Supplement

1 Read data

```
gestures <- read_csv("./data/gestures.csv")

gestures_tot <- gestures %>%
  group_by(dyad, background, months, gesture) %>%
  summarise(
    count = sum(count),
    ct = sum(ct)
  ) %>%
  ungroup() %>%
  mutate(
    gesture = factor(gesture, levels = c("reach", "point", "ho_gv"))
  ) %>%
  mutate_if(is.character, as.factor) %>%
  mutate(
    back_o = ordered(background, levels = c("English", "Bangladeshi", "Chinese"))
  )

contrasts(gestures_tot$back_o) <- "contr.treatment"

utterances <- read_csv("./data/utterances.csv")

# NAs not removed in sum()
utterances_tot <- utterances %>%
  group_by(dyad, background, months) %>%
  summarise(
    utterances = sum(utterances)
  ) %>%
  ungroup() %>%
  mutate_if(is.character, as.factor) %>%
  mutate(
    back_o = ordered(background, levels = c("English", "Bangladeshi", "Chinese"))
  )

contrasts(utterances_tot$back_o) <- "contr.treatment"

hg_tot <- filter(gestures_tot, gesture == "ho_gv")
reach_tot <- filter(gestures_tot, gesture == "reach")
point_tot <- filter(gestures_tot, gesture == "point")
all_tot <- gestures_tot %>%
  group_by(dyad, back_o, months) %>%
  summarise(count = sum(count), ct = sum(ct))

hg_point_lead <- gestures_tot %>%
  dplyr::select(-ct) %>%
  spread(gesture, count) %>%
  dplyr::select(-reach) %>%
  group_by(dyad) %>%
```

```
  mutate(
    lead_point = lead(point)
  ) %>%
  filter(months != 12)

reach_point_lead <- gestures_tot %>%
  dplyr::select(-ct) %>%
  spread(gesture, count) %>%
  dplyr::select(-ho_gv) %>%
  group_by(dyad) %>%
  mutate(
    lead_point = lead(point)
  ) %>%
  filter(months != 12)

reach_point_lead <- gestures_tot %>%
  dplyr::select(-ct) %>%
  spread(gesture, count) %>%
  dplyr::select(-ho_gv) %>%
  group_by(dyad) %>%
  mutate(
    lead_point = lead(point)
  ) %>%
  filter(months != 12)

ct_point_lead <- gestures_tot %>%
  filter(gesture == "point") %>%
  dplyr::select(-gesture) %>%
  rename(point = count) %>%
  group_by(dyad) %>%
  mutate(
    lead_point = lead(point)
  ) %>%
  filter(months != 12)

utter_point_lead <- gestures_tot %>%
  filter(gesture == "point") %>%
  right_join(y = utterances_tot) %>%
  group_by(dyad) %>%
  mutate(
    lead_point = lead(count)
  ) %>%
  filter(months != 12)

pointing <- gestures_tot %>%
  dplyr::select(-ct) %>%
  spread(gesture, count)

gestures <- read_csv("./data/gestures.csv")

gestures_tot_2 <- gestures %>%
  group_by(dyad, background, months, gesture) %>%
  summarise(
```

```
    count = sum(count),
    ct = sum(ct)
  ) %>%
  ungroup() %>%
  mutate(
    gesture = factor(gesture, levels = c("reach", "point", "ho_gv"))
  ) %>%
  mutate_if(is.character, as.factor)

point <- gestures %>%
  filter(gesture == "point") %>%
  group_by(dyad, background, months) %>%
  summarise(
    count = sum(count)
  ) %>%
  ungroup() %>%
  mutate_if(is.character, as.factor)

utter_point <- left_join(utterances_tot, point) %>%
  rename(point = count)

utterances_compl <- utterances %>% na.omit()
utterances_tcompl <- utterances_tot %>% na.omit()

hgp_tot <- gestures_tot_2 %>%
  filter(gesture != "reach") %>%
  group_by(dyad, background) %>%
  summarise(hgp_tot = sum(count))

reach_tot_2 <- gestures_tot_2 %>%
  filter(gesture == "reach") %>%
  group_by(dyad, background) %>%
  summarise(reach_tot = sum(count))

vocab_gest <- gestures_tot_2 %>%
  group_by(dyad, background) %>%
  summarise(count_tot = sum(count), ct_tot = sum(ct)) %>%
  full_join(y = hgp_tot) %>%
  full_join(y = reach_tot_2)

vocab_utt <- utterances_tot %>%
  group_by(dyad, background) %>%
  summarise(utt_tot = sum(utterances))

vocab <- read_csv("./data/vocab.csv") %>%
  full_join(y = vocab_gest) %>%
  full_join(y = vocab_utt) %>%
  arrange(dyad, months) %>%
  mutate_if(is.character, as.factor)
```

2 Analysis 1a. The development of reaches, hold out and gives (HoGs), and points from 10-12 months.

2.1 Reaches development

```
reach_nb <- glm.nb(count ~ months, data = reach_tot)

reach_gam <- gam(
  count ~
    back_o +
    s(months, k = 3) +
    s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = reach_tot,
  method = "ML",
  family = negbin(0.986)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(reach_gam)
```

```
##
## Family: Negative Binomial(0.986)
## Link function: log
##
## Formula:
## count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Parametric coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.6377     0.1920   3.322 0.000895 ***
## back_oBangladeshi   0.5873     0.2601   2.258 0.023930 *
## back_oChinese       0.2402     0.2650   0.906 0.364737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                edf  Ref.df Chi.sq p-value
## s(months)                    1.155   1.286  1.181  0.2854
## s(months):back_oBangladeshi  1.000   1.000  0.437  0.5086
## s(months):back_oChinese      1.000   1.000  0.125  0.7237
## s(months,dyad)              14.509 112.000 20.040  0.0316 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.165   Deviance explained = 21.4%
## -ML = 378.53  Scale est. = 1         n = 173
```

```
reach_gam_null <- gam(
  count ~
    # back_o +
```

```
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = reach_tot,
  method = "ML",
  family = negbin(0.986)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(reach_gam_null, reach_gam)
```

```
## reach_gam_null: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## reach_gam: count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##            Model    Score Edf Difference    Df p.value Sig.
## 1 reach_gam_null 381.3264   5
## 2      reach_gam 378.5345  11      2.792 6.000   0.471
##
## AIC difference: -1.91, model reach_gam_null has lower AIC.
```

```
## Warning in compareML(reach_gam_null, reach_gam): Only small difference in ML...
```

```
reach_gam_2 <- gam(
  count ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = reach_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

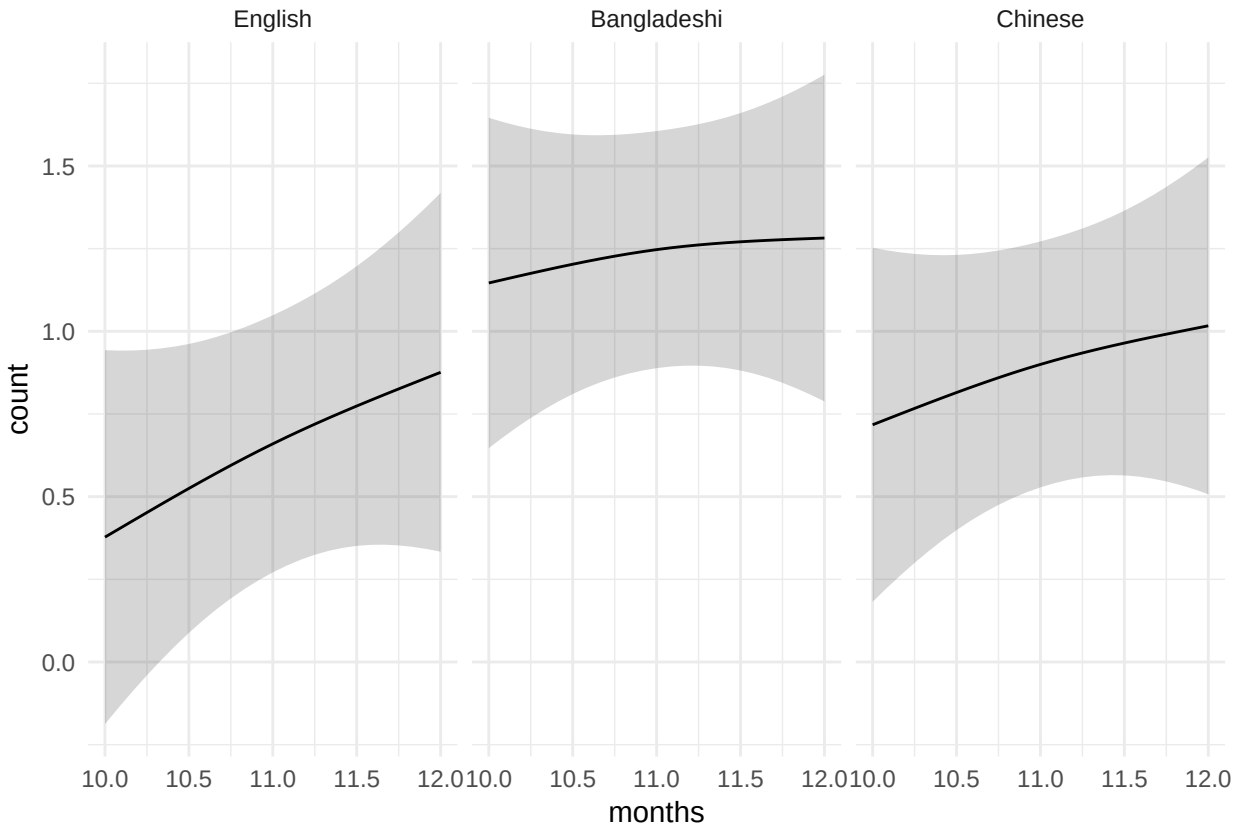
```
reach_gam_2_null <- gam(
  count ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = reach_tot,
  method = "ML",
  family = negbin(0.1946)
)
compareML(reach_gam_2_null, reach_gam_2)
```

```
## reach_gam_2_null: count ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## reach_gam_2: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
```

```
## -----
##               Model    Score Edf Difference    Df p.value Sig.
## 1 reach_gam_2_null 431.9091   3
## 2      reach_gam_2 431.7370   5      0.172 2.000   0.842
##
## AIC difference: -1.66, model reach_gam_2_null has lower AIC.
```

```
## Warning in compareML(reach_gam_2_null, reach_gam_2): Only small difference in ML...
```

```
plot_smooths(reach_gam, months, facet_terms = back_o, series_length = 25)
```



2.2 HGs development

```
hg_nb <- glm.nb(count ~ months, data = hg_tot)

hg_gam <- gam(
  count ~
    back_o +
    s(months, k = 3) +
    s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = hg_tot,
  method = "ML",
  family = negbin(0.6434)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
```

```
## repeated 1-d smooths of same variable.
```

```
summary(hg_gam)
```

```
##
## Family: Negative Binomial(0.643)
## Link function: log
##
## Formula:
## count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Parametric coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.7491     0.2316   3.234  0.00122 **
## back_oBangladeshi   0.9117     0.3143   2.901  0.00372 **
## back_oChinese       0.7257     0.3163   2.295  0.02176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf Ref.df Chi.sq p-value
## s(months)                    1.0      1  9.707 0.00184 **
## s(months):back_oBangladeshi  1.0      1  0.025 0.87559
## s(months):back_oChinese      1.0      1  0.426 0.51391
## s(months,dyad)              17.7    112 26.330 0.01075 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.335   Deviance explained = 38.5%
## -ML = 451.06  Scale est. = 1         n = 173
```

```
hg_gam_null <- gam(
  count ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = hg_tot,
  method = "ML",
  family = negbin(0.6434)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(hg_gam_null, hg_gam)
```

```
## hg_gam_null: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## hg_gam: count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
```

```
##         Model    Score Edf Difference    Df p.value Sig.
## 1 hg_gam_null 455.3697   5
## 2      hg_gam 451.0601  11      4.310 6.000   0.196
##
## AIC difference: -2.20, model hg_gam_null has lower AIC.
```

```
## Warning in compareML(hg_gam_null, hg_gam): Only small difference in ML...
```

```
hg_gam_2 <- gam(
  count ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = hg_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

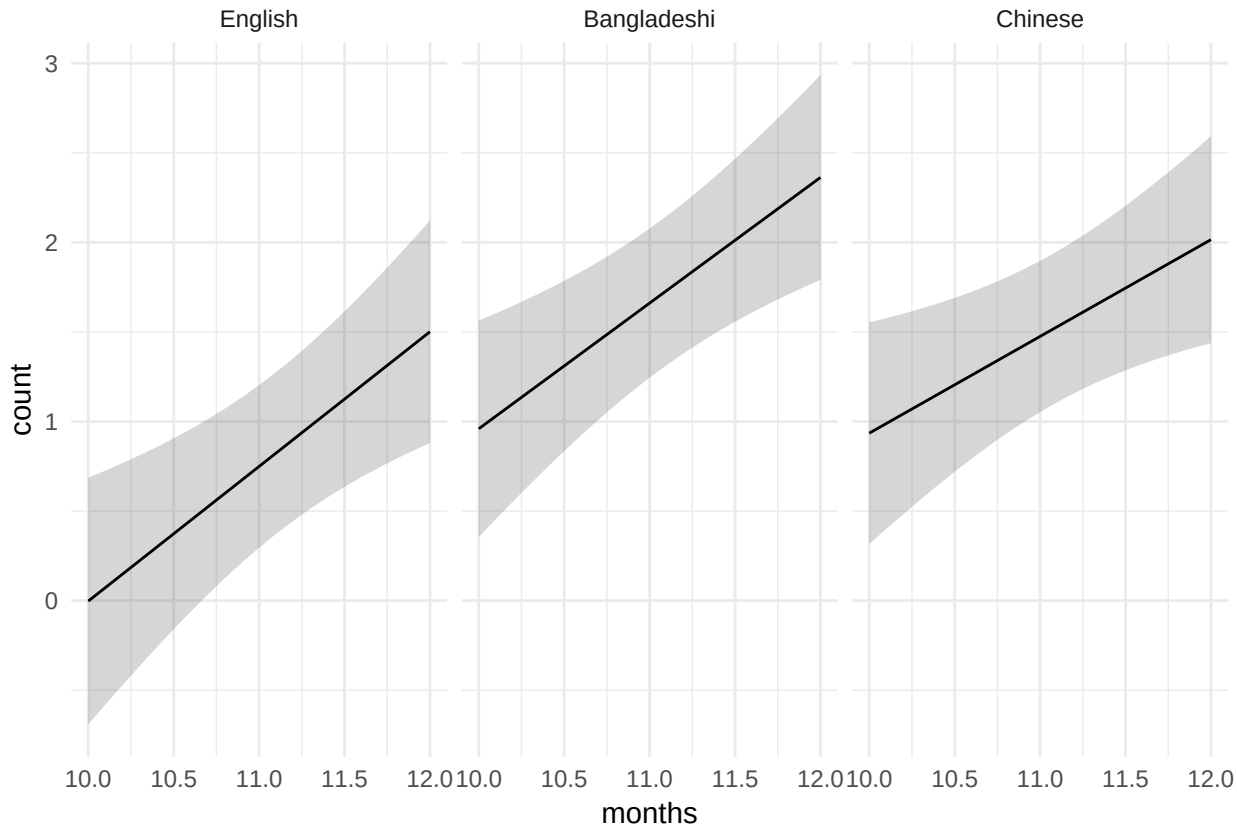
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
hg_gam_2_null <- gam(
  count ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = hg_tot,
  method = "ML",
  family = negbin(0.1946)
)
compareML(hg_gam_2_null, hg_gam_2)
```

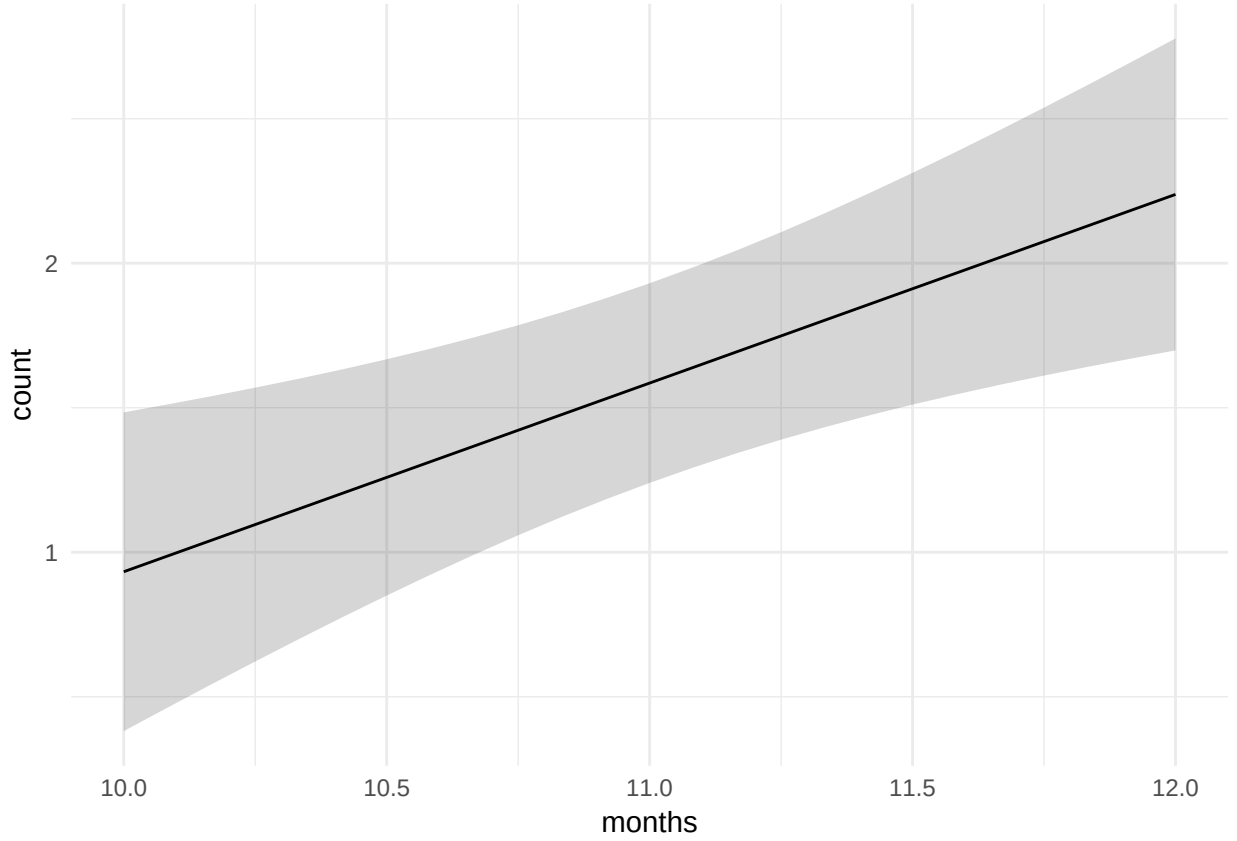
```
## hg_gam_2_null: count ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## hg_gam_2: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
## -----
##           Model    Score Edf Difference    Df p.value Sig.
## 1 hg_gam_2_null 501.9866   3
## 2      hg_gam_2 497.7652   5      4.221 2.000   0.015   *
##
## AIC difference: 6.44, model hg_gam_2 has lower AIC.
```

```
## Warning in compareML(hg_gam_2_null, hg_gam_2): Only small difference in ML...
```

```
plot_smooths(hg_gam, months, facet_terms = back_o, series_length = 25)
```




```
plot_smooths(hg_gam_2, months, series_length = 25)
```



2.3 Points development

```
point_nb <- glm.nb(count ~ months, data = point_tot)

point_gam <- gam(
  count ~
    back_o +
    s(months, k = 3) +
    s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(point_gam)
```

```
##
## Family: Negative Binomial(0.195)
## Link function: log
##
## Formula:
## count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Parametric coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.6917     0.3953   1.750   0.0802 .
## back_oBangladeshi  -0.4993     0.5588  -0.894   0.3716
## back_oChinese      -0.5735     0.5675  -1.011   0.3122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                edf  Ref.df Chi.sq p-value
## s(months)                    1.000   1.000  1.068  0.3014
## s(months):back_oBangladeshi  1.538   1.786  0.726  0.5736
## s(months):back_oChinese      1.000   1.000  2.118  0.1456
## s(months,dyad)              18.373 112.000 26.009  0.0224 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.332   Deviance explained =   41%
## -ML = 326.23  Scale est. = 1         n = 173
```

```
point_gam_null <- gam(
  count ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
```

```
  method = "ML",
  family = negbin(0.1946)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(point_gam_null, point_gam)
```

```
## point_gam_null: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## point_gam: count ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##            Model    Score Edf Difference    Df p.value Sig.
## 1 point_gam_null 327.9346   5
## 2      point_gam 326.2345  11      1.700 6.000   0.757
##
## AIC difference: -7.40, model point_gam_null has lower AIC.
```

```
## Warning in compareML(point_gam_null, point_gam): Only small difference in ML...
```

```
point_gam_2 <- gam(
  count ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

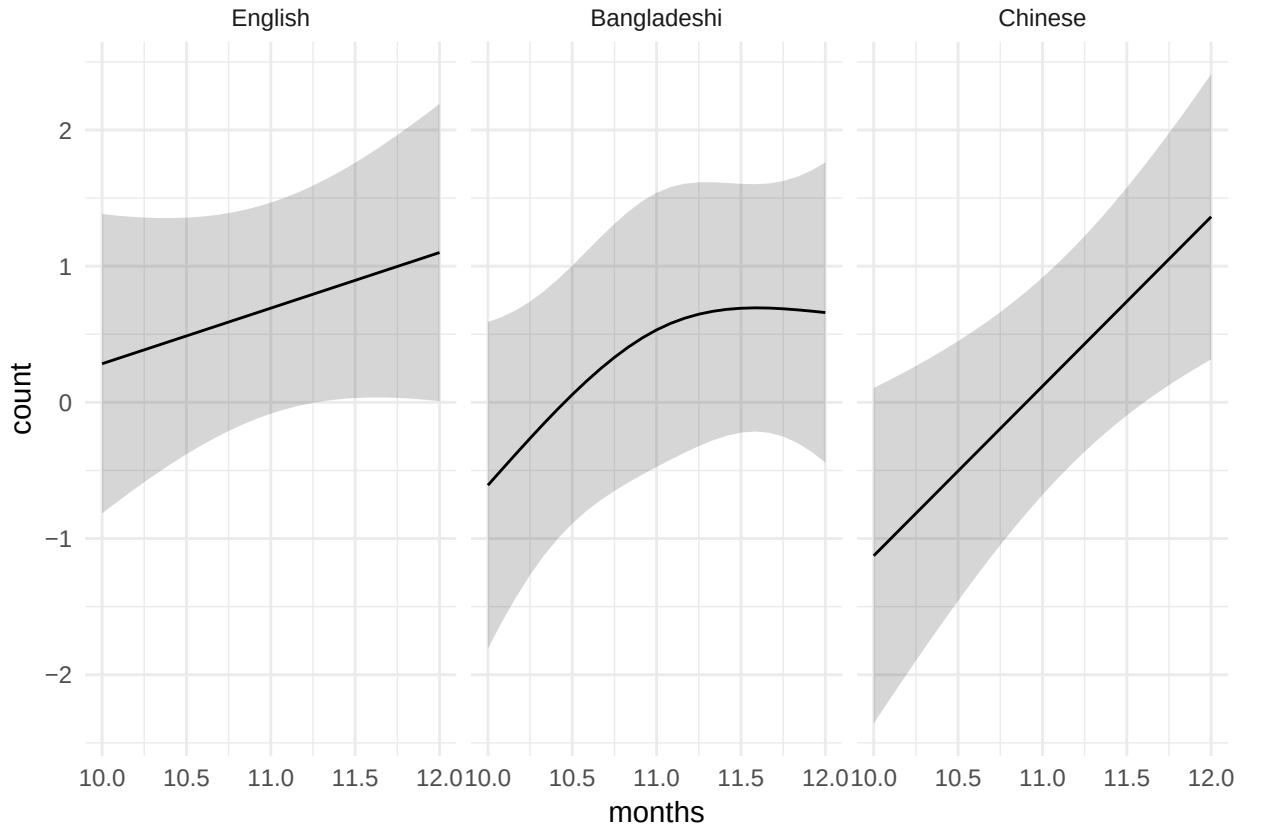
```
point_gam_2_null <- gam(
  count ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
  method = "ML",
  family = negbin(0.1946)
)
compareML(point_gam_2_null, point_gam_2)
```

```
## point_gam_2_null: count ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## point_gam_2: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
## -----
##              Model    Score Edf Difference    Df p.value Sig.
## 1 point_gam_2_null 332.5507   3
## 2      point_gam_2 327.9346   5      4.616 2.000   0.010  **
```

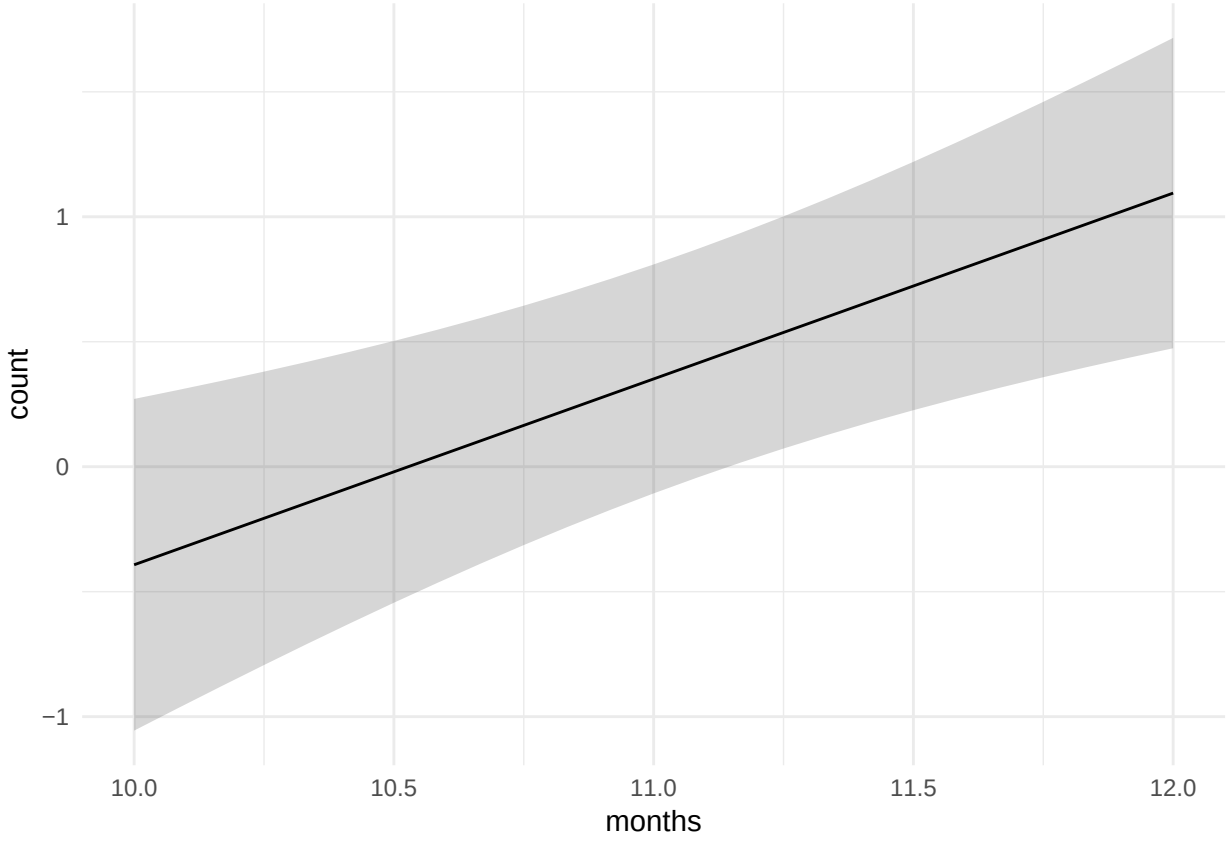
```
##
## AIC difference: 10.13, model point_gam_2 has lower AIC.
```

```
## Warning in compareML(point_gam_2_null, point_gam_2): Only small difference in ML...
```

```
plot_smooths(point_gam, months, facet_terms = back_o, series_length = 25)
```




```
plot_smooths(point_gam_2, months, series_length = 25)
```



3 Analysis 1b. Frequency of maternal utterances and contingent talk to infants aged 10-12 months.

3.1 Maternal utterances development

```
utter_gam <- gam(
  utterances ~
    back_o +
    s(months, k = 3) +
    s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML"
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(utter_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
```

```
## utterances ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         284.44      27.10  10.494   <2e-16 ***
## back_oBangladeshi   -65.59      37.82  -1.734   0.0865 .
## back_oChinese       -37.80      37.74  -1.002   0.3193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                edf  Ref.df     F p-value
## s(months)                    1.693   1.880 0.966   0.333
## s(months):back_oBangladeshi  1.001   1.001 1.065   0.305
## s(months):back_oChinese      1.334   1.533 1.924   0.107
## s(months,dyad)              73.930 111.000 7.087  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.837   Deviance explained = 91.6%
## -ML = 991.97  Scale est. = 2827.4    n = 167
```

```
utter_gam_null <- gam(
  utterances ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML"
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(utter_gam_null, utter_gam)
```

```
## utter_gam_null: utterances ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## utter_gam: utterances ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##            Model    Score Edf Difference    Df p.value Sig.
## 1 utter_gam_null 995.3291   5
## 2      utter_gam 991.9724  11      3.357 6.000   0.348
##
## AIC difference: -3.68, model utter_gam_null has lower AIC.
```

```
## Warning in compareML(utter_gam_null, utter_gam): Only small difference in ML...
```

```
utter_gam_2 <- gam(
  utterances ~
```

```
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

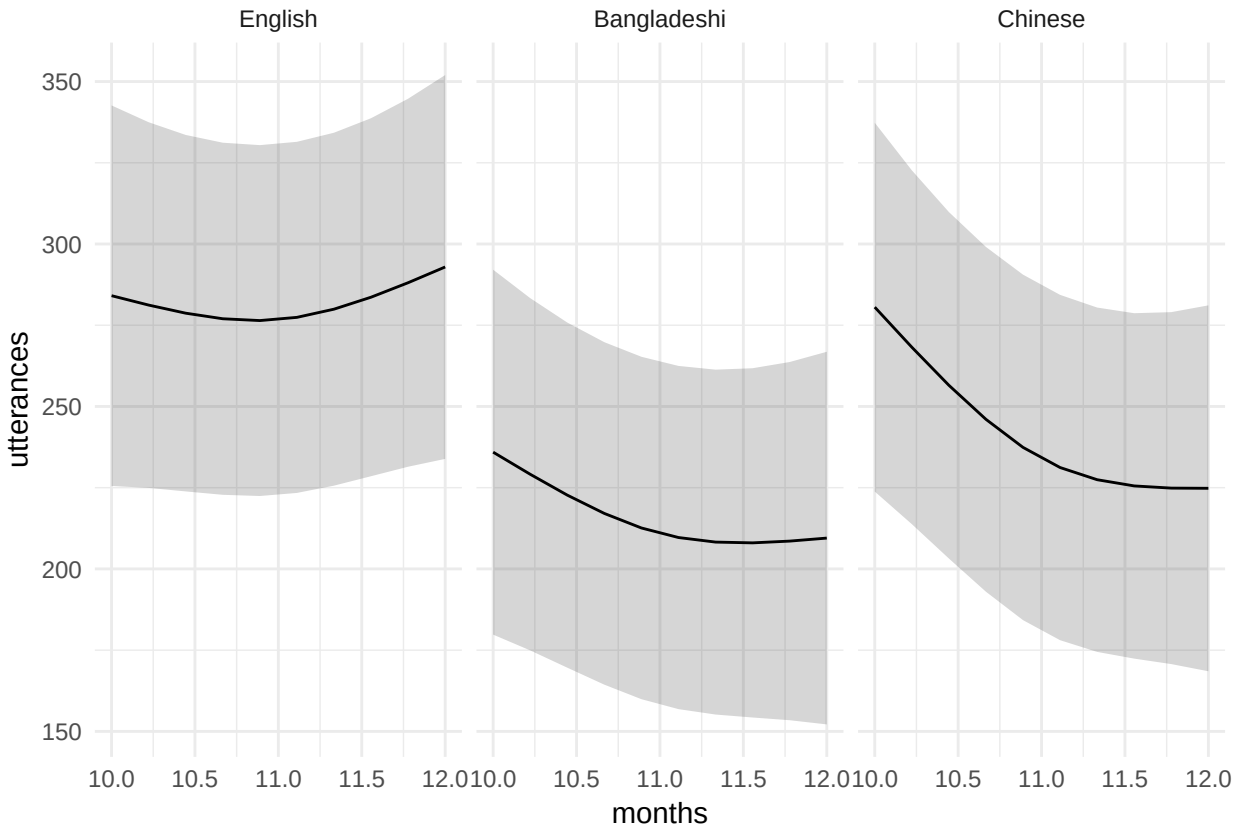
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
utter_gam_2_null <- gam(
  utterances ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML",
  family = negbin(0.1946)
)
compareML(utter_gam_2_null, utter_gam_2)
```

```
## utter_gam_2_null: utterances ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## utter_gam_2: utterances ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
## -----
##              Model    Score Edf Difference    Df p.value Sig.
## 1 utter_gam_2_null 1236.874   3
## 2      utter_gam_2 1236.819   5      0.055 2.000   0.946
##
## AIC difference: -1.89, model utter_gam_2_null has lower AIC.
```

```
## Warning in compareML(utter_gam_2_null, utter_gam_2): Only small difference in ML...
```

```
plot_smooths(utter_gam, months, facet_terms = back_o, series_length = 10)
```



3.2 Contingent talks development

```
ct_gam <- gam(
  ct ~
    back_o +
    s(months, k = 3) +
    s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = all_tot,
  method = "ML"
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(ct_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ct ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         3.6440     0.9612   3.791 0.000258 ***
## back_oBangladeshi  -2.4920     1.3422  -1.857 0.066340 .
## back_oChinese      -1.3008     1.3439  -0.968 0.335413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                edf  Ref.df     F p-value
## s(months)                    1.000   1.000 8.748 0.00385 **
## s(months):back_oBangladeshi  1.699   1.909 1.777 0.12264
## s(months):back_oChinese      1.000   1.000 0.847 0.35963
## s(months,dyad)              66.372 112.000 5.607 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.805   Deviance explained = 88.7%
## -ML = 453.21  Scale est. = 4.5108    n = 172
```

```
ct_gam_null <- gam(
  ct ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = all_tot,
  method = "ML"
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(ct_gam_null, ct_gam)
```

```
## ct_gam_null: ct ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs", m = 1)
##
## ct_gam: ct ~ back_o + s(months, k = 3) + s(months, k = 3, by = back_o) +
##     s(months, dyad, k = 2, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##         Model    Score Edf Difference    Df p.value Sig.
## 1 ct_gam_null 458.3513   5
## 2      ct_gam 453.2132  11      5.138 6.000   0.113
##
## AIC difference: -0.76, model ct_gam_null has lower AIC.
```

```
ct_gam_2 <- gam(
  count ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = all_tot,
  method = "ML",
  family = negbin(0.1946)
)
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
```

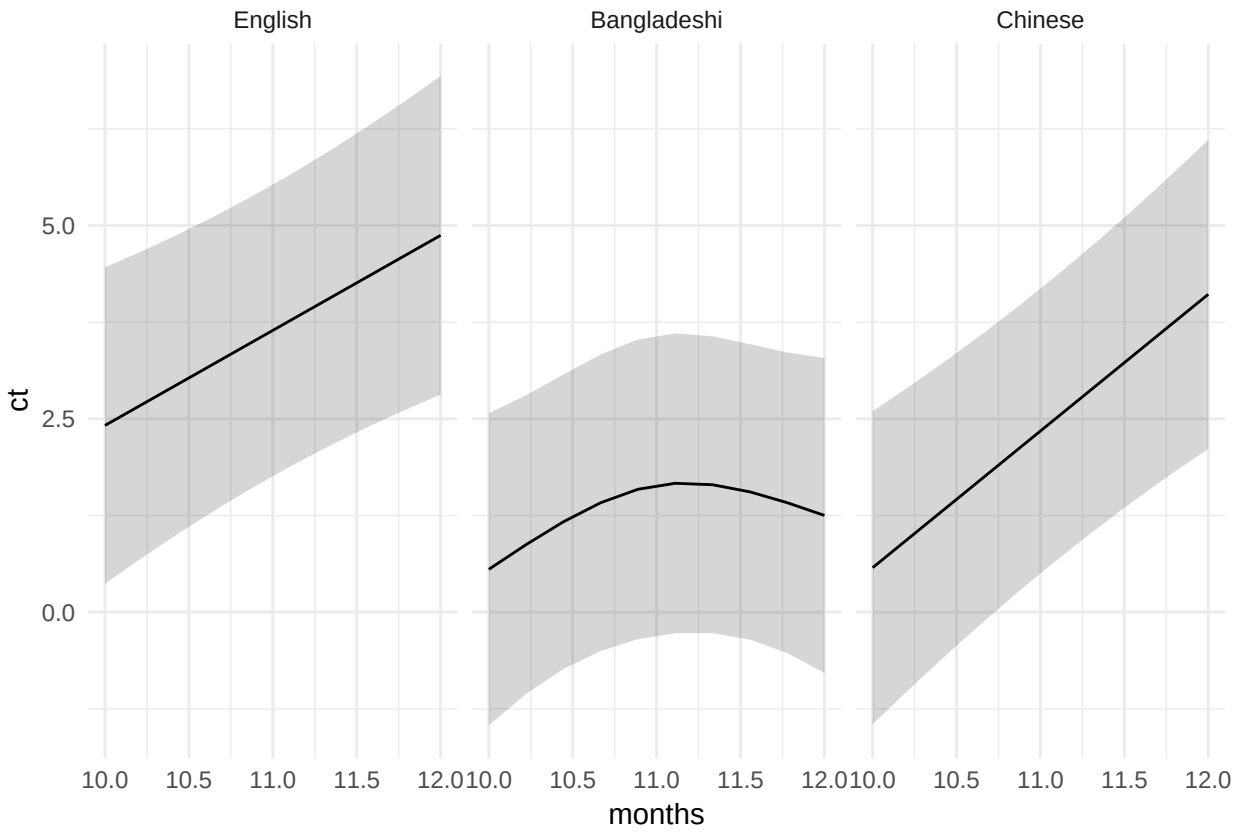
```
## repeated 1-d smooths of same variable.
```

```
ct_gam_2_null <- gam(
  count ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = all_tot,
  method = "ML",
  family = negbin(0.1946)
)
compareML(ct_gam_2_null, ct_gam_2)
```

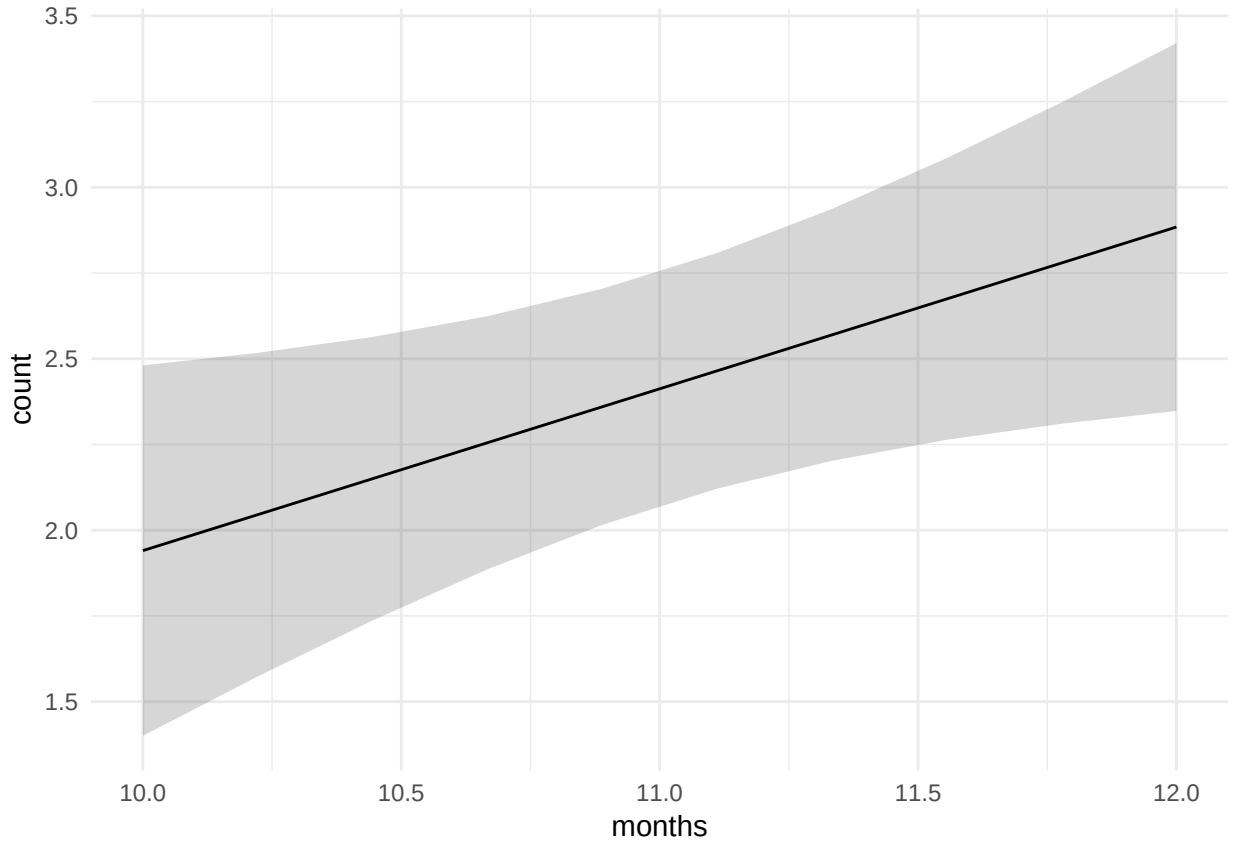
```
## ct_gam_2_null: count ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## ct_gam_2: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
## -----
##           Model    Score Edf Difference    Df p.value Sig.
## 1 ct_gam_2_null 696.0918   3
## 2      ct_gam_2 693.7841   5      2.308 2.000   0.099
##
## AIC difference: 2.62, model ct_gam_2 has lower AIC.
```

```
## Warning in compareML(ct_gam_2_null, ct_gam_2): Only small difference in ML...
```

```
plot_smooths(ct_gam, months, facet_terms = back_o, series_length = 10)
```



```
plot_smooths(ct_gam_2, months, series_length = 10)
```



4 Analysis 1c. Predictors of pointing at 12 months

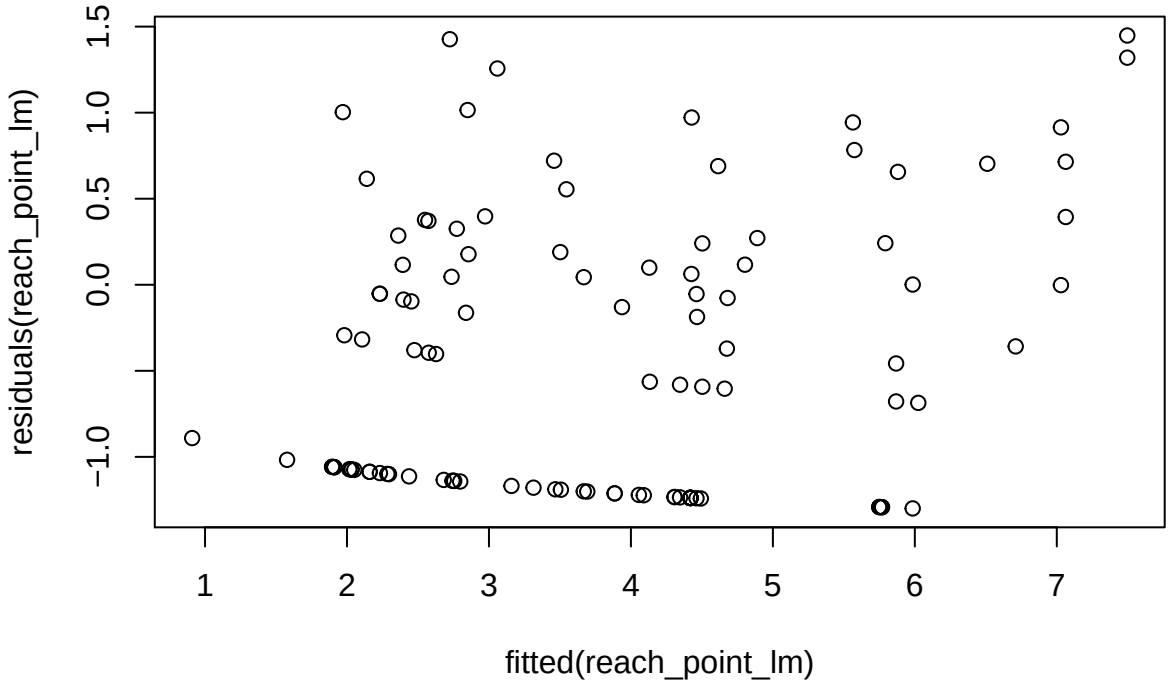
4.1 Reaches

```
reach_point_lead_nb <- glm.nb(lead_point ~ reach, data = reach_point_lead)

reach_point_lm <- glmer(
  lead_point ~
    reach *
    background +
    (1|dyad),
  data = reach_point_lead,
  family = negbin(0.2681)
)
summary(reach_point_lm)
```

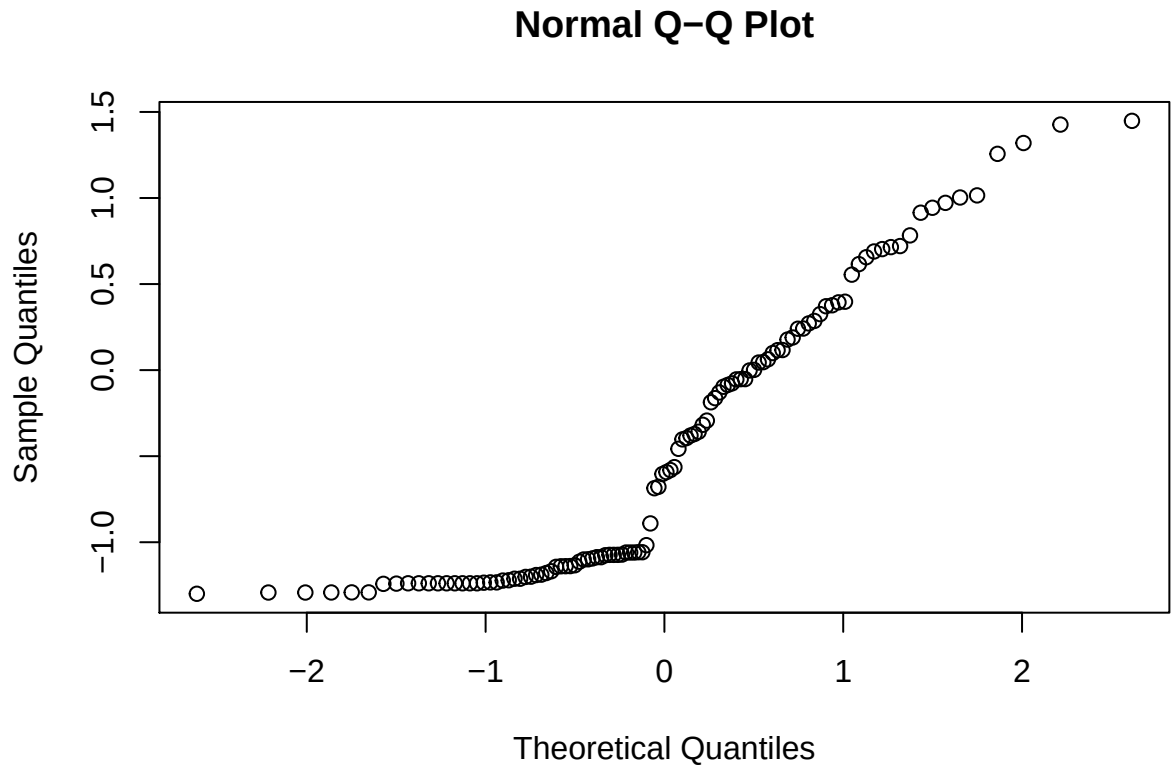
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Negative Binomial(0.268)  ( log )
## Formula: lead_point ~ reach * background + (1 | dyad)
##    Data: reach_point_lead
##
```

```
##      AIC      BIC   logLik deviance df.resid
##    523.3    545.1   -253.7    507.3      104
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.5066 -0.4982 -0.3934  0.1437  3.0203
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  dyad   (Intercept) 0.1569   0.396
## Number of obs: 112, groups:  dyad, 57
##
## Fixed effects:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.72163    0.60141   1.200    0.230
## reach                      0.06136    0.09716   0.632    0.528
## backgroundChinese          1.10777    0.72841   1.521    0.128
## backgroundEnglish          0.84357    0.68166   1.238    0.216
## reach:backgroundChinese -0.24686    0.16105  -1.533    0.125
## reach:backgroundEnglish -0.08716    0.13746  -0.634    0.526
##
## Correlation of Fixed Effects:
##             (Intr) reach  bckgrC bckgrE rch:bC
## reach       -0.724
## bckgrndChns -0.709  0.550
## bckgrndEngl -0.557  0.506  0.508
## rch:bckgrnC  0.453 -0.610 -0.710 -0.298
## rch:bckgrnE  0.449 -0.681 -0.366 -0.599  0.412
```

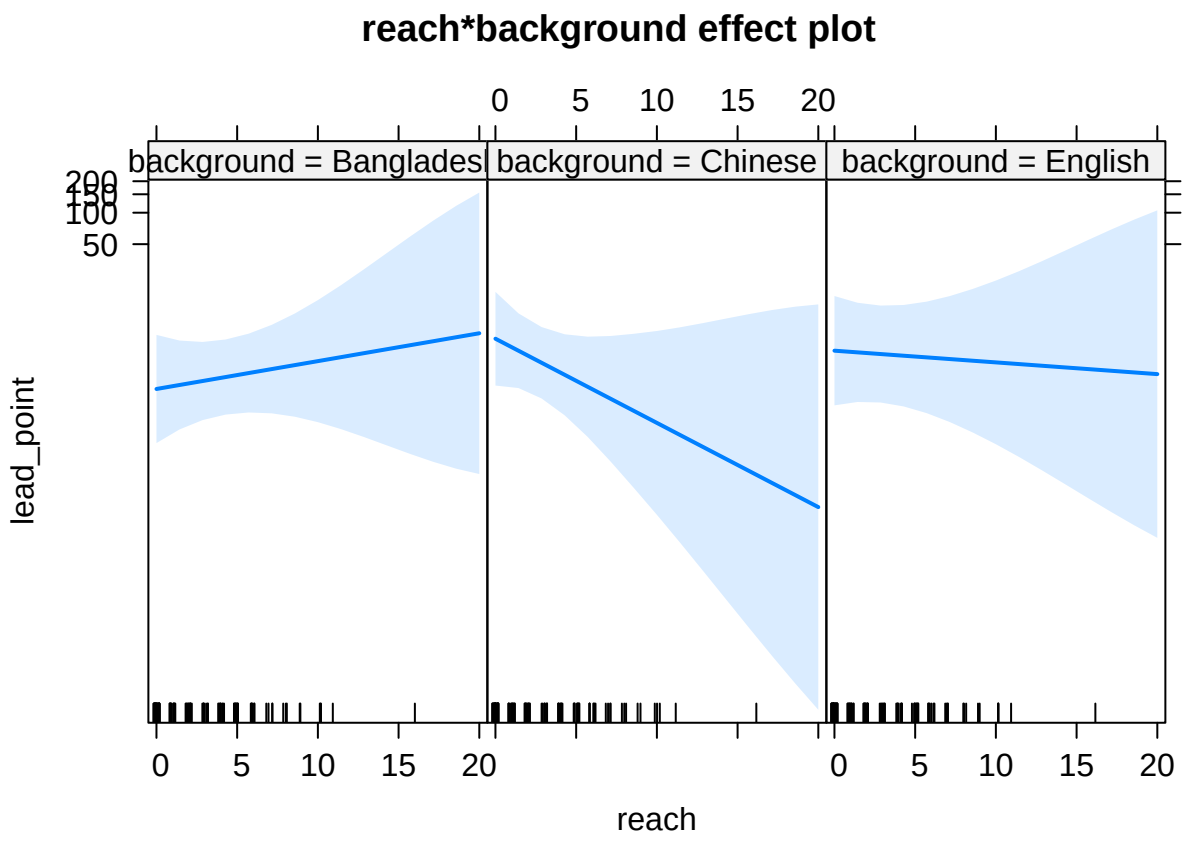
```
plot(fitted(reach_point_lm), residuals(reach_point_lm))
```



```
qqnorm(residuals(reach_point_lm))
```




```
plot(allEffects(reach_point_lm))
```



4.2 HoGs

```
hg_point_lead_nb <- glm.nb(lead_point ~ ho_gv, data = filter(hg_point_lead, ho_gv < 20))

hg_point_lm <- glmer(
  lead_point ~
    ho_gv *
    background +
    (1|dyad),
  data = filter(hg_point_lead, ho_gv < 20),
  family = negbin(0.2606)
)
```

```
## singular fit
```

```
summary(hg_point_lm)
```

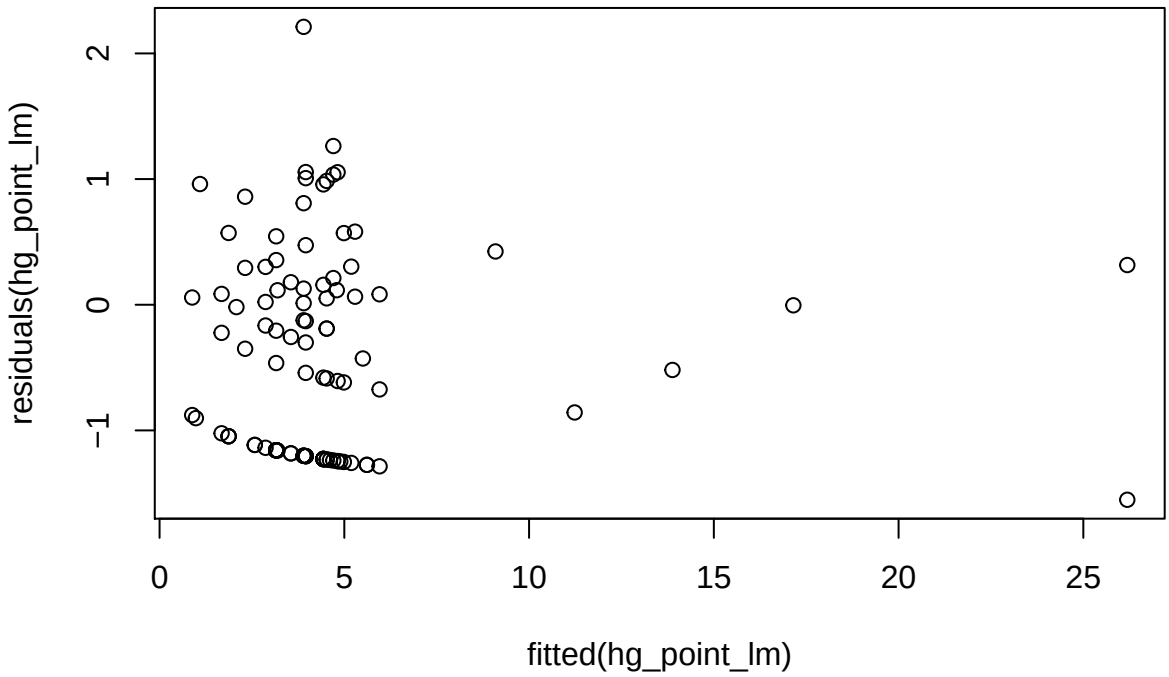
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Negative Binomial(0.261)  ( log )
## Formula: lead_point ~ ho_gv * background + (1 | dyad)
##    Data: filter(hg_point_lead, ho_gv < 20)
##
##      AIC      BIC   logLik deviance df.resid
##    503.8    525.3   -243.9    487.8      101
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.5080 -0.4942 -0.3979  0.1241  6.0969
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  dyad   (Intercept) 1.41e-10 1.187e-05
## Number of obs: 109, groups:  dyad, 57
##
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.37529    0.46393   2.964  0.00303 **
## ho_gv                    -0.10718    0.08031  -1.335  0.18200
## backgroundChinese         0.11400    0.68904   0.165  0.86859
## backgroundEnglish        -0.22613    0.62893  -0.360  0.71919
## ho_gv:backgroundChinese   0.12680    0.13875   0.914  0.36081
## ho_gv:backgroundEnglish   0.31880    0.15566   2.048  0.04056 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ho_gv  bckgrC bckgrE h_gv:C
## ho_gv       -0.681
## bckgrndChns -0.673  0.459
## bckgrndEngl -0.738  0.502  0.497
## h_gv:bckgrC  0.394 -0.579 -0.714 -0.291
## h_gv:bckgrE  0.351 -0.516 -0.237 -0.621  0.299
## convergence code: 0
```

```
## singular fit
```

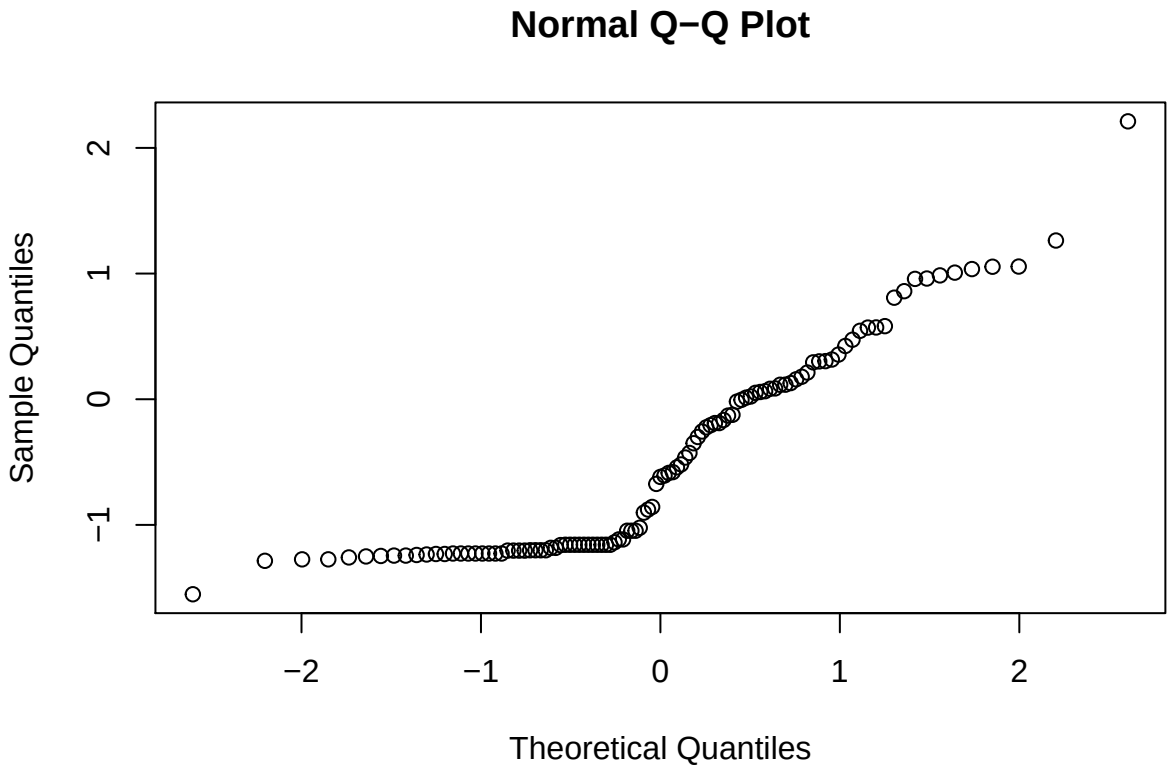
```
hg_point_lm_null <- glmer(
  lead_point ~
    ho_gv +
    background +
    (1|dyad),
  data = filter(hg_point_lead, ho_gv < 20),
  family = negbin(0.2606)
)
anova(hg_point_lm_null, hg_point_lm)
```

```
## Data: filter(hg_point_lead, ho_gv < 20)
## Models:
## hg_point_lm_null: lead_point ~ ho_gv + background + (1 | dyad)
## hg_point_lm: lead_point ~ ho_gv * background + (1 | dyad)
##                  Df    AIC    BIC  logLik deviance  Chisq Chi Df
## hg_point_lm_null  6 504.69 520.84 -246.35   492.69
## hg_point_lm       8 503.79 525.32 -243.89   487.79 4.9055      2
##                  Pr(>Chisq)
## hg_point_lm_null
## hg_point_lm         0.08606 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

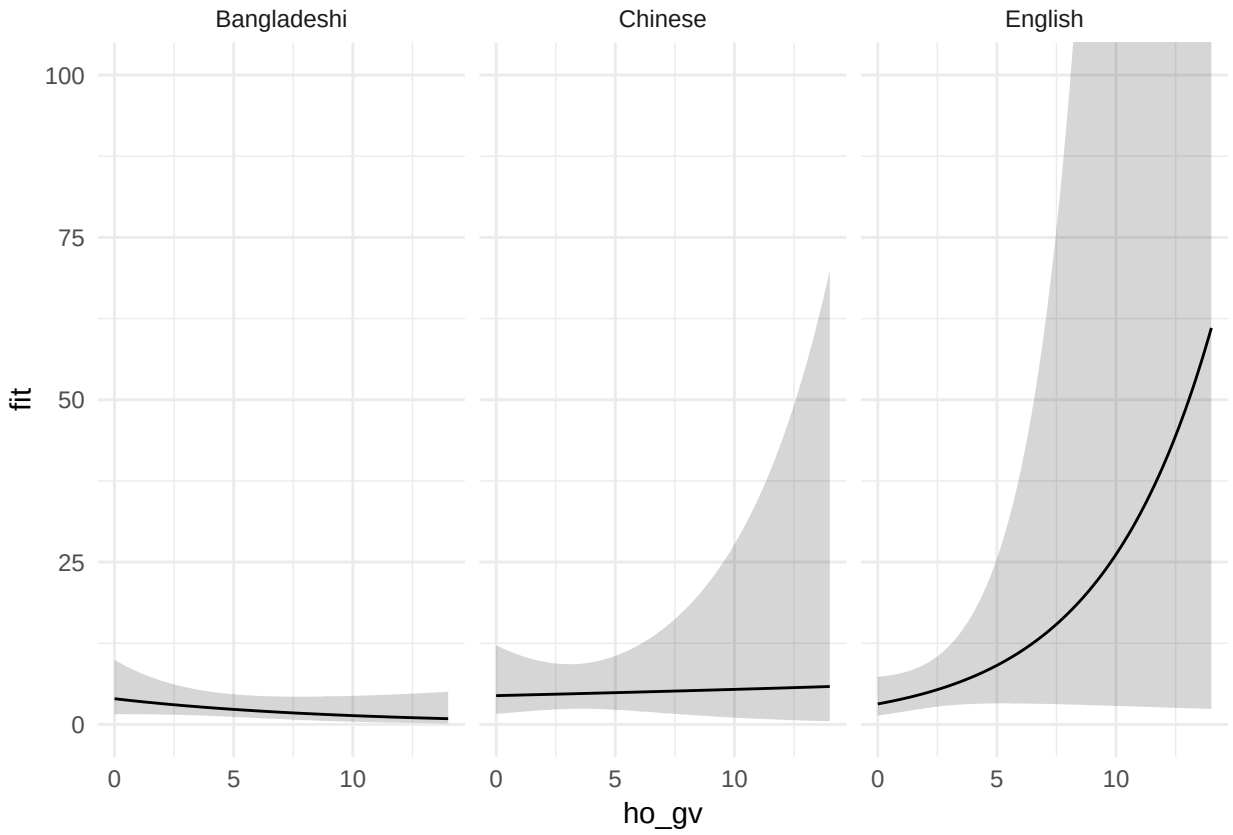
```
plot(fitted(hg_point_lm), residuals(hg_point_lm))
```



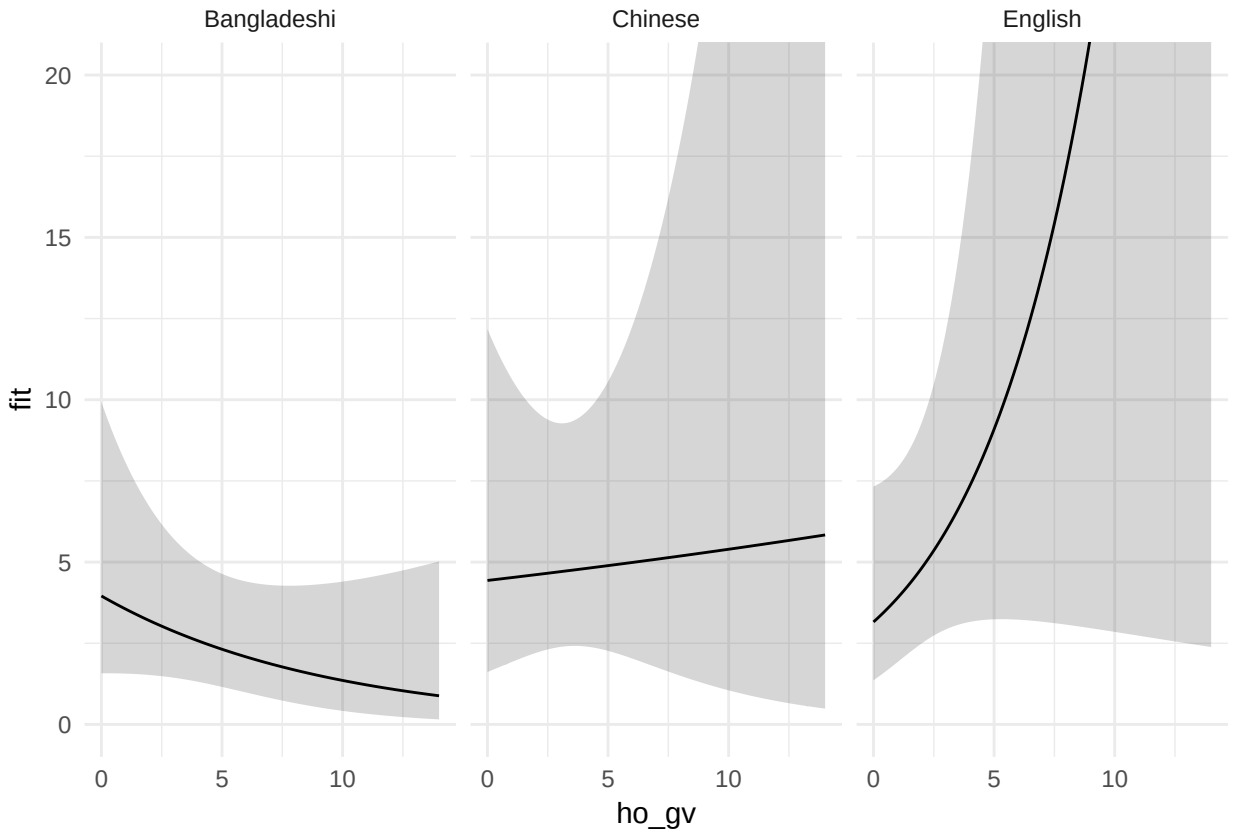

```
qqnorm(residuals(hg_point_lm))
```




```
hg_eff <- as_tibble(effect("ho_gv:background", hg_point_lm, xlevels = 100))

ggplot(hg_eff, aes(ho_gv, fit)) +
  geom_ribbon(aes(ymax = upper, ymin = lower), alpha = 0.2) +
  geom_line() +
  facet_grid(~ background) +
  coord_cartesian(ylim = c(0, 100))
```




```
ggplot(hg_eff, aes(ho_gv, fit)) +
  geom_ribbon(aes(ymax = upper, ymin = lower), alpha = 0.2) +
  geom_line() +
  facet_grid(~ background) +
  coord_cartesian(ylim = c(0, 20))
```



5 Analysis 2. Predictors of vocabulary scores at 12 and 18 months

5.1 Comprehension at 12 and 18 months

5.1.1 All gestures combined

```
all_gest_lm <- lmer(
  understand ~
    count_tot *
    months *
    background +
    (1|dyad),
  data = vocab
)
summary(all_gest_lm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ count_tot * months * background + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1180.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

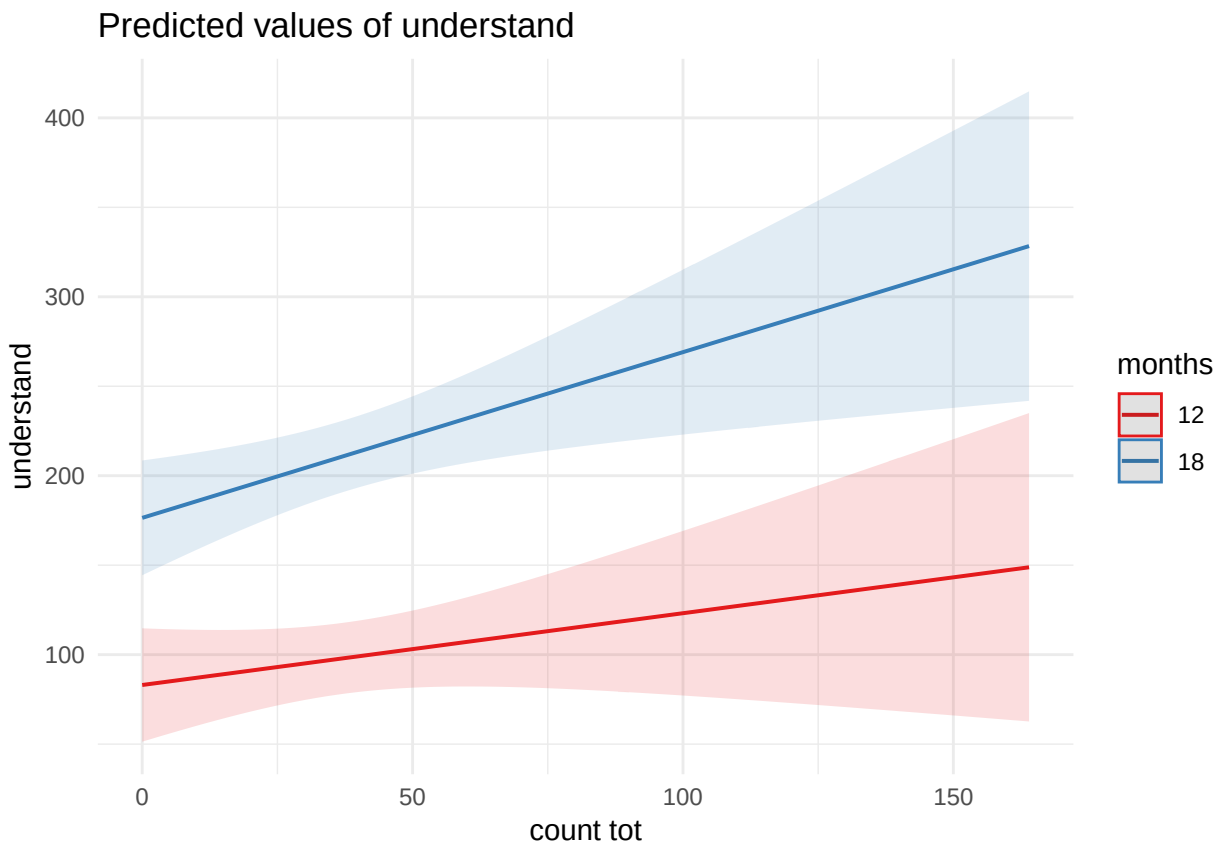
```
## -1.74717 -0.54694  0.01362  0.40251  1.85188
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 2671     51.68
##  Residual             2299     47.95
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                                     Estimate Std. Error         df
## (Intercept)                        -11.53326   90.20899   60.50470
## count_tot                           -1.71945    2.03317   60.50470
## months                              10.30257    5.65012   47.87064
## backgroundChinese                 -215.33019  116.59958   60.50470
## backgroundEnglish                  -90.30873  106.47214   60.64043
## count_tot:months                     0.11738    0.12734   47.87064
## count_tot:backgroundChinese          3.78521    2.52892   60.50470
## count_tot:backgroundEnglish         -0.24189    2.31867   60.53955
## months:backgroundChinese             9.90219    7.30306   47.87064
## months:backgroundEnglish             6.81052    6.69456   48.26704
## count_tot:months:backgroundChinese  -0.10980    0.15840   47.87064
## count_tot:months:backgroundEnglish   0.01585    0.14537   47.97274
##                                    t value Pr(>|t|)
## (Intercept)                         -0.128   0.8987
## count_tot                           -0.846   0.4011
## months                               1.823   0.0745 .
## backgroundChinese                   -1.847   0.0697 .
## backgroundEnglish                   -0.848   0.3997
## count_tot:months                     0.922   0.3613
## count_tot:backgroundChinese          1.497   0.1397
## count_tot:backgroundEnglish         -0.104   0.9173
## months:backgroundChinese             1.356   0.1815
## months:backgroundEnglish             1.017   0.3141
## count_tot:months:backgroundChinese  -0.693   0.4915
## count_tot:months:backgroundEnglish   0.109   0.9136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cnt_tt months bckgrC bckgrE cnt_t: cnt_:C cnt_:E mnth:C
## count_tot   -0.888
## months      -0.940  0.835
## bckgrndChns -0.774  0.687  0.727
## bckgrndEngl -0.847  0.753  0.796  0.655
## cnt_tt:mnth  0.835 -0.940 -0.888 -0.646 -0.707
## cnt_tt:bckC  0.714 -0.804 -0.671 -0.861 -0.605  0.755
## cnt_tt:bckE  0.779 -0.877 -0.732 -0.603 -0.826  0.824  0.705
## mnths:bckgC  0.727 -0.646 -0.774 -0.940 -0.616  0.687  0.809  0.566
## mnths:bckgE  0.793 -0.705 -0.844 -0.613 -0.940  0.750  0.566  0.775  0.653
## cnt_tt:mn:C -0.671  0.755  0.714  0.809  0.569 -0.804 -0.940 -0.662 -0.861
## cnt_tt:mn:E -0.731  0.823  0.778  0.566  0.777 -0.876 -0.662 -0.940 -0.602
##             mnth:E cn_::C
## count_tot
## months
```

```
## bckgrndChns
## bckgrndEngl
## cnt_tt:mnth
## cnt_tt:bckC
## cnt_tt:bckE
## mnths:bckgC
## mnths:bckgE
## cnt_tt:mn:C -0.603
## cnt_tt:mn:E -0.825  0.704
```

```
all_gest_lm_2 <- lmer(
  understand ~
    count_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(all_gest_lm_2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ count_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1220.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.53820 -0.47681 -0.09134  0.45523  1.97984
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3279     57.26
##  Residual             2365     48.63
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                     Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)      -103.79077   40.02556   67.41098  -2.593   0.0117 *
## count_tot          -0.64884    0.83279   67.33012  -0.779   0.4386
## months             15.56858    2.50342   52.42242   6.219 8.42e-08 ***
## count_tot:months    0.08750    0.05192   52.11407   1.685   0.0979 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cnt_tt months
## count_tot   -0.781
## months      -0.933  0.728
## cnt_tt:mnth  0.731 -0.932 -0.783
```

```
plot_model(all_gest_lm_2, type = "pred", terms = c("count_tot", "months"))
```



5.1.2 HoGs + points

```
hgp_lm <- lmer(
  understand ~
    hgp_tot *
    months *
    background +
    (1|dyad),
  data = vocab
)
summary(hgp_lm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ hgp_tot * months * background + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1183.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.64675 -0.51967 -0.00448  0.42619  1.76485
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
```

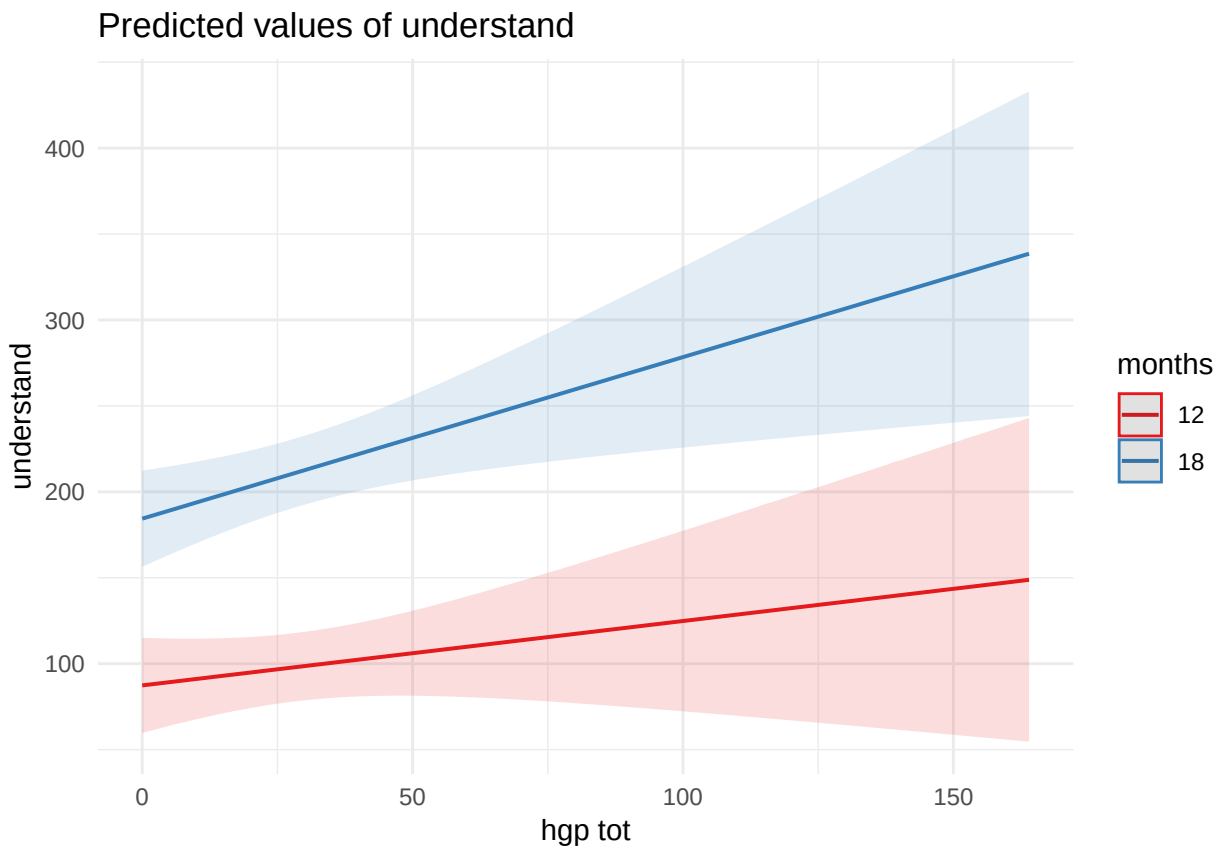
```
##  dyad     (Intercept) 2890     53.75
##  Residual             2304     48.00
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                                    Estimate Std. Error         df t value
## (Intercept)                      -3.025e+01  7.880e+01  6.121e+01  -0.384
## hgp_tot                          -1.743e+00  2.377e+00  6.121e+01  -0.733
## months                            1.113e+01  4.919e+00  4.790e+01   2.262
## backgroundChinese                -1.724e+02  1.018e+02  6.121e+01  -1.694
## backgroundEnglish                -8.627e+01  9.437e+01  6.133e+01  -0.914
## hgp_tot:months                    1.350e-01  1.484e-01  4.790e+01   0.910
## hgp_tot:backgroundChinese         3.597e+00  2.822e+00  6.121e+01   1.275
## hgp_tot:backgroundEnglish        -1.949e-01  2.624e+00  6.123e+01  -0.074
## months:backgroundChinese          9.052e+00  6.353e+00  4.790e+01   1.425
## months:backgroundEnglish          7.037e+00  5.916e+00  4.830e+01   1.189
## hgp_tot:months:backgroundChinese -1.246e-01  1.762e-01  4.790e+01  -0.707
## hgp_tot:months:backgroundEnglish -5.375e-03  1.639e-01  4.797e+01  -0.033
##                                  Pr(>|t|)
## (Intercept)                        0.7024
## hgp_tot                            0.4664
## months                             0.0283 *
## backgroundChinese                  0.0954 .
## backgroundEnglish                  0.3642
## hgp_tot:months                     0.3676
## hgp_tot:backgroundChinese          0.2073
## hgp_tot:backgroundEnglish          0.9410
## months:backgroundChinese           0.1607
## months:backgroundEnglish           0.2401
## hgp_tot:months:backgroundChinese   0.4829
## hgp_tot:months:backgroundEnglish   0.9740
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) hgp_tt months bckgrC bckgrE hgp_t: hgp_:C hgp_:E mnth:C
## hgp_tot     -0.850
## months      -0.936  0.796
## bckgrndChns -0.774  0.658  0.725
## bckgrndEngl -0.835  0.709  0.782  0.647
## hgp_tt:mnth  0.796 -0.936 -0.850 -0.616 -0.664
## hgp_tt:bckC  0.716 -0.842 -0.670 -0.809 -0.597  0.789
## hgp_tt:bckE  0.770 -0.906 -0.721 -0.596 -0.771  0.848  0.763
## mnths:bckgC  0.725 -0.616 -0.774 -0.936 -0.605  0.658  0.758  0.558
## mnths:bckgE  0.779 -0.661 -0.831 -0.603 -0.937  0.706  0.557  0.721  0.644
## hgp_tt:mn:C -0.670  0.789  0.716  0.758  0.559 -0.842 -0.936 -0.715 -0.809
## hgp_tt:mn:E -0.720  0.848  0.769  0.558  0.724 -0.905 -0.714 -0.937 -0.596
##             mnth:E hg_::C
## hgp_tot
## months
## bckgrndChns
## bckgrndEngl
## hgp_tt:mnth
## hgp_tt:bckC
```

```
## hgp_tt:bckE
## mnths:bckgC
## mnths:bckgE
## hgp_tt:mn:C -0.595
## hgp_tt:mn:E -0.771  0.763
```

```
hgp_lm_2 <- lmer(
  understand ~
    hgp_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(hgp_lm_2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ hgp_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1220.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.48779 -0.47461 -0.06162  0.46424  1.92329
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3315     57.58
##  Residual             2349     48.47
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                  Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    -106.62133   34.78108   67.61030  -3.065  0.00312 **
## hgp_tot          -0.75634    0.84955   67.53140  -0.890  0.37648
## months           16.16696    2.17268   52.40323   7.441 9.39e-10 ***
## hgp_tot:months    0.09424    0.05290   52.09060   1.782  0.08065 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) hgp_tt months
## hgp_tot     -0.698
## months      -0.932  0.650
## hgp_tt:mnth  0.652 -0.932 -0.700
```

```
plot_model(hgp_lm_2, type = "pred", terms = c("hgp_tot", "months"))
```



5.1.3 Reaches

```
reach_lm <- lmer(
  understand ~
    reach_tot *
    months *
    background +
    (1|dyad),
  data = vocab
)
summary(reach_lm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ reach_tot * months * background + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1178.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.77264 -0.51430  0.02511  0.54580  1.62157
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
```

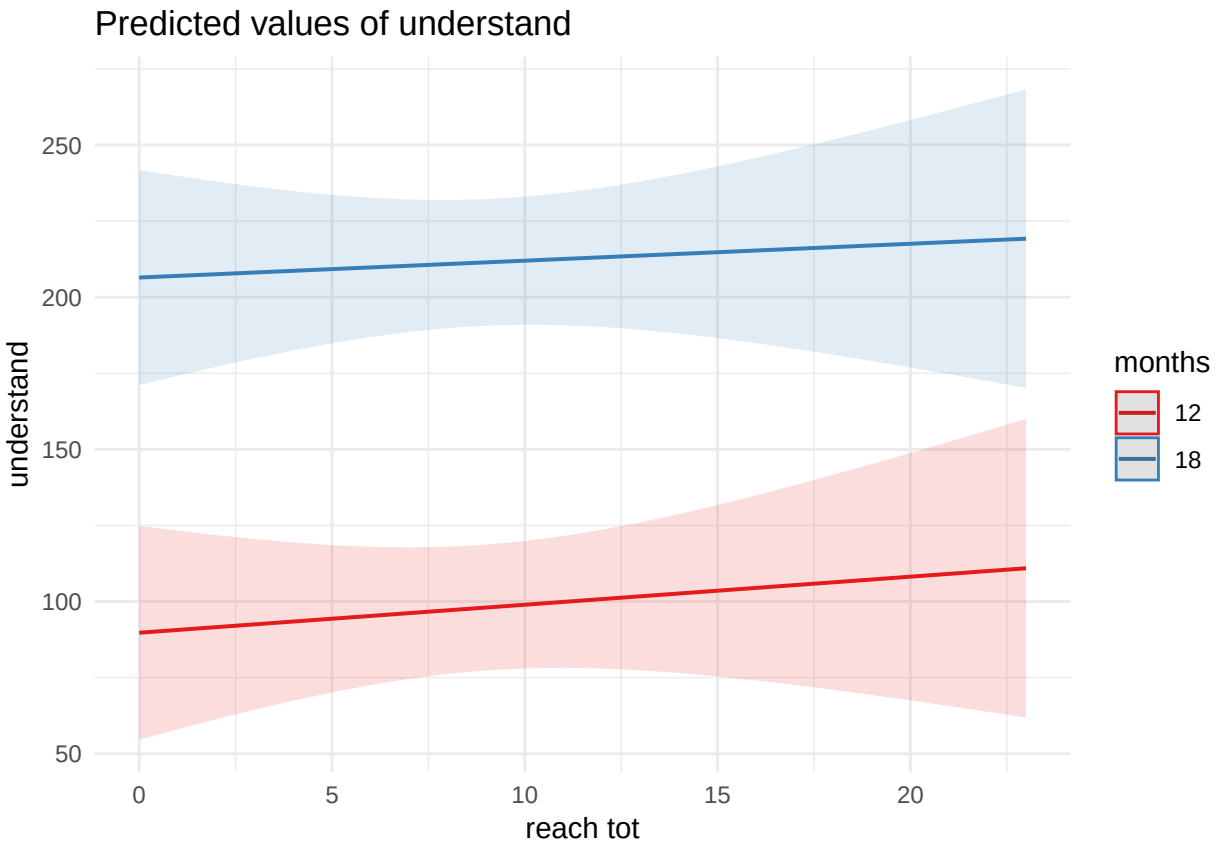
```
##  dyad     (Intercept) 3654     60.45
##  Residual             2485     49.85
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                                    Estimate Std. Error        df t value
## (Intercept)                         -31.8808    86.6703   62.7555  -0.368
## reach_tot                            -4.2115     6.6556   62.7555  -0.633
## months                               12.9846     5.3702   47.8794   2.418
## backgroundChinese                  -163.7787   118.8588   62.7555  -1.378
## backgroundEnglish                  -138.0321   108.6959   62.8027  -1.270
## reach_tot:months                      0.1727     0.4124   47.8794   0.419
## reach_tot:backgroundChinese          10.3022    10.4027   62.7555   0.990
## reach_tot:backgroundEnglish           4.5599     9.4843   62.7579   0.481
## months:backgroundChinese              8.1027     7.3647   47.8794   1.100
## months:backgroundEnglish              8.3412     6.7510   48.0873   1.236
## reach_tot:months:backgroundChinese   -0.2407     0.6446   47.8794  -0.373
## reach_tot:months:backgroundEnglish   -0.1323     0.5877   47.8900  -0.225
##                                    Pr(>|t|)
## (Intercept)                          0.7142
## reach_tot                            0.5292
## months                               0.0195 *
## backgroundChinese                    0.1731
## backgroundEnglish                    0.2088
## reach_tot:months                     0.6772
## reach_tot:backgroundChinese          0.3258
## reach_tot:backgroundEnglish          0.6323
## months:backgroundChinese             0.2767
## months:backgroundEnglish             0.2226
## reach_tot:months:backgroundChinese   0.7105
## reach_tot:months:backgroundEnglish   0.8229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rch_tt months bckgrC bckgrE rch_t: rch_:C rch_:E mnth:C
## reach_tot   -0.865
## months      -0.929  0.804
## bckgrndChns -0.729  0.631  0.678
## bckgrndEngl -0.797  0.690  0.741  0.581
## rch_tt:mnth  0.804 -0.929 -0.865 -0.586 -0.641
## rch_tt:bckC  0.553 -0.640 -0.514 -0.843 -0.441  0.595
## rch_tt:bckE  0.607 -0.702 -0.564 -0.443 -0.795  0.652  0.449
## mnths:bckgC  0.678 -0.586 -0.729 -0.929 -0.540  0.631  0.783  0.411
## mnths:bckgE  0.739 -0.639 -0.795 -0.539 -0.930  0.688  0.409  0.738  0.580
## rch_tt:mn:C -0.514  0.595  0.553  0.783  0.410 -0.640 -0.929 -0.417 -0.843
## rch_tt:mn:E -0.564  0.652  0.607  0.411  0.739 -0.702 -0.417 -0.929 -0.443
##             mnth:E rc_::C
## reach_tot
## months
## bckgrndChns
## bckgrndEngl
## rch_tt:mnth
## rch_tt:bckC
```

```
## rch_tt:bckE
## mnths:bckgC
## mnths:bckgE
## rch_tt:mn:C -0.440
## rch_tt:mn:E -0.794  0.449
```

```
reach_lm_2 <- lmer(
  understand ~
    reach_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(reach_lm_2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ reach_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1221.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5140 -0.5941 -0.0561  0.5158  1.7552
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3640     60.34
##  Residual             2473     49.73
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                     Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)      -143.62983   43.54107   68.11489  -3.299  0.00155 **
## reach_tot           1.65200    3.91756   68.06536   0.422  0.67458
## months             19.44798    2.70583   52.22044   7.187 2.43e-09 ***
## reach_tot:months   -0.06093    0.24289   52.00199  -0.251  0.80291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rch_tt months
## reach_tot   -0.809
## months      -0.930  0.751
## rch_tt:mnth  0.753 -0.929 -0.809
```

```
plot_model(reach_lm_2, type = "pred", terms = c("reach_tot", "months"))
```



5.1.4 Maternal utterances

```
utt_lm <- lmer(
  understand ~
    utt_tot *
    months *
    background +
    (1|dyad),
  data = vocab
)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
summary(utt_lm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ utt_tot * months * background + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1111.2
##
```

```
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.51565 -0.49198  0.00639  0.48531  1.73592
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3872     62.23
##  Residual             2410     49.09
## Number of obs: 99, groups:  dyad, 50
##
## Fixed effects:
##                                   Estimate Std. Error         df t value
## (Intercept)                     -8.473e+01  7.748e+01  5.740e+01  -1.094
## utt_tot                          6.841e-03  1.028e-01  5.740e+01   0.067
## months                           1.444e+01  4.778e+00  4.322e+01   3.021
## backgroundChinese               -1.069e+02  1.257e+02  5.740e+01  -0.851
## backgroundEnglish                1.413e+02  2.405e+02  5.764e+01   0.587
## utt_tot:months                   1.704e-03  6.341e-03  4.322e+01   0.269
## utt_tot:backgroundChinese        5.678e-02  1.572e-01  5.740e+01   0.361
## utt_tot:backgroundEnglish       -2.117e-01  2.786e-01  5.756e+01  -0.760
## months:backgroundChinese         6.038e+00  7.750e+00  4.322e+01   0.779
## months:backgroundEnglish         1.574e+00  1.511e+01  4.459e+01   0.104
## utt_tot:months:backgroundChinese -1.938e-03  9.693e-03  4.322e+01  -0.200
## utt_tot:months:backgroundEnglish  1.116e-03  1.740e-02  4.414e+01   0.064
##                                  Pr(>|t|)
## (Intercept)                       0.27869
## utt_tot                           0.94718
## months                            0.00422 **
## backgroundChinese                 0.39851
## backgroundEnglish                 0.55921
## utt_tot:months                    0.78945
## utt_tot:backgroundChinese         0.71923
## utt_tot:backgroundEnglish         0.45037
## months:backgroundChinese          0.44021
## months:backgroundEnglish          0.91753
## utt_tot:months:backgroundChinese  0.84243
## utt_tot:months:backgroundEnglish  0.94916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) utt_tt months bckgrC bckgrE utt_t: utt_:C utt_:E mnth:C
## utt_tot     -0.821
## months      -0.925  0.760
## bckgrndChns -0.617  0.506  0.570
## bckgrndEngl -0.322  0.265  0.298  0.199
## utt_tt:mnth  0.760 -0.925 -0.821 -0.468 -0.245
## utt_tt:bckC  0.537 -0.654 -0.497 -0.867 -0.173  0.605
## utt_tt:bckE  0.303 -0.369 -0.280 -0.187 -0.953  0.341  0.241
## mnths:bckgC  0.570 -0.468 -0.617 -0.925 -0.184  0.506  0.802  0.173
## mnths:bckgE  0.293 -0.240 -0.316 -0.180 -0.927  0.260  0.157  0.881  0.195
## utt_tt:mn:C -0.497  0.605  0.537  0.802  0.160 -0.654 -0.925 -0.223 -0.867
## utt_tt:mn:E -0.277  0.337  0.299  0.171  0.887 -0.364 -0.221 -0.926 -0.185
##             mnth:E ut_::C
```

```
## utt_tot
## months
## bckgrndChns
## bckgrndEngl
## utt_tt:mnth
## utt_tt:bckC
## utt_tt:bckE
## mnths:bckgC
## mnths:bckgE
## utt_tt:mn:C -0.170
## utt_tt:mn:E -0.954  0.238
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```
utt_lm_2 <- lmer(
  understand ~
    utt_tot *
    months +
    (1|dyad),
  data = vocab
)
```

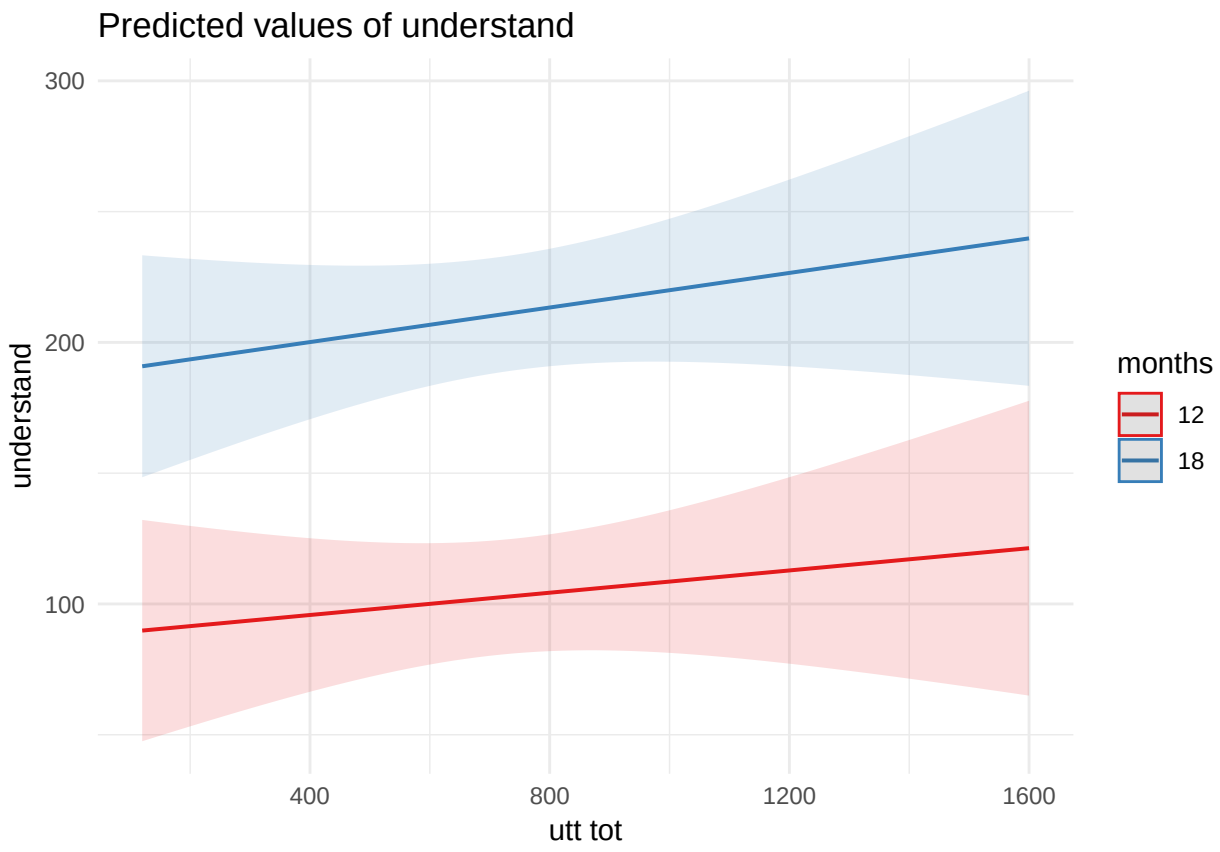
```
## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
summary(utt_lm_2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ utt_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1122.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.50968 -0.52981  0.00072  0.46811  1.80340
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3963     62.95
##  Residual             2290     47.86
## Number of obs: 99, groups:  dyad, 50
##
## Fixed effects:
##                  Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    -1.120e+02  5.767e+01  6.337e+01  -1.943   0.0565 .
## utt_tot        -2.221e-03  7.058e-02  6.336e+01  -0.031   0.9750
## months          1.661e+01  3.547e+00  4.711e+01   4.682 2.43e-05 ***
## utt_tot:months  1.959e-03  4.336e-03  4.701e+01   0.452   0.6534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##             (Intr) utt_tt months
## utt_tot     -0.892
## months      -0.921  0.821
## utt_tt:mnth  0.822 -0.921 -0.891
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```
plot_model(utt_lm_2, type = "pred", terms = c("utt_tot", "months"))
```



5.1.5 Contingent talks

```
ct_lm <- lmer(
  understand ~
    ct_tot *
    months *
    background +
    (1|dyad),
  data = vocab
)
summary(ct_lm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ ct_tot * months * background + (1 | dyad)
```

```
##    Data: vocab
##
## REML criterion at convergence: 1158.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5331 -0.5368  0.0162  0.4602  1.6119
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3625     60.21
##  Residual             2446     49.46
## Number of obs: 107, groups:  dyad, 54
##
## Fixed effects:
##                                  Estimate Std. Error        df t value
## (Intercept)                     -110.6215    53.7851   61.5684  -2.057
## ct_tot                             9.8165     9.9876   61.5684   0.983
## months                            16.6784     3.3313   46.9256   5.007
## backgroundChinese                -47.4939    87.2923   61.5684  -0.544
## backgroundEnglish                -18.8444    74.4312   61.6706  -0.253
## ct_tot:months                     -0.4354     0.6186   46.9256  -0.704
## ct_tot:backgroundChinese          -7.3882    11.7782   61.5684  -0.627
## ct_tot:backgroundEnglish         -13.0948    10.2215   61.5696  -1.281
## months:backgroundChinese           2.4688     5.4067   46.9256   0.457
## months:backgroundEnglish           2.3351     4.6350   47.3826   0.504
## ct_tot:months:backgroundChinese    0.5818     0.7295   46.9256   0.797
## ct_tot:months:backgroundEnglish    0.6581     0.6331   46.9308   1.039
##                                  Pr(>|t|)
## (Intercept)                         0.044 *
## ct_tot                              0.330
## months                           8.25e-06 ***
## backgroundChinese                   0.588
## backgroundEnglish                   0.801
## ct_tot:months                       0.485
## ct_tot:backgroundChinese            0.533
## ct_tot:backgroundEnglish            0.205
## months:backgroundChinese            0.650
## months:backgroundEnglish            0.617
## ct_tot:months:backgroundChinese     0.429
## ct_tot:months:backgroundEnglish     0.304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ct_tot months bckgrC bckgrE ct_tt: ct_t:C ct_t:E mnth:C
## ct_tot      -0.596
## months      -0.929  0.554
## bckgrndChns -0.616  0.367  0.572
## bckgrndEngl -0.723  0.431  0.671  0.445
## ct_tt:mnths  0.554 -0.929 -0.596 -0.341 -0.400
## ct_tt:bckgC  0.506 -0.848 -0.470 -0.624 -0.365  0.788
## ct_tt:bckgE  0.583 -0.977 -0.541 -0.359 -0.493  0.908  0.829
## mnths:bckgC  0.572 -0.341 -0.616 -0.929 -0.414  0.367  0.579  0.333
```

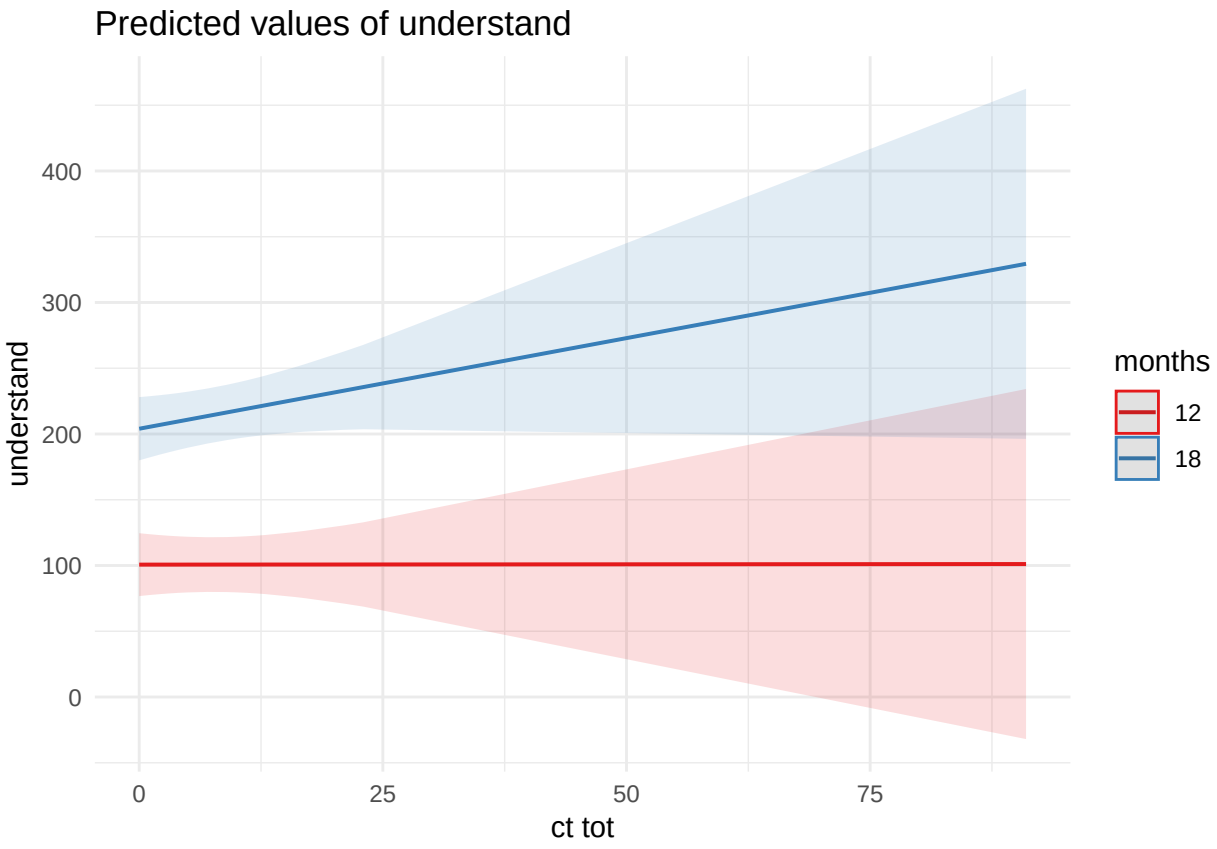
```
## mnths:bckgE  0.668 -0.398 -0.719 -0.411 -0.930  0.428  0.338  0.456  0.443
## ct_tt:mnt:C -0.470  0.788  0.506  0.579  0.339 -0.848 -0.929 -0.770 -0.624
## ct_tt:mnt:E -0.541  0.908  0.583  0.333  0.458 -0.977 -0.770 -0.929 -0.359
##             mnth:E ct_::C
## ct_tot
## months
## bckgrndChns
## bckgrndEngl
## ct_tt:mnths
## ct_tt:bckgC
## ct_tt:bckgE
## mnths:bckgC
## mnths:bckgE
## ct_tt:mnt:C -0.363
## ct_tt:mnt:E -0.491  0.829
```

```
ct_lm_2 <- lmer(
  understand ~
    ct_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(ct_lm_2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: understand ~ ct_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1199.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.50281 -0.53484 -0.06465  0.45958  1.71123
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept) 3680     60.67
##  Residual             2393     48.92
## Number of obs: 107, groups:  dyad, 54
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   -105.9926    29.3160   67.3472  -3.616 0.000574 ***
## ct_tot          -2.7417     1.9235   67.2909  -1.425 0.158670
## months          17.2216     1.8184   51.2881   9.471 7.53e-13 ***
## ct_tot:months    0.2289     0.1189   51.0070   1.924 0.059887 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ct_tot months
## ct_tot      -0.494
## months      -0.928  0.457
```

```
## ct_tt:mnths  0.459 -0.927 -0.494
```

```
plot_model(ct_lm_2, type = "pred", terms = c("ct_tot", "months"))
```



5.2 Production at 12 and 18 months

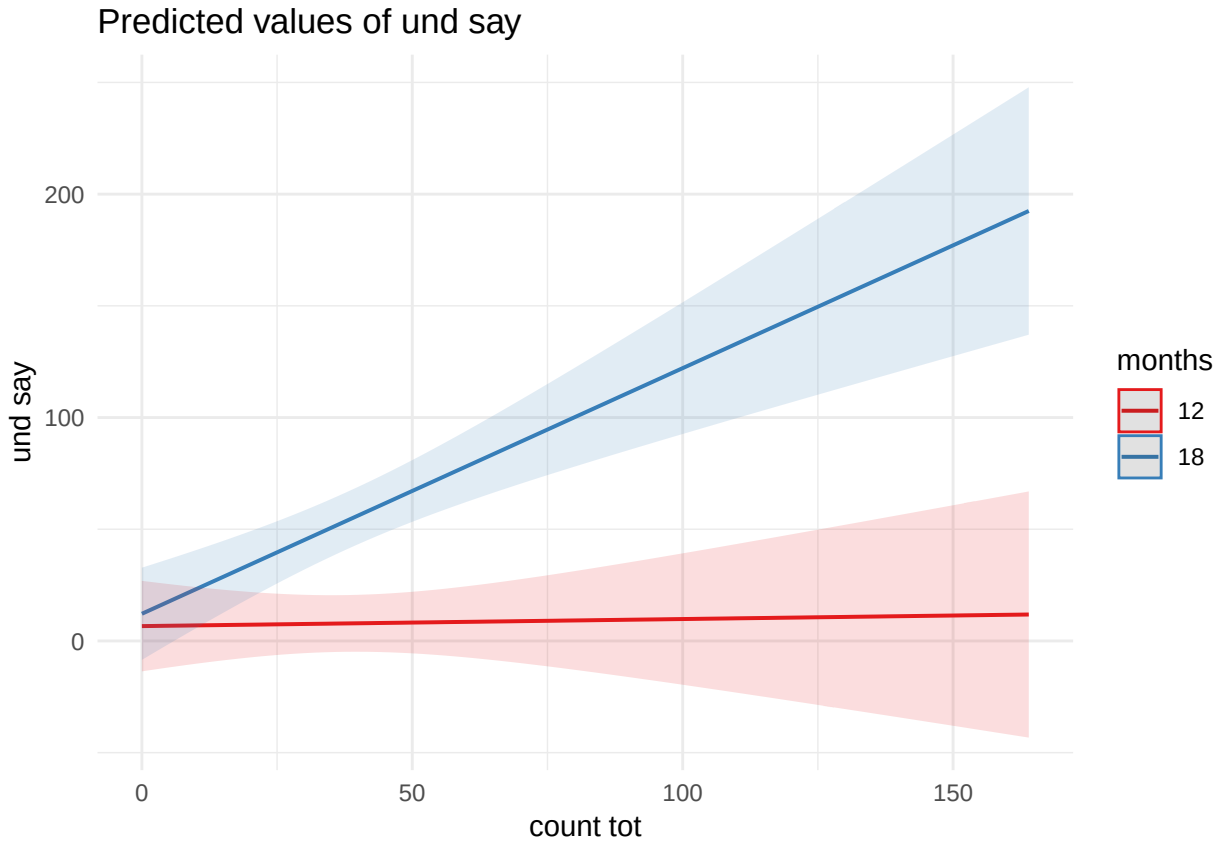
5.2.1 All gestures combined

```
all_gest_lm_2_undsay <- lmer(
  und_say ~
    count_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(all_gest_lm_2_undsay)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: und_say ~ count_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1147.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1.6451 -0.2949 -0.0400  0.1150  5.1936
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept)  281     16.76
##  Residual             2026     45.01
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                   Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)      -4.40898   35.33834 58.60030  -0.125 0.901137
## count_tot        -2.10393    0.73592 58.32093  -2.859 0.005887 **
## months            0.92064    2.30729 53.64769   0.399 0.691468
## count_tot:months  0.17796    0.04795 53.17912   3.711 0.000495 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cnt_tt months
## count_tot   -0.781
## months      -0.975  0.761
## cnt_tt:mnth  0.763 -0.975 -0.782
```

```
plot_model(all_gest_lm_2_undsay, type = "pred", terms = c("count_tot", "months"))
```

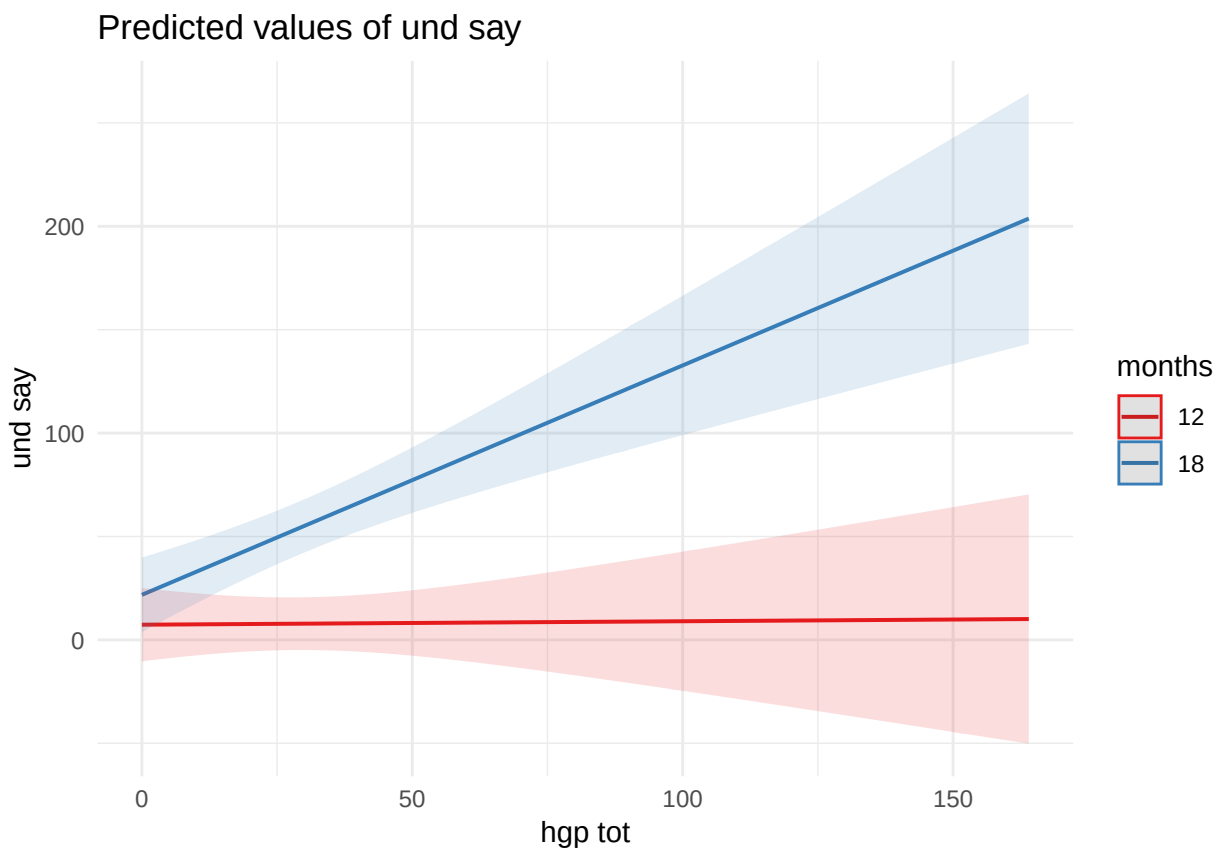


5.2.2 HoGs + point

```
hgp_lm_2_undsay <- lmer(
  und_say ~
    hgp_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(hgp_lm_2_undsay)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: und_say ~ hgp_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1147.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6473 -0.2989 -0.0408  0.1149  5.2699
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept)  296.9   17.23
##  Residual             2024.2   44.99
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                 Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    -21.53337   30.78529  58.62260  -0.699 0.487025
## hgp_tot         -2.16893    0.75263  58.33881  -2.882 0.005525 **
## months           2.40664    2.00891  53.57824   1.198 0.236195
## hgp_tot:months   0.18213    0.04901  53.10111   3.716 0.000488 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) hgp_tt months
## hgp_tot     -0.697
## months      -0.975  0.680
## hgp_tt:mnth  0.681 -0.975 -0.699
```

```
plot_model(hgp_lm_2_undsay, type = "pred", terms = c("hgp_tot", "months"))
```



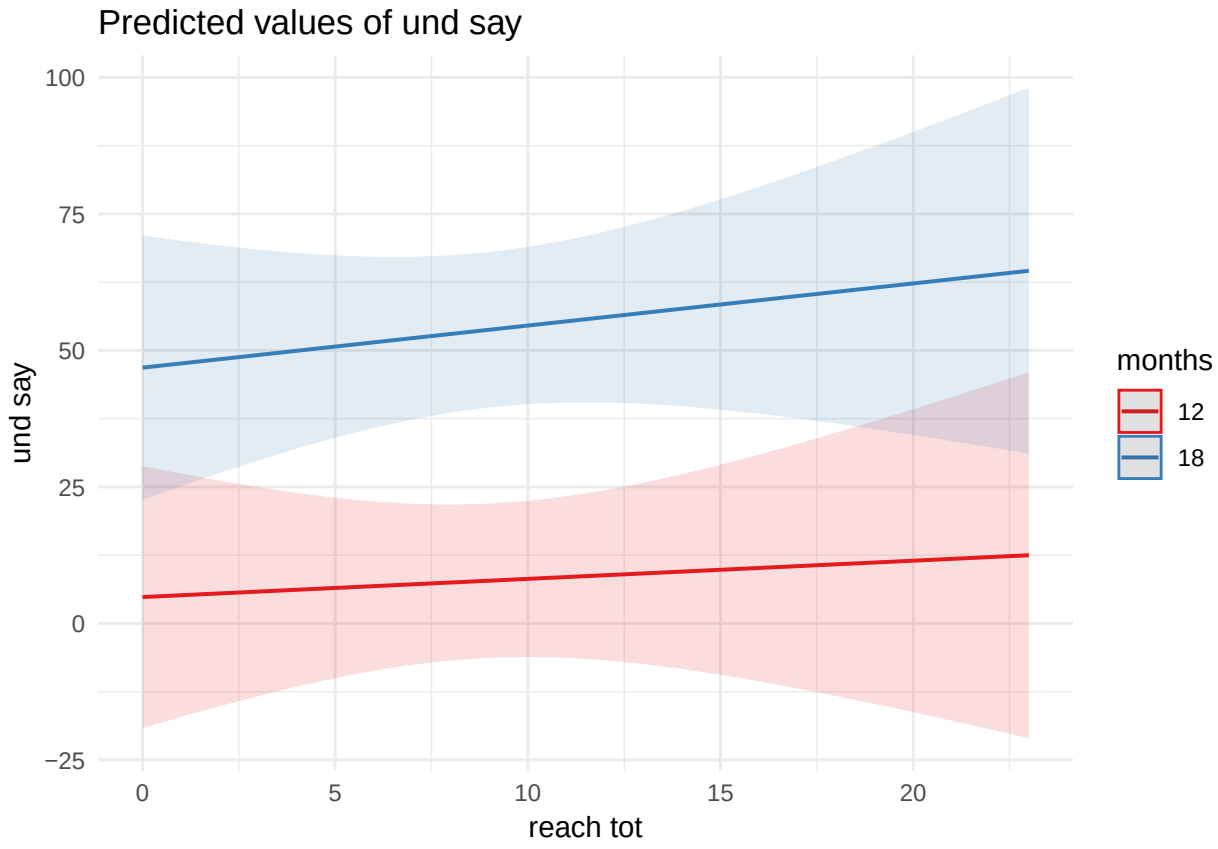
5.2.3 Reaches

```
reach_lm_2_undsay <- lmer(
  und_say ~
    reach_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(reach_lm_2_undsay)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: und_say ~ reach_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1163.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0021 -0.5942 -0.0473  0.0622  4.8782
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept)  301.7   17.37
```

```
##  Residual             2548.9   50.49
## Number of obs: 109, groups:  dyad, 55
##
## Fixed effects:
##                    Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)      -79.18384   42.04927  58.12865  -1.883   0.0647 .
## reach_tot         -0.54333    3.78589  57.92651  -0.144   0.8864
## months             7.00150    2.74140  53.13124   2.554   0.0136 *
## reach_tot:months   0.07302    0.24647  52.79269   0.296   0.7682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rch_tt months
## reach_tot   -0.809
## months      -0.976  0.789
## rch_tt:mnth  0.790 -0.976 -0.809
```

```
plot_model(reach_lm_2_undsay, type = "pred", terms = c("reach_tot", "months"))
```



5.2.4 Maternal utterances

```
utt_lm_2_undsay <- lmer(
  und_say ~
    utt_tot *
    months +
```

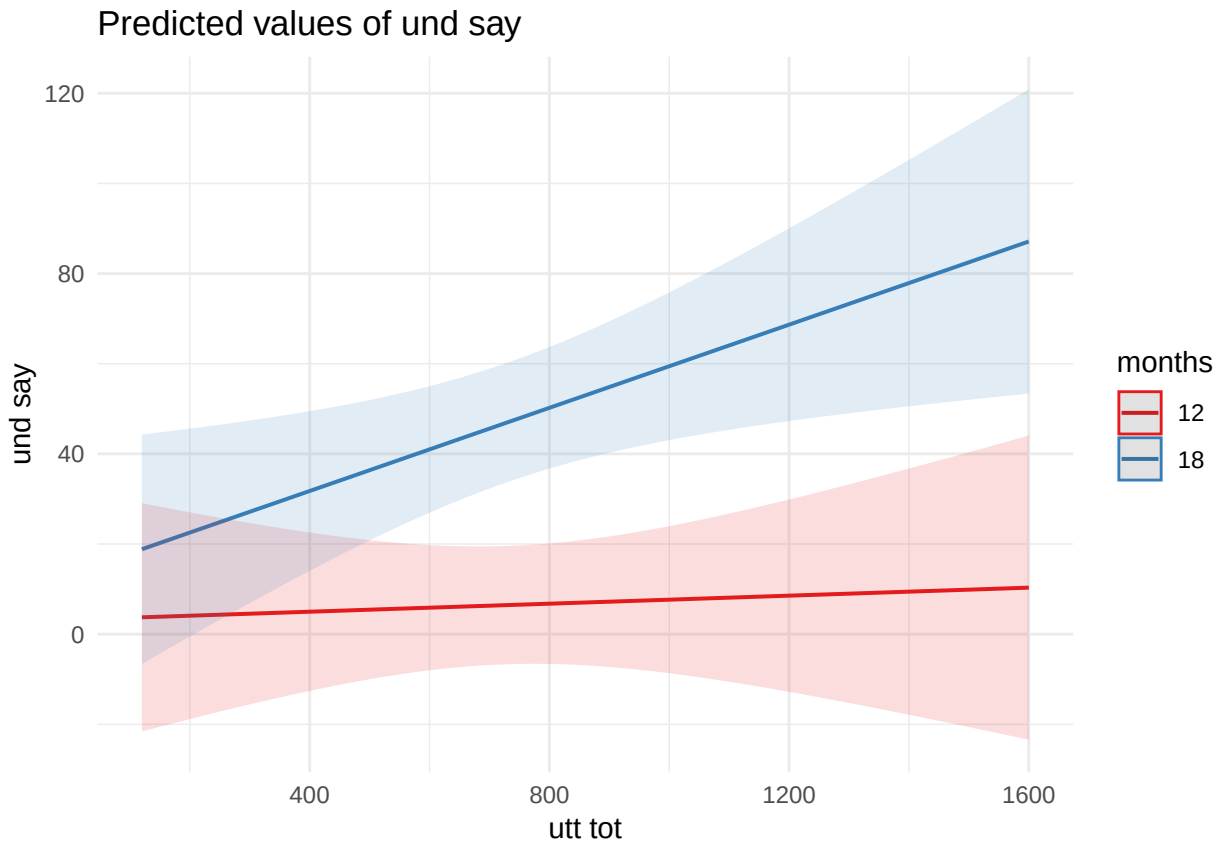
```
    (1|dyad),
  data = vocab
)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
summary(utt_lm_2_undsay)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: und_say ~ utt_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1048.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5455 -0.3786 -0.0439  0.0721  4.9551
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept)  252.7   15.90
##  Residual             1988.0   44.59
## Number of obs: 99, groups:  dyad, 50
##
## Fixed effects:
##                  Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    -17.010657  50.696871  52.550812  -0.336   0.7386
## utt_tot         -0.078926   0.062078  52.447768  -1.271   0.2092
## months           1.684411   3.301037  47.882950   0.510   0.6122
## utt_tot:months   0.006948   0.004039  47.709819   1.720   0.0919 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) utt_tt months
## utt_tot     -0.892
## months      -0.976  0.870
## utt_tt:mnth  0.871 -0.976 -0.892
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```
plot_model(utt_lm_2_undsay, type = "pred", terms = c("utt_tot", "months"))
```



5.2.5 Contingent talks

```
ct_lm_2_undsay <- lmer(
  und_say ~
    ct_tot *
    months +
    (1|dyad),
  data = vocab
)
summary(ct_lm_2_undsay)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: und_say ~ ct_tot * months + (1 | dyad)
##    Data: vocab
##
## REML criterion at convergence: 1121.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4538 -0.4998 -0.0264  0.0844  5.0390
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dyad     (Intercept)  272.8   16.52
```

```
##  Residual             2004.2   44.77
## Number of obs: 107, groups:  dyad, 54
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   -48.6655    25.4770  57.1883  -1.910 0.061130 .
## ct_tot         -4.9476     1.6731  56.9250  -2.957 0.004514 **
## months          4.6434     1.6602  52.1618   2.797 0.007210 **
## ct_tot:months   0.4205     0.1088  51.7189   3.864 0.000312 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ct_tot months
## ct_tot      -0.494
## months      -0.975  0.481
## ct_tt:mnths  0.482 -0.975 -0.494
```

```
plot_model(ct_lm_2_undsay, type = "pred", terms = c("ct_tot", "months"))
```