2025LMMCourse Session 5 Practical

Fang Yang

2025-05-16

Try to explore syntactic priming effect in the cheese data.

Each time a participant and a confederate took turns to describe pictures to each other. The confederate pretended to be a real participant but read out scripts. A response were coded as 1 if the real participant reused the confederate's sentence structure, otherwise 0. Participants did the experiments online with or without seeing each other's faces (video_call vs audio_call).

0.1 Data loading, cleaning and visualisation

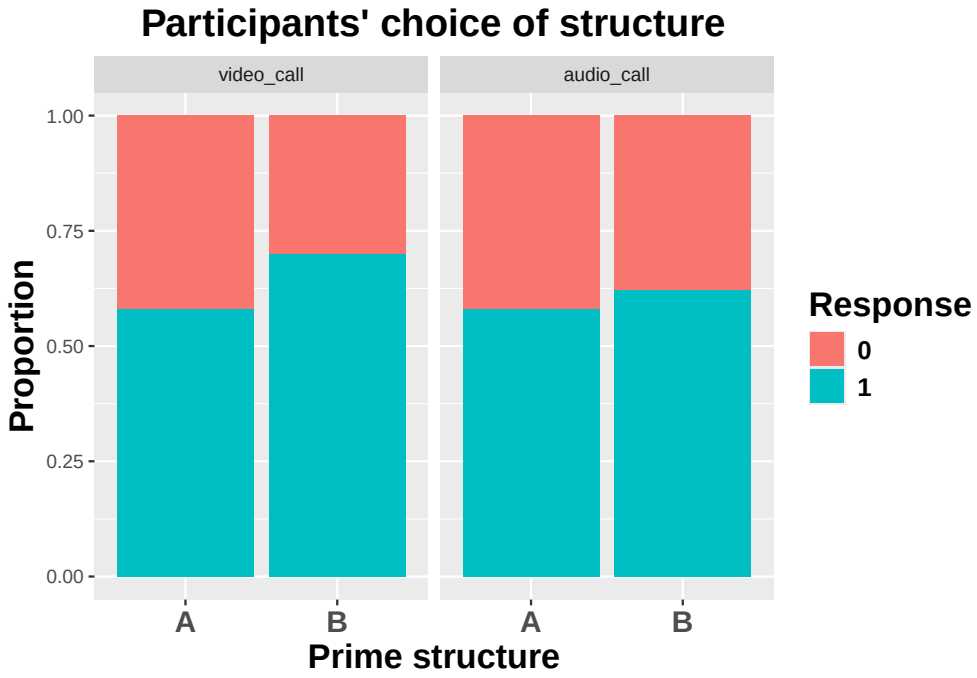
```
data_cheese <- read_csv("dog_man_cheese.csv")
```

```
summary(data_cheese)
```

```
##   participant        item              prime           communication
##  Min.   : 1.00   Length:1728        Length:1728        Length:1728
##  1st Qu.:18.75   Class :character   Class :character   Class :character
##  Median :36.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   :36.50
##  3rd Qu.:54.25
##  Max.   :72.00
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.6208
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :138
```

```
data_cheese$Response<-as.factor(data_cheese$Response)   # 1= aligned; 0 = not aligned


data_cheese$prime <- factor(data_cheese$prime, levels=c("A","B")) # set structure A as reference level

data_cheese$communication <- factor(data_cheese$communication, levels=c("video_call","audio_call"))
```

```r
data_cheese |>
    drop_na() |>
    ggplot(aes( x=prime, fill=Response)) +
  geom_bar(position = "fill") +
  facet_wrap(~communication)+
  ggtitle(label = "Participants' choice of structure")+
  labs(y = "Proportion",
       x= "Prime structure" )+
  theme(plot.title = element_text(hjust = 0.5, size=18,face = "bold"),
        axis.text.x = element_text(size=14,face = "bold"),
        axis.title.x = element_text(size=16,face = "bold"),
        axis.title.y = element_text(size=16,face = "bold"),
        legend.title = element_text(size=16,face = "bold"),
        legend.text = element_text(size=12,face = "bold")
  )
```



0.2 Fit an additive model

Fit an intercept-only model including the following as fixed effects: (1) main effect of prime and (2) main effect of communication.

```r
m_addi_datacheese <- glmer(Response ~ prime + communication+
                       (1|participant) +
                       (1|item),
                     data = data_cheese,
                     family='binomial',
                     na.action=na.exclude,
                     control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))

summary(m_addi_datacheese)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Response ~ prime + communication + (1 | participant) + (1 | item)
##    Data: data_cheese
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1723.4    1750.3    -856.7    1713.4      1585
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0595 -0.6015  0.2864  0.5829  5.1843
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 2.2988   1.5162
##  item        (Intercept) 0.8344   0.9134
## Number of obs: 1590, groups:  participant, 72; item, 36
##
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.5743     0.2598   2.211   0.0271 *
## primeB                    0.5208     0.1274   4.089 4.33e-05 ***
## communicationaudio_call  -0.1880     0.1265  -1.486   0.1372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) primeB
## primeB      -0.224
## cmmnctnd_cl -0.243 -0.010
```

0.3 Fit an interactive model

Fit an intercept-only model including the following as fixed effects: (1) main effect of prime, (2) main effect of communication, and (3) the interaction effect between prime and communication.

```
m_int_datacheese <- glmer(Response ~ prime*communication+
                      (1|participant) +
                      (1|item),
                    data = data_cheese,
                    family='binomial',
                    na.action=na.exclude,
                    control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))

summary(m_int_datacheese)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Response ~ prime * communication + (1 | participant) + (1 | item)
##    Data: data_cheese
```

```
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1719.9    1752.2    -854.0    1707.9      1584
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4368 -0.6027  0.2799  0.5921  5.7209
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 2.337    1.5287
##  item        (Intercept) 0.856    0.9252
## Number of obs: 1590, groups:  participant, 72; item, 36
##
## Fixed effects:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.4311     0.2685   1.606   0.1083
## primeB                           0.8321     0.1839   4.524 6.07e-06 ***
## communicationaudio_call          0.1048     0.1767   0.593   0.5532
## primeB:communicationaudio_call  -0.6076     0.2561  -2.372   0.0177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) primeB cmmnc_
## primeB      -0.306
## cmmnctnd_cl -0.321  0.485
## prmB:cmmnc_  0.221 -0.719 -0.696
```

0.4 Model comparison

```
anova(m_addi_datacheese, m_int_datacheese)
```

```
## Data: data_cheese
## Models:
## m_addi_datacheese: Response ~ prime + communication + (1 | participant) + (1 | item)
## m_int_datacheese: Response ~ prime * communication + (1 | participant) + (1 | item)
##                   npar    AIC    BIC  logLik -2*log(L)  Chisq Df Pr(>Chisq)
## m_addi_datacheese    5 1723.4 1750.3 -856.70    1713.4
## m_int_datacheese     6 1719.9 1752.2 -853.97    1707.9 5.4742  1     0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

0.5 Interpret the results

```
summary(m_int_datacheese)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Response ~ prime * communication + (1 | participant) + (1 | item)
##    Data: data_cheese
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1719.9    1752.2    -854.0    1707.9      1584
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4368 -0.6027  0.2799  0.5921  5.7209
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 2.337    1.5287
##  item        (Intercept) 0.856    0.9252
## Number of obs: 1590, groups:  participant, 72; item, 36
##
## Fixed effects:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.4311     0.2685   1.606   0.1083
## primeB                           0.8321     0.1839   4.524 6.07e-06 ***
## communicationaudio_call          0.1048     0.1767   0.593   0.5532
## primeB:communicationaudio_call  -0.6076     0.2561  -2.372   0.0177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) primeB cmmnc_
## primeB      -0.306
## cmmnctnd_cl -0.321  0.485
## prmB:cmmnc_  0.221 -0.719 -0.696
```

Hint: remember the coefficients are log odds. you need to transfer them to odds and then probability

0.5.1 Convert log odds to odds

0.5.2 (1) Probability of producing B in video+primeA condition (intercept)

```
Prob_A_video <- exp(0.43)/(exp(0.43)+1)
Prob_A_video
```

```
## [1] 0.6058737
```

0.5.3 (2) Probability of producing B in video+primeB condition(intercept + slope)

```
Prob_B_video <- exp(1.26)/(exp(1.26)+1)
Prob_B_video
```

```
## [1] 0.7790261
```

0.5.4 (3) Probability of producing B in audio+primeA condition (intercept + slope)

```
Prob_A_audio <- exp(0.53)/(exp(0.53)+1)
Prob_A_audio
```

```
## [1] 0.6294831
```

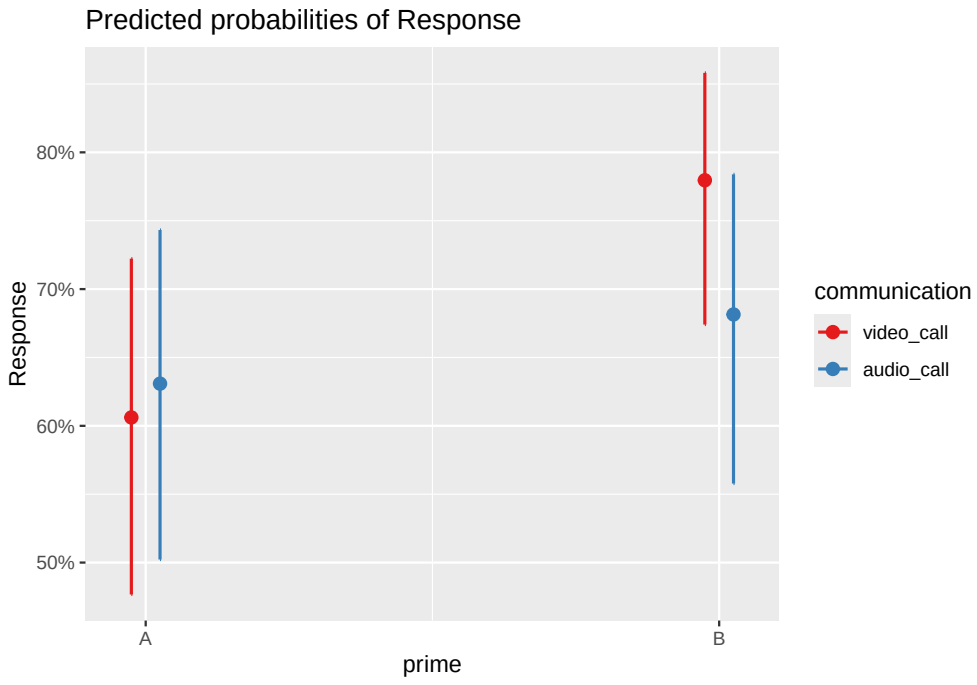
0.5.5 (4)Probability of producing B in audio+primeB condition (intercept + slope)

```
Prob_B_audio <- exp(0.65)/(exp(0.65)+1)
Prob_B_audio
```

```
## [1] 0.6570105
```

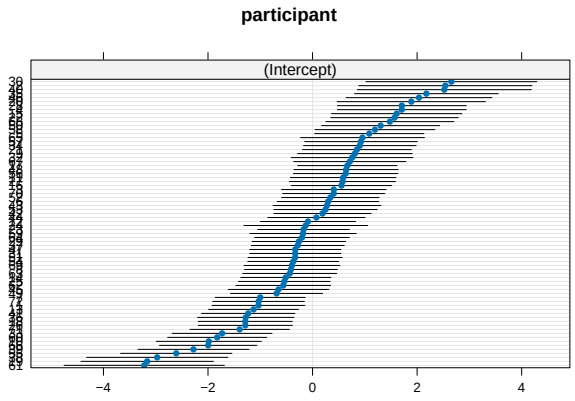
0.6 Visulise the Model

```
# fixed effects
plot_model(m_int_datacheese, type = "int")
```

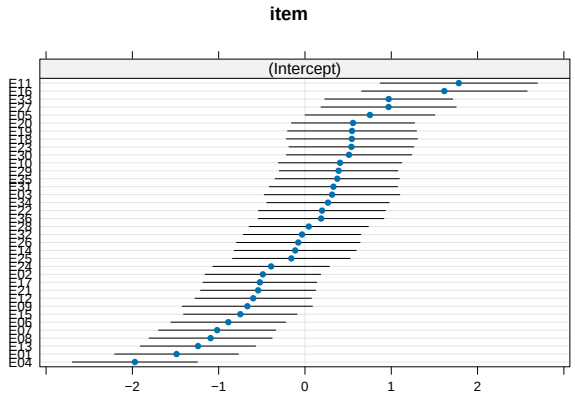



```
# random effects
randoms_cheese <- ranef(m_int_datacheese, condVar=TRUE)
dotplot.ranef.mer(randoms_cheese)
```

```
## $participant
```




```
##
## $item
```



0.7 Check and plot random effects

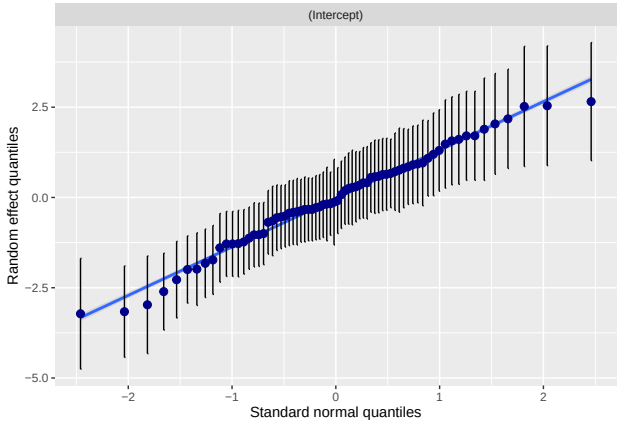
```
### to check the random effect
ranef(m_int_datacheese)

### to check the fixed effect
fixef(m_int_datacheese)

### group level coefficient
coef(m_int_datacheese)
```

0.8 Check Model Assumptions

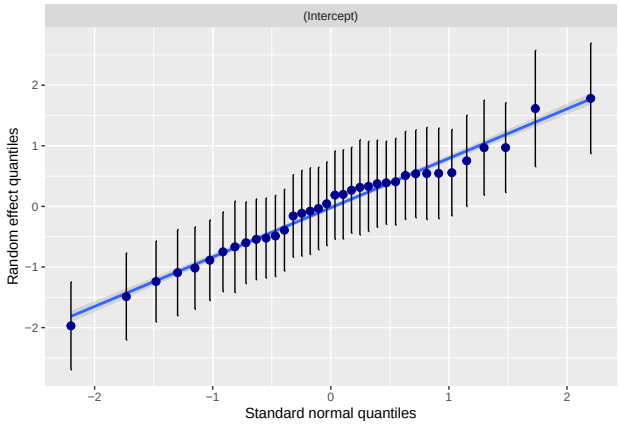
For generalized linear mixed models, returns the QQ-plot for random effects.

```
sjPlot::plot_model(m_int_datacheese, type = "diag")
```

```
## $participant
```


(Intercept)
Random effect quantiles
2.5
0.0
−2.5
−5.0
−2
−1
0
1
2
Standard normal quantiles

```
##
## $item
```


(Intercept)
Random effect quantiles
2
1
0
−1
−2
−2
−1
0
1
2
Standard normal quantiles