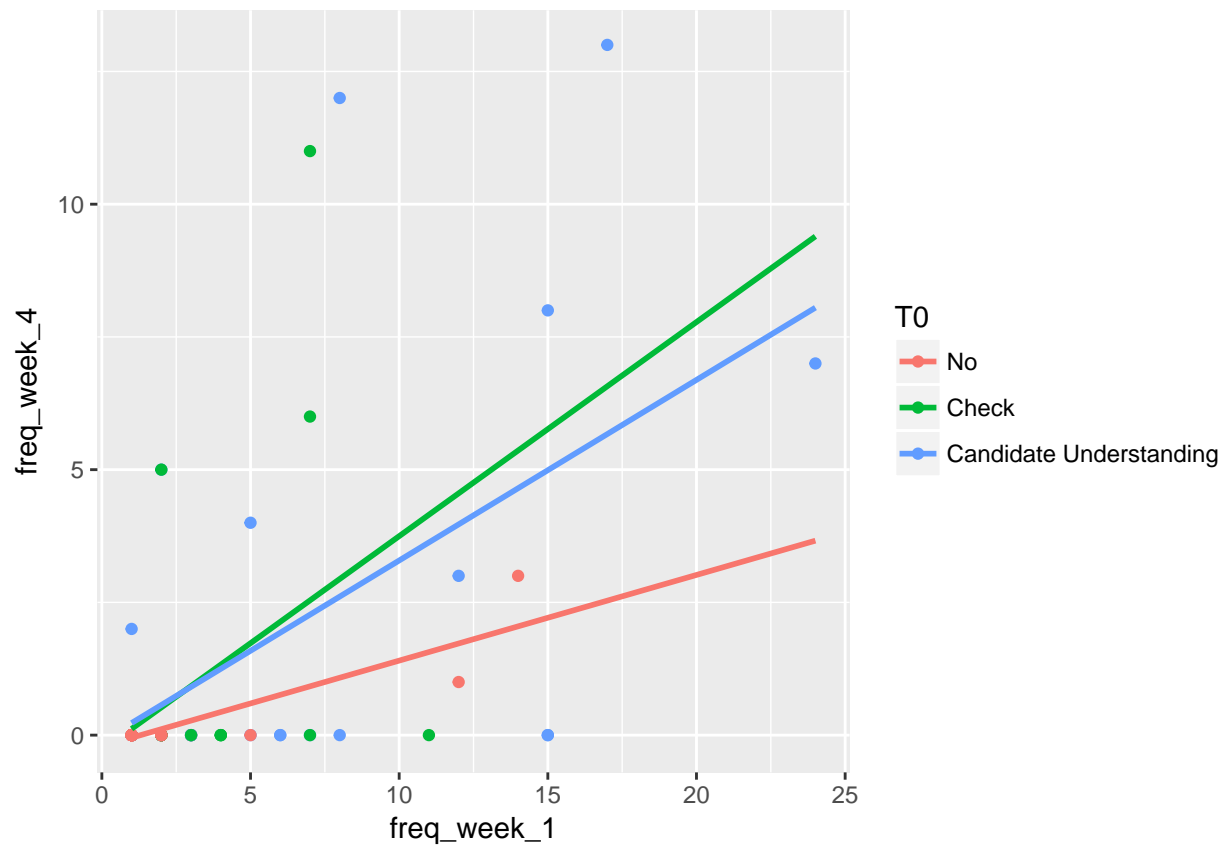
```
## pdf
## 2
```

Plot interactions

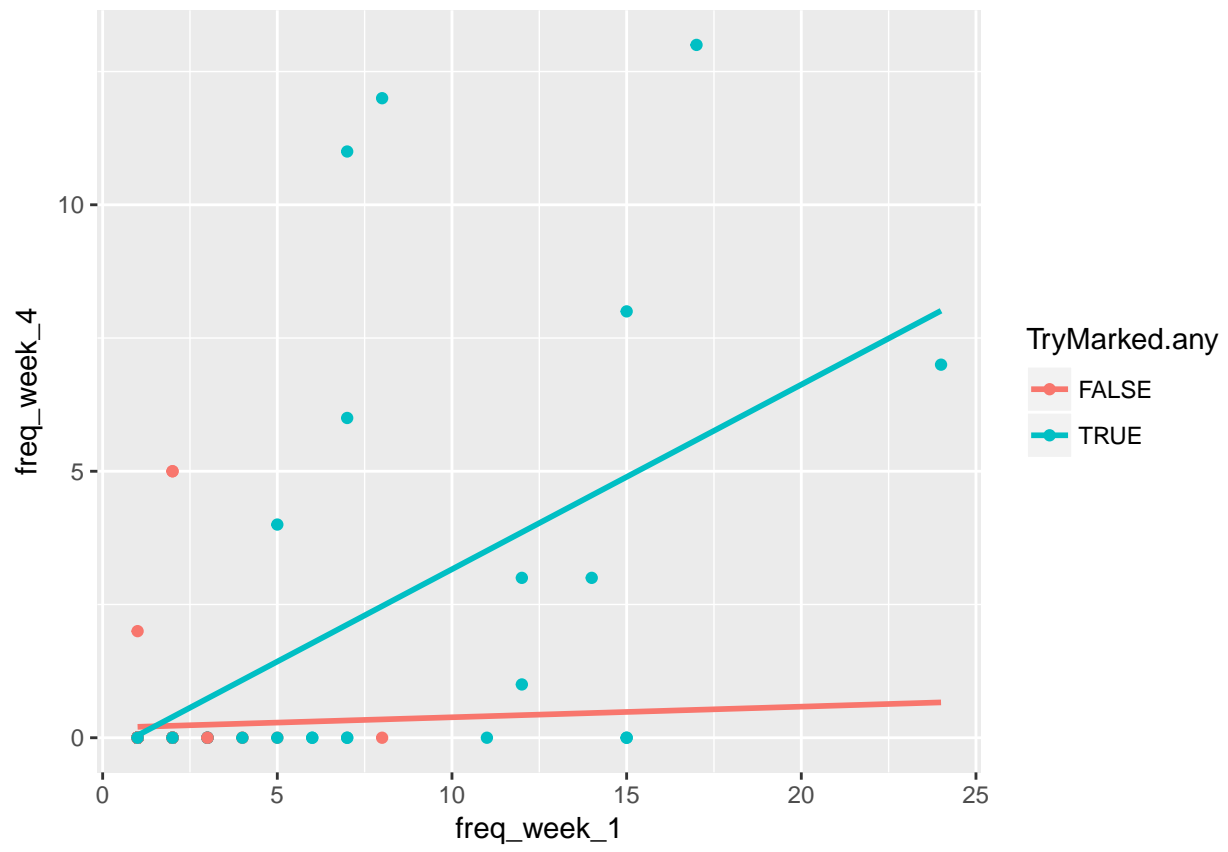
```
ggplot(variants, aes(x=freq_week_1,  
                     y=freq_week_4,  
                     colour=Teach)) +  
  geom_point()+  
  stat_smooth(method = "gam", fullrange = T, se=F)
```



```
ggplot(variants, aes(x=freq_week_1,  
                     y=freq_week_4,  
                     colour=T0)) +  
  geom_point()+  
  stat_smooth(method = "gam", fullrange = T, se=F)
```



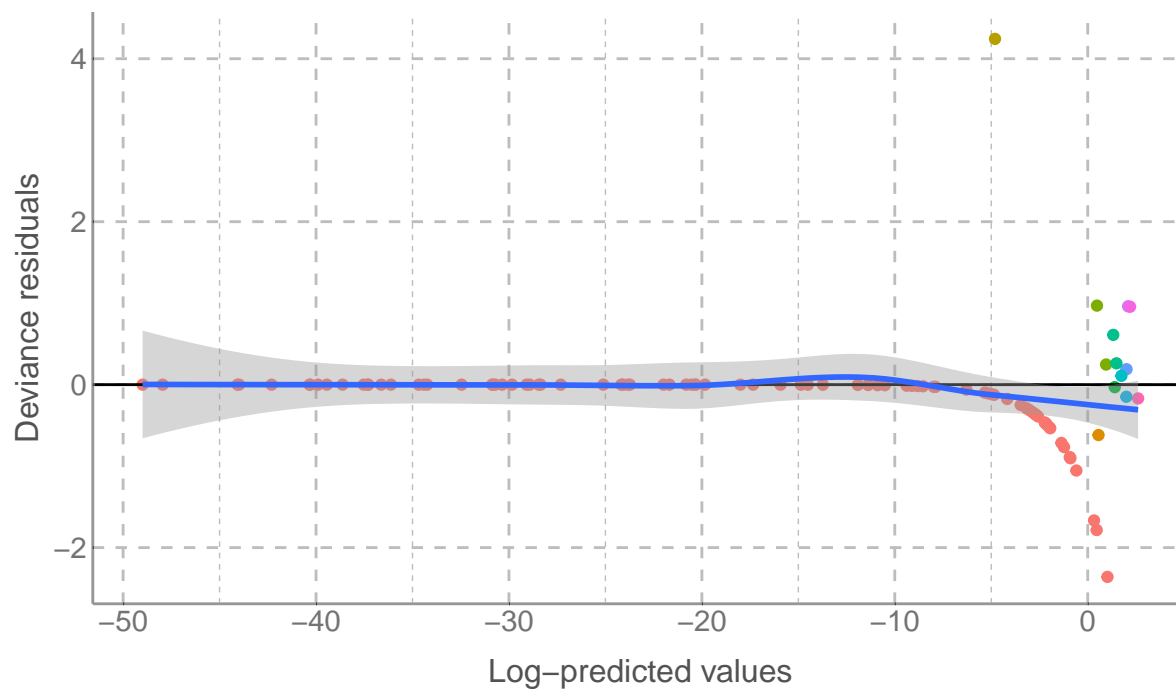
```
ggplot(variants, aes(x=freq_week_1,
                     y=freq_week_4,
                     colour=TryMarked.any)) +
  geom_point()+
  stat_smooth(method = "gam", fullrange = T, se=F)
```



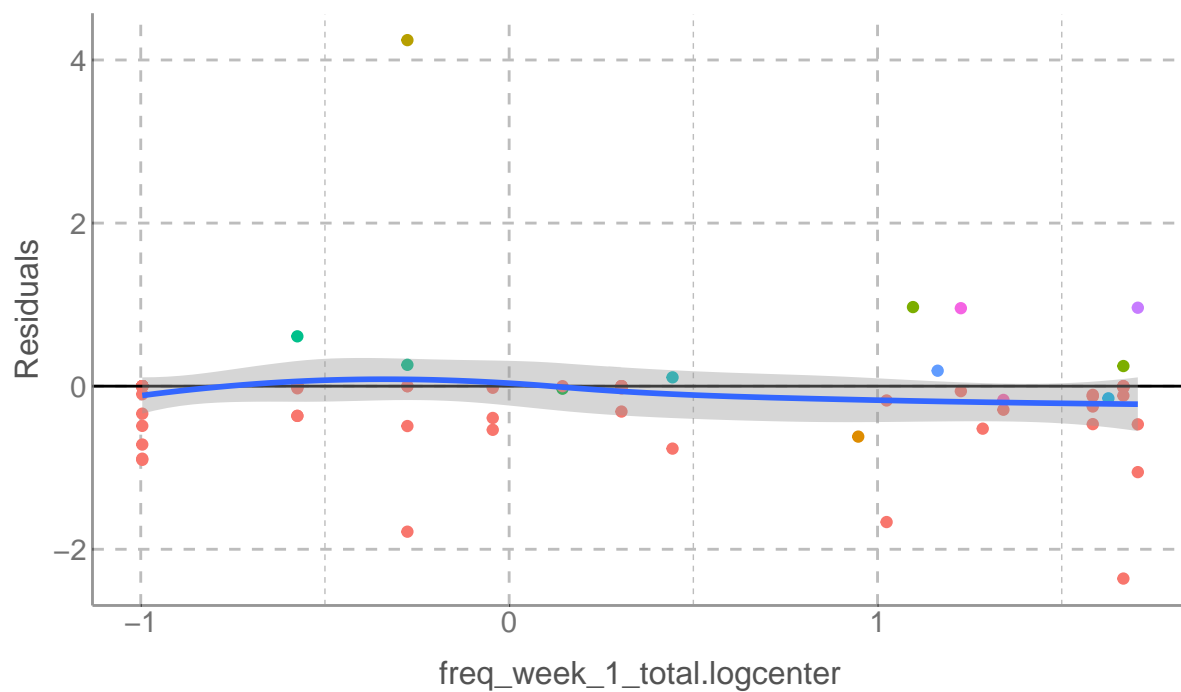
Check model assumptions. The model is not good at predicting higher values.

```
sjp.glmer(finalModel, 'ma')
```

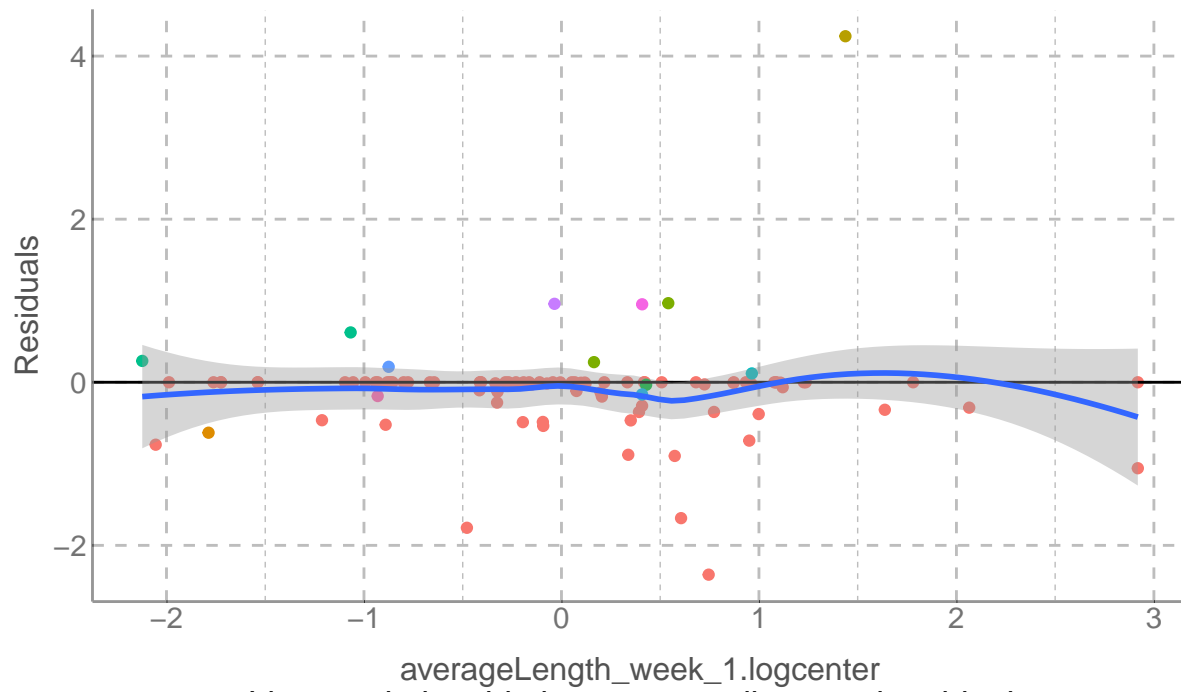
Residual plot (original model)



Linear relationship between predictor and residuals



Linear relationship between predictor and residuals



Linear relationship between predictor and residuals

