

Statistical analysis

1 Read data

These chunks read the data and processes it for analysis.

The following reads `gestures.csv` and `utterances.csv` into `gesture_tot` and `utterances_tot`. `gesture_tot` has time series data of infant gestures and maternal Contingent Talks at 10, 11, and 12 months. `utterance_tot` has time series data of maternal utterances at 10, 11, and 12 months. Data is aggregated from the two experimental activities.

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

gesture_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(
    # Needed for GAMs
    back_o = ordered(background, levels = c("English", "Bengali", "Chinese"))
  )

# Needed for GAMs
contrasts(gesture_tot$back_o) <- "contr.treatment"

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

utterances_tot <- utterances %>%
  group_by(dyad, background, months) %>%
  summarise(
    utterances = sum(utterances) # there are NAs that must be kept
  ) %>%
  ungroup() %>%
  mutate_if(is.character, as.factor) %>%
  mutate(
    # Needed for GAMs
    back_o = ordered(background, levels = c("English", "Bengali", "Chinese"))
  )

# Needed for GAMs
contrasts(utterances_tot$back_o) <- "contr.treatment"
```

Here we create individual datasets for HoGs, reaches, pointing, and a dataset with aggregated gestures count and maternal contingent talks (`all_tot`).

```

hg_tot <- filter(gestures_tot, gesture == "ho_gv")
reach_tot <- filter(gestures_tot, gesture == "reach")
point_tot <- filter(gestures_tot, gesture == "point")

# Count = all gestures count, CT is aggregated from all gestures types
all_tot <- gestures_tot %>%
  group_by(dyad, back_o, months) %>%
  summarise(count = sum(count), ct = sum(ct))

```

The following code creates datasets for the analysis of pointing as predicted by HoGs, reaches, maternal CTs, and maternal utterances. The datasets are constructed so that the count of pointing at 11 months is matched with the count of gesture/utterances at 10 months, and the pointing at 12 is matched with the count of gesture/utterances at 11 months. Pointing at 10 months is dropped (since there is no data at 9 months).

```

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)

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)

```

The following creates a dataset with the infants' vocabulary counts and total counts of all gestures, HoGs + point, reaches,

maternal utterances and maternal contingent talks.

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

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

vocab_gest <- gestures_tot %>%
  group_by(dyad, background) %>%
  summarise(count_tot = sum(count), ct_tot = sum(ct)) %>%
  ungroup() %>%
  full_join(y = hgp_tot) %>%
  full_join(y = reach_tot_2) %>%
  mutate_if(is.factor, as.character)

vocab_utt <- utterances_tot %>%
  group_by(dyad, background) %>%
  summarise(utt_tot = sum(utterances)) %>%
  ungroup() %>%
  mutate_if(is.factor, as.character)

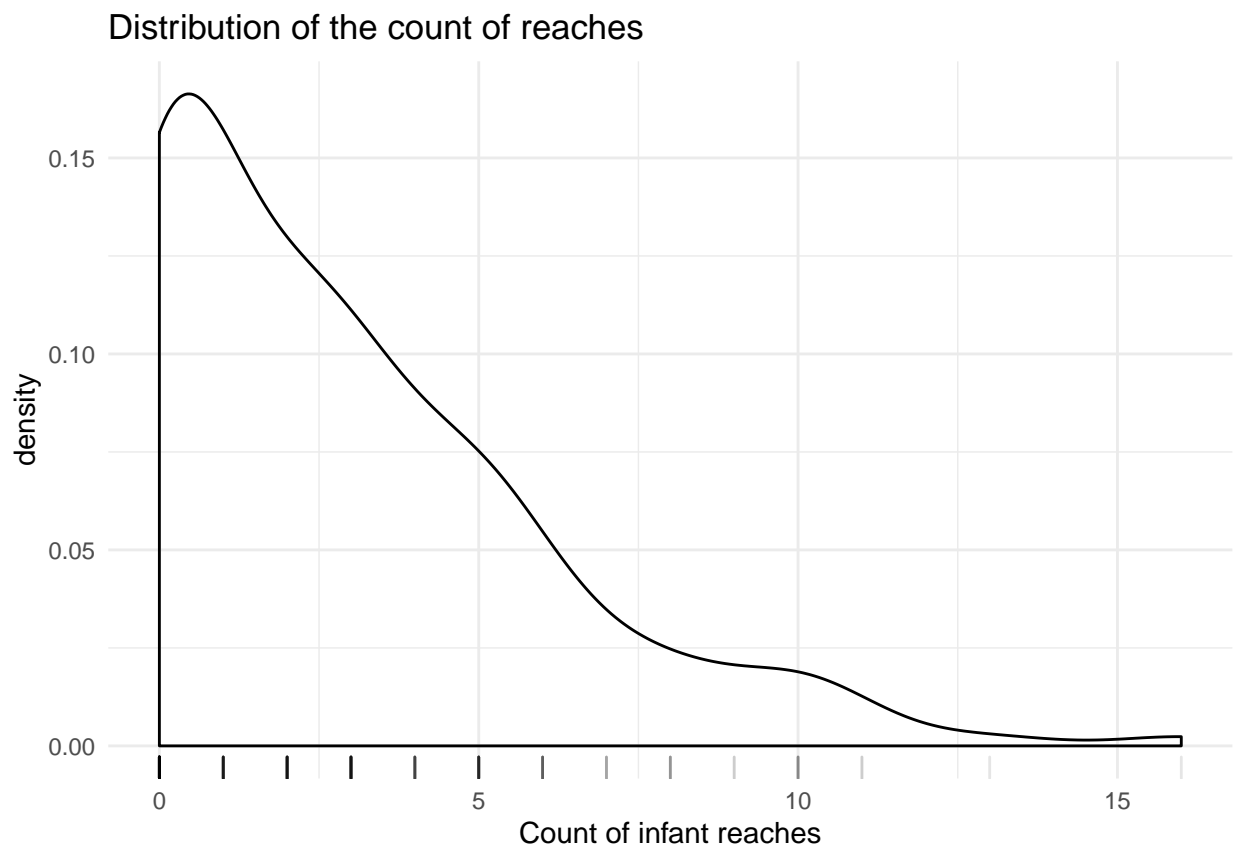
vocab <- read_csv("../data/vocab.csv") %>%
  full_join(y = vocab_gest) %>%
  full_join(y = vocab_utt) %>%
  arrange(dyad, months) %>%
  mutate(
    months = as.factor(months),
    background = factor(background, levels = c("English", "Bengali", "Chinese"))
  ) %>%
  mutate_if(is.character, as.factor)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

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

For analysis 1a, we fitted a series of GAMMs using the negative binomial function. The choice of using the negative binomial rather than the Poisson distribution is justified by the overdispersion of the data (and the very long tail in the distribution). The negative binomial distribution requires the specification of the theta parameter. The parameter has been estimated from the data by fitting a generalised linear model with the negative binomial distribution using MASS : `glm.nb`.

2.1 Reaches development

```
reach_tot %>%  
  ggplot(aes(count)) + geom_density() + geom_rug(alpha = 0.1) +  
  labs(  
    title = "Distribution of the count of reaches",  
    x = "Count of infant reaches"  
  )
```



The following models test cultural group.

```
# Estimation of theta for the negbin() family  
reach_nb <- glm.nb(count ~ months, data = reach_tot)  
theta <- summary(reach_nb)[["theta"]]
```

```
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(theta)
)
```

```
## 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)    0.6375     0.1920   3.321 0.000898 ***
## back_oBengali   0.5874     0.2601   2.258 0.023923 *
## back_oChinese   0.2403     0.2651   0.906 0.364704
## ---
## 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)         1.156     1.287   1.181  0.2853
## s(months):back_oBengali 1.000     1.000   0.437  0.5085
## s(months):back_oChinese 1.000     1.000   0.125  0.7238
## s(months,dyad)      14.522    112.000  20.065  0.0315 *
## ---
## 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(theta)
)
```

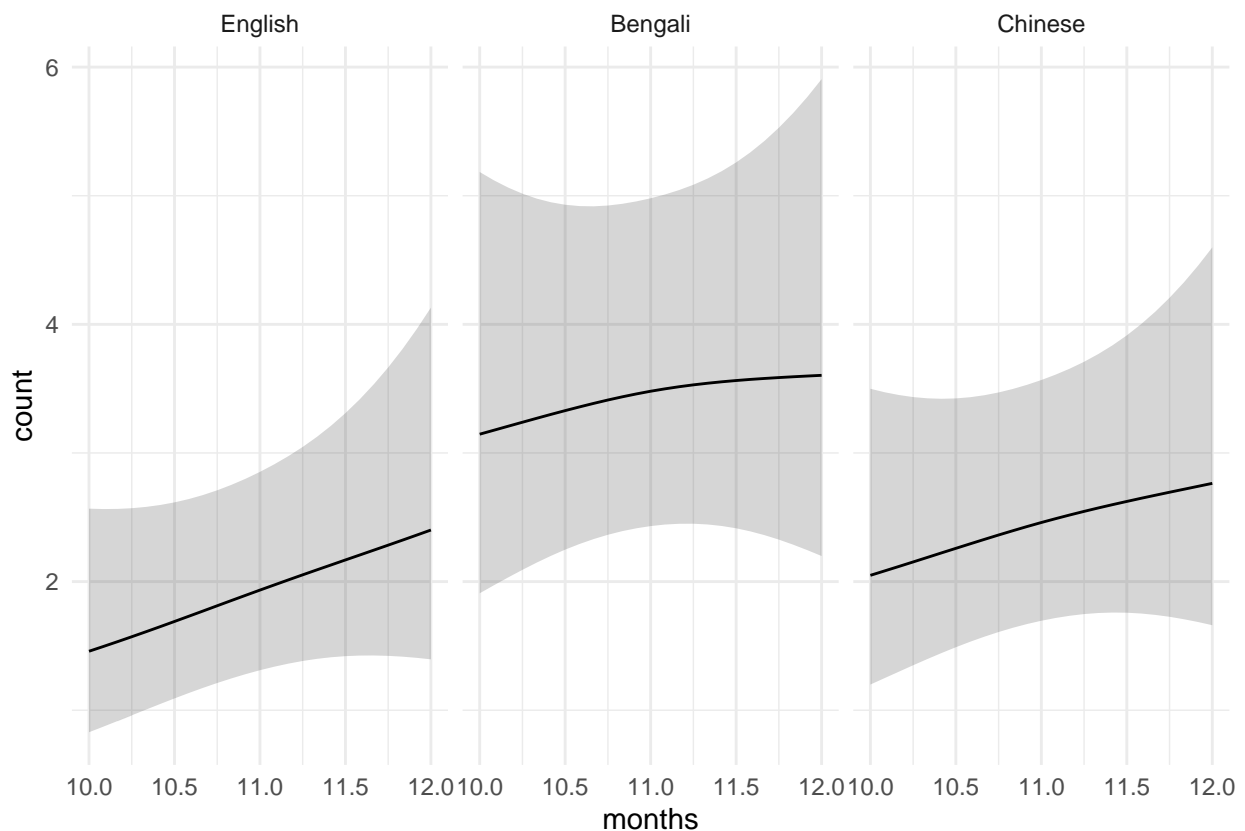
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
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 381.3235   5
## 2      reach_gam 378.5313  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...
```

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



The following models test time sample.

```
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(theta)
)
```

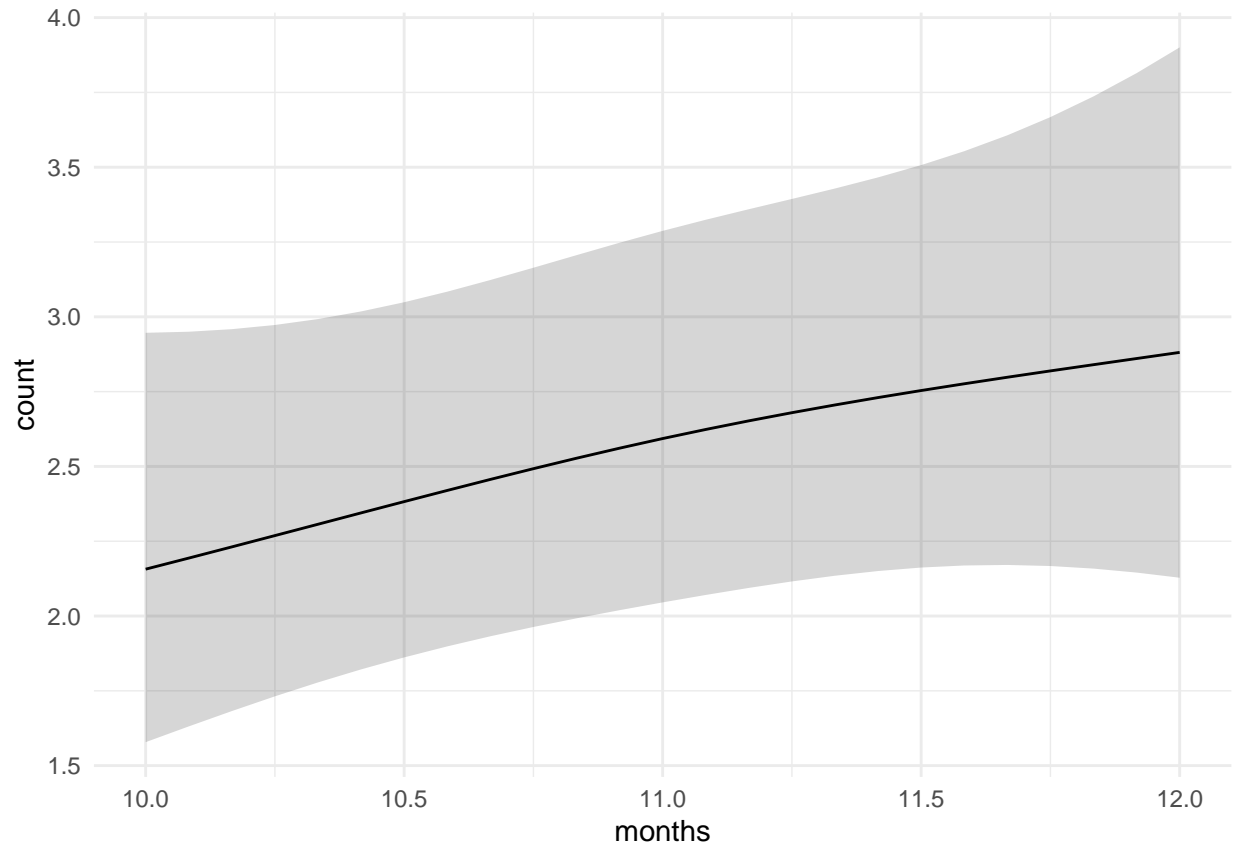
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
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(theta)
)
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 382.1529   3
## 2      reach_gam_2 381.3235   5      0.829 2.000   0.436
##
## AIC difference: -3.95, model reach_gam_2_null has lower AIC.
```

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

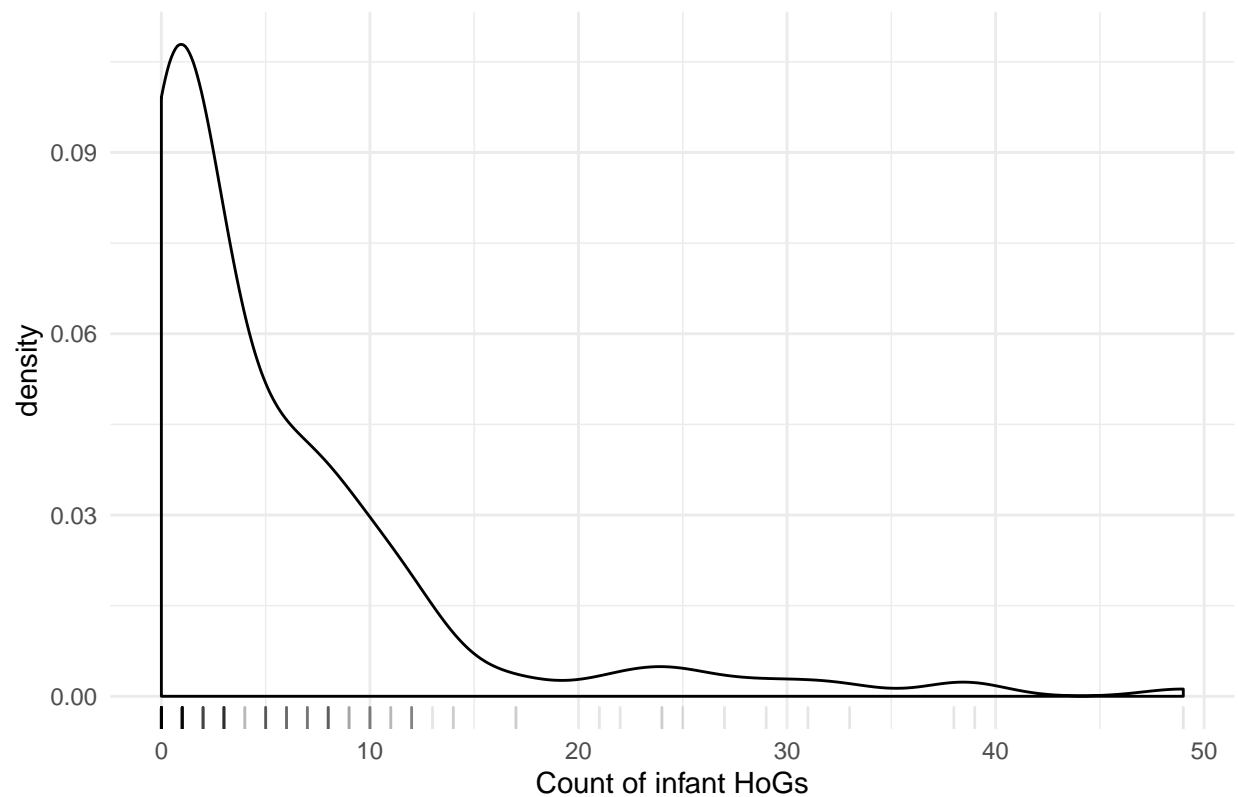
```
plot_smooths(reach_gam_2, months, series_length = 25, transform = exp)
```



2.2 HGs development

```
hg_tot %>%  
  ggplot(aes(count)) + geom_density() + geom_rug(alpha = 0.1) +  
  labs(  
    title = "Distribution of the count of HoGs",  
    x = "Count of infant HoGs"  
  )
```


Distribution of the count of HoGs



The following models test cultural group.

```
hg_nb <- glm.nb(count ~ months, data = hg_tot)
theta_2 <- summary(hg_nb)[["theta"]]
```

```
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(theta_2)
)
```

```
## 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_oBengali  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)          1.00     1  9.708 0.00184 **
## s(months):back_oBengali 1.00     1  0.025 0.87559
## s(months):back_oChinese 1.00     1  0.426 0.51391
## s(months,dyad)      17.71    112 26.332 0.01074 *
## ---
## 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(theta_2)
)
```

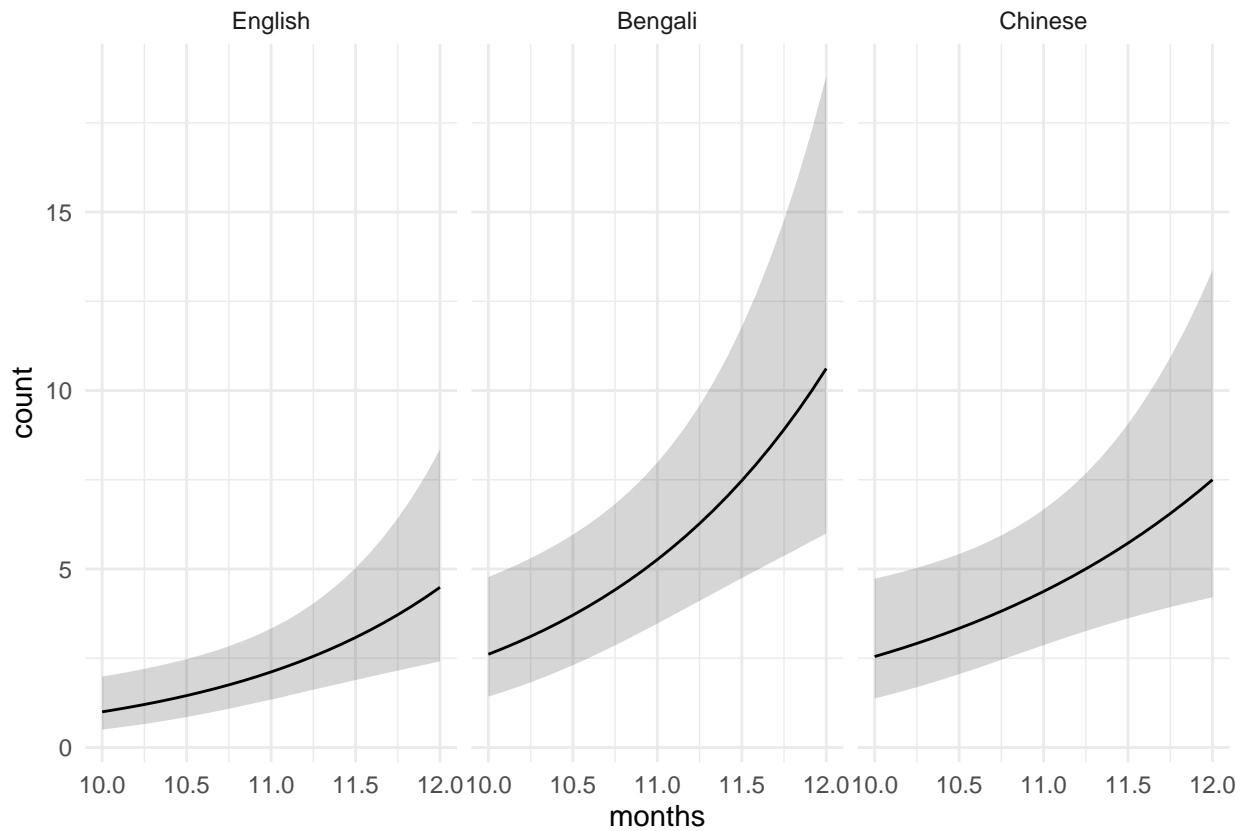
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
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 455.3692    5
## 2 hg_gam 451.0596  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...
```

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



The following models test time sample.

```
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(theta_2)  
)
```

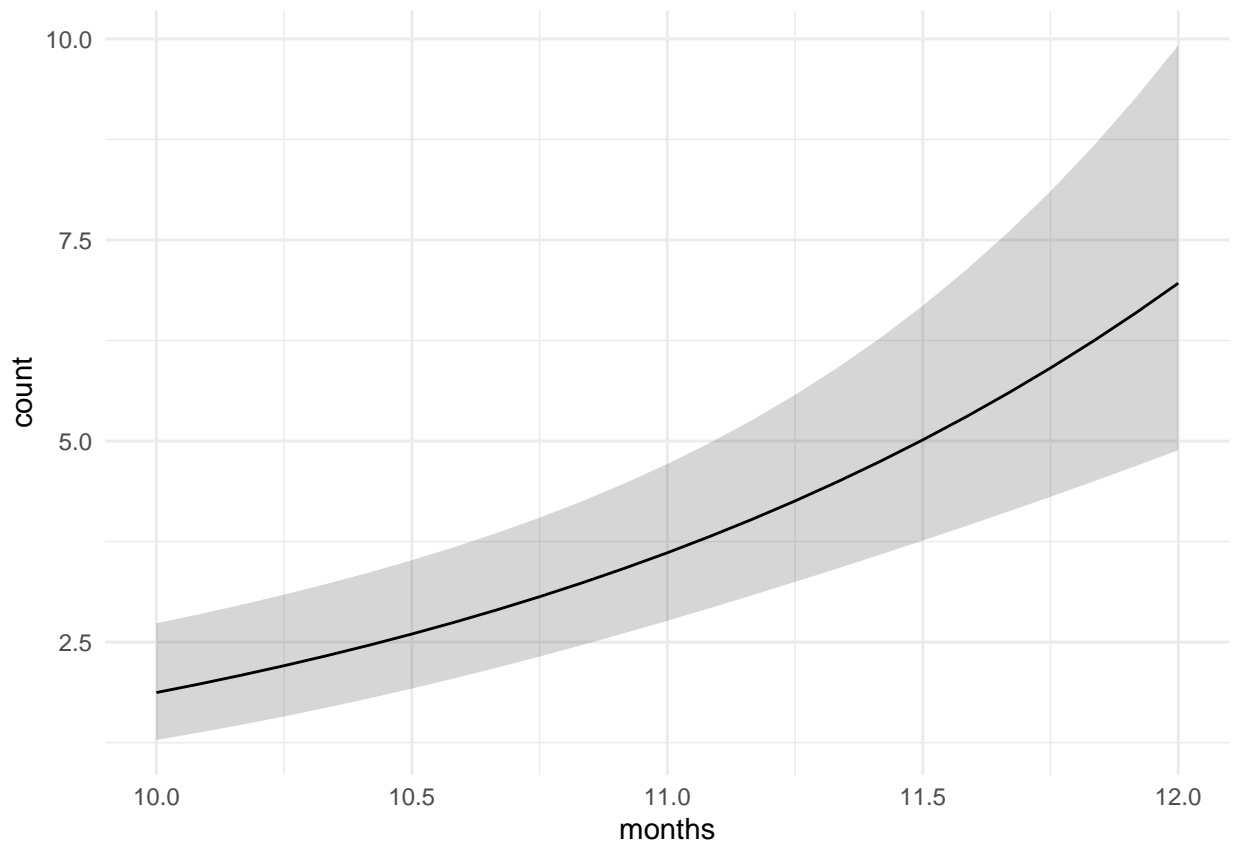
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has  
## repeated 1-d smooths of same variable.
```

```
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(theta_2)
```

```
)
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 467.6971    3
## 2      hg_gam_2 455.3692    5      12.328 2.000 4.427e-06 ***
##
## AIC difference: 29.27, model hg_gam_2 has lower AIC.

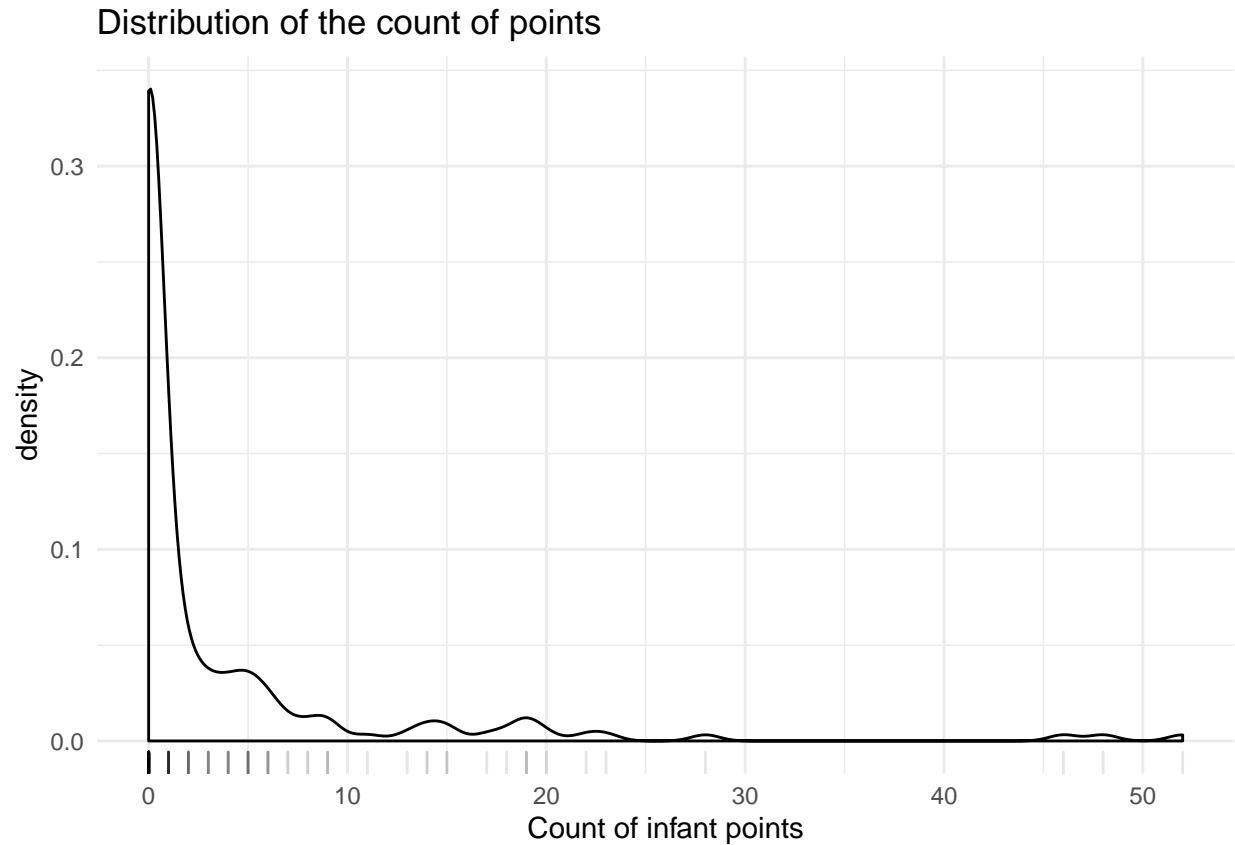
plot_smooths(hg_gam_2, months, series_length = 25, transform = exp)
```



2.3 Points development

```
point_tot %>%
  ggplot(aes(count)) + geom_density() + geom_rug(alpha = 0.1) +
```

```
labs(
  title = "Distribution of the count of points",
  x = "Count of infant points"
)
```



The following models test cultural group.

```
point_nb <- glm.nb(count ~ months, data = point_tot)
theta_3 <- summary(point_nb)[["theta"]]
```

```
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(theta_3)
)
```

```
## 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)    0.6919    0.3953   1.750  0.0801 .
## back_oBengali -0.4994    0.5588  -0.894  0.3715
## 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_oBengali 1.538     1.786   0.726  0.5737
## s(months):back_oChinese 1.000     1.000   2.118  0.1456
## s(months,dyad)      18.368    112.000  25.998  0.0225 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.332   Deviance explained =  41%
## -ML = 326.24   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(theta_3)
)
```

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

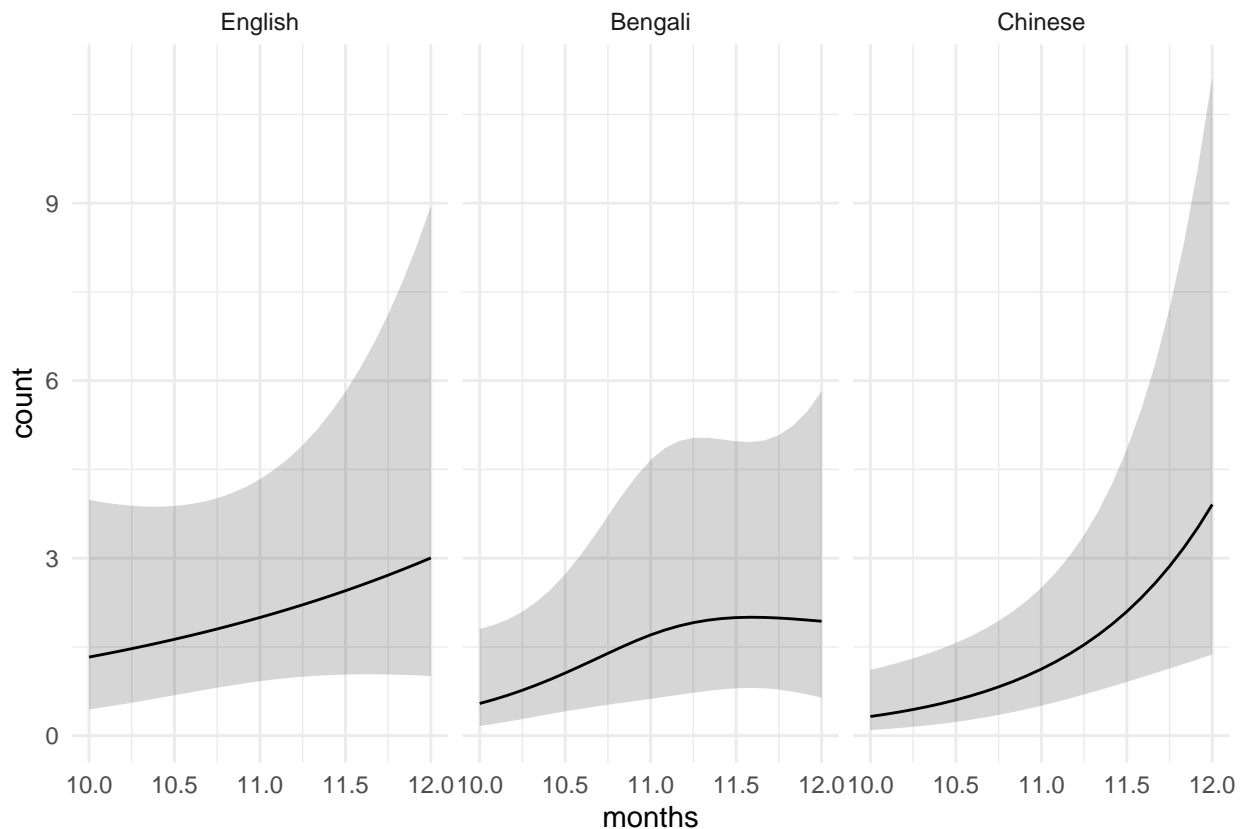
```
compareML(point_gam_null, point_gam)
```

```
## 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 327.9371    5
## 2      point_gam 326.2371   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...

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



The following models test time sample.

```
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(theta_3)
)
```

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

```

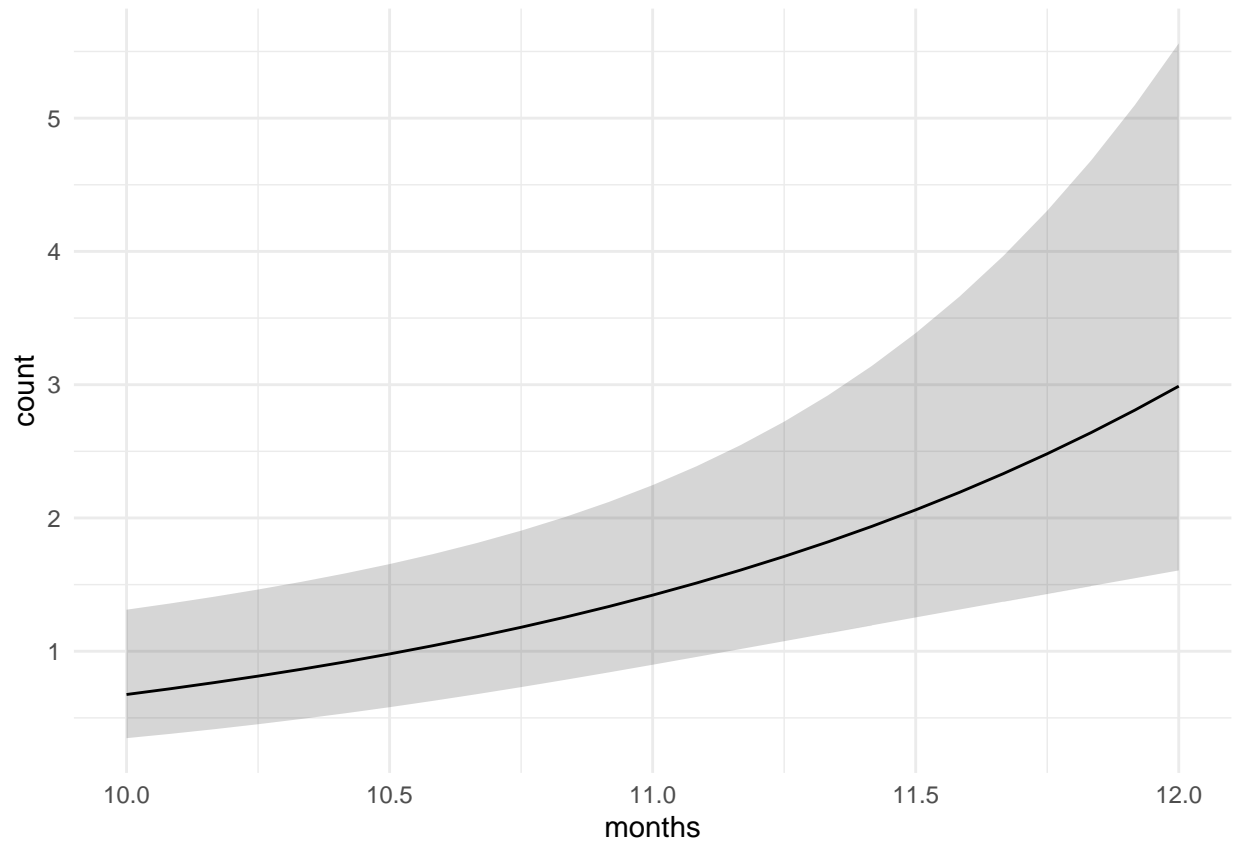
point_gam_2_null <- gam(
  count ~
    # s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
  method = "ML",
  family = negbin(theta_3)
)
compareML(point_gam_2_null, point_gam_2)

## 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 332.5523   3
## 2   point_gam_2 327.9371   5      4.615 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_2, months, series_length = 25, transform = exp)

```

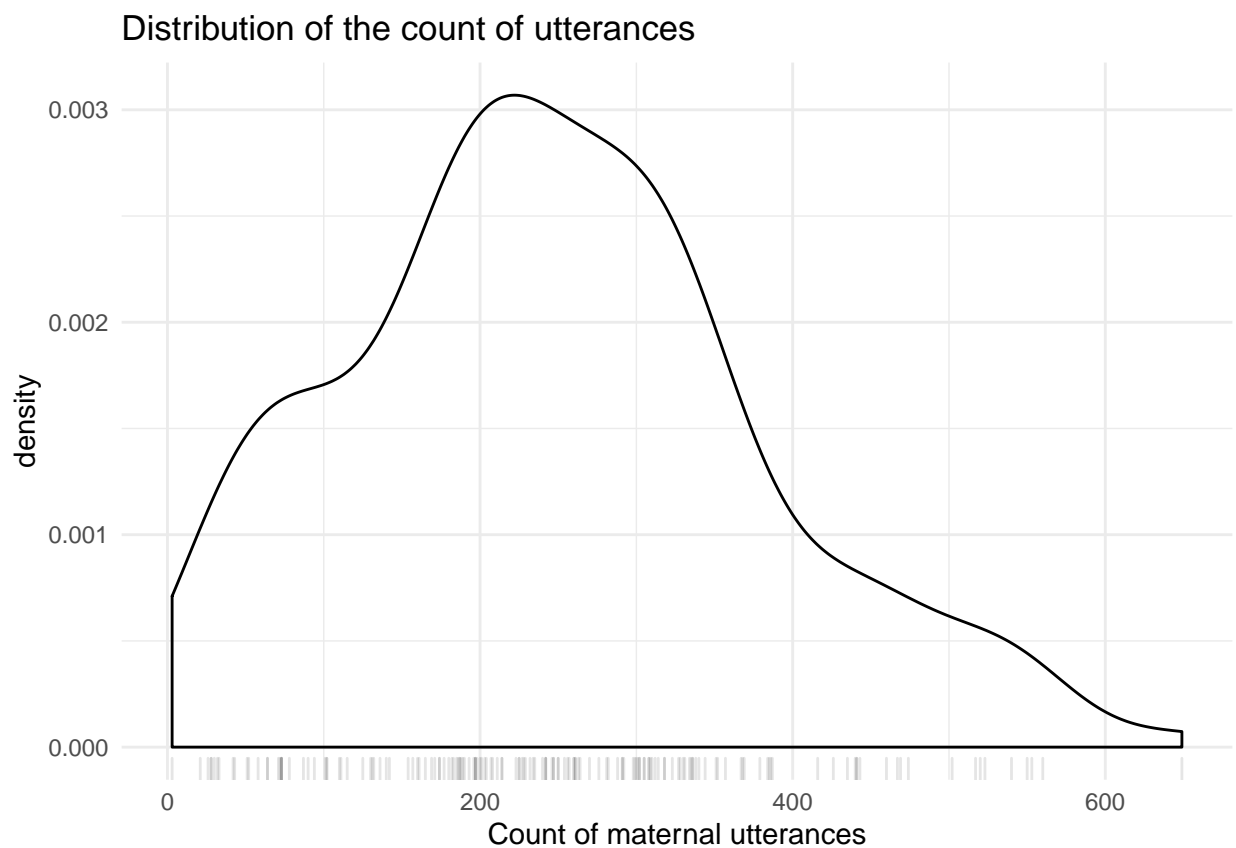



3 Analysis 1b. Frequency of maternal utterances and contingent talk to infants aged 10-12 months.

For maternal utterances we used a normal distribution, since the distribution of the data was almost normal. For maternal contingent talks instead we used again the negative binomial distribution for the same reasons as above.

3.1 Maternal utterances development

```
utterances_tot %>%  
  ggplot(aes(utterances)) + geom_density() + geom_rug(alpha = 0.1) +  
  labs(  
    title = "Distribution of the count of utterances",  
    x = "Count of maternal utterances"  
  )
```



The following models test cultural group.

```
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_oBengali   -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_oBengali 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.

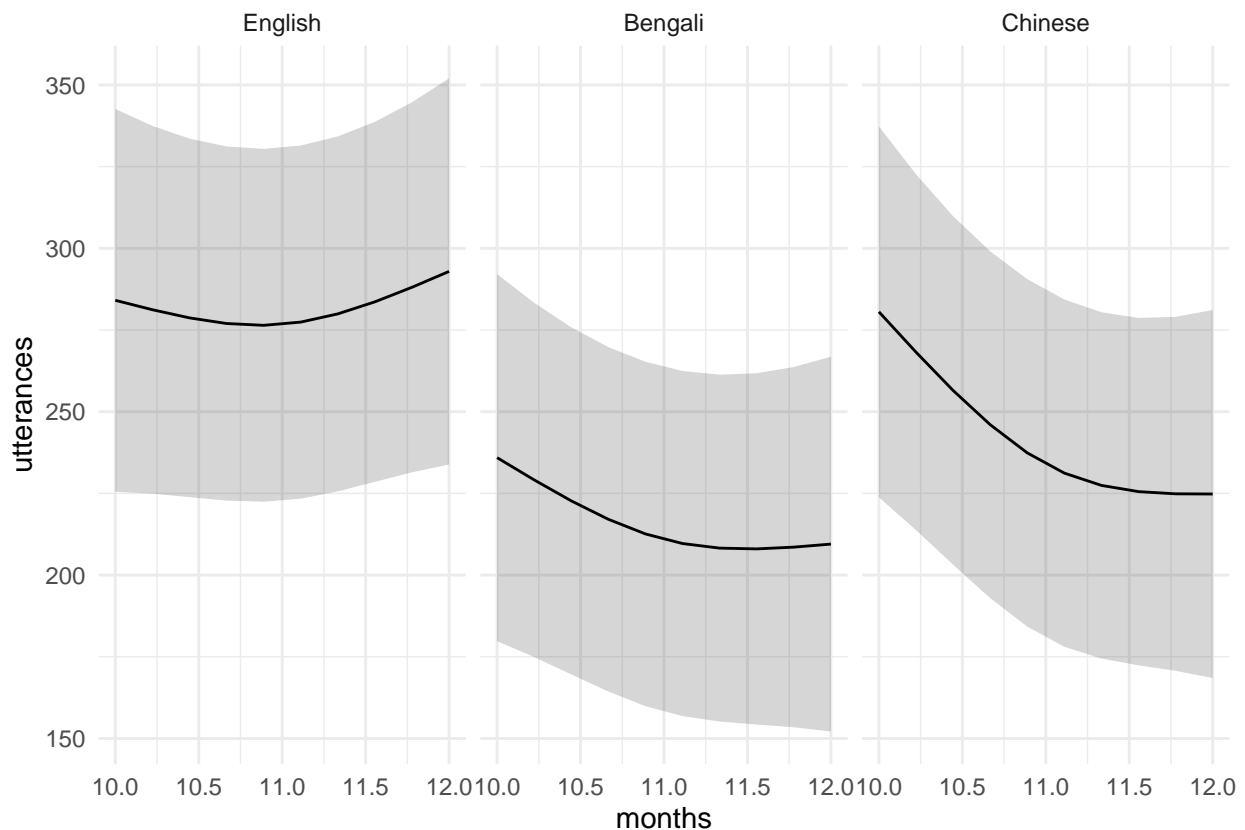
```

```
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...
```

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



The following models test time sample.

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

```

    s(months, k = 3) +
    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.

```

```

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

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 997.9664   3
## 2   utter_gam_2 995.3291   5      2.637 2.000  0.072
##
## AIC difference: 6.07, model utter_gam_2 has lower AIC.

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

```

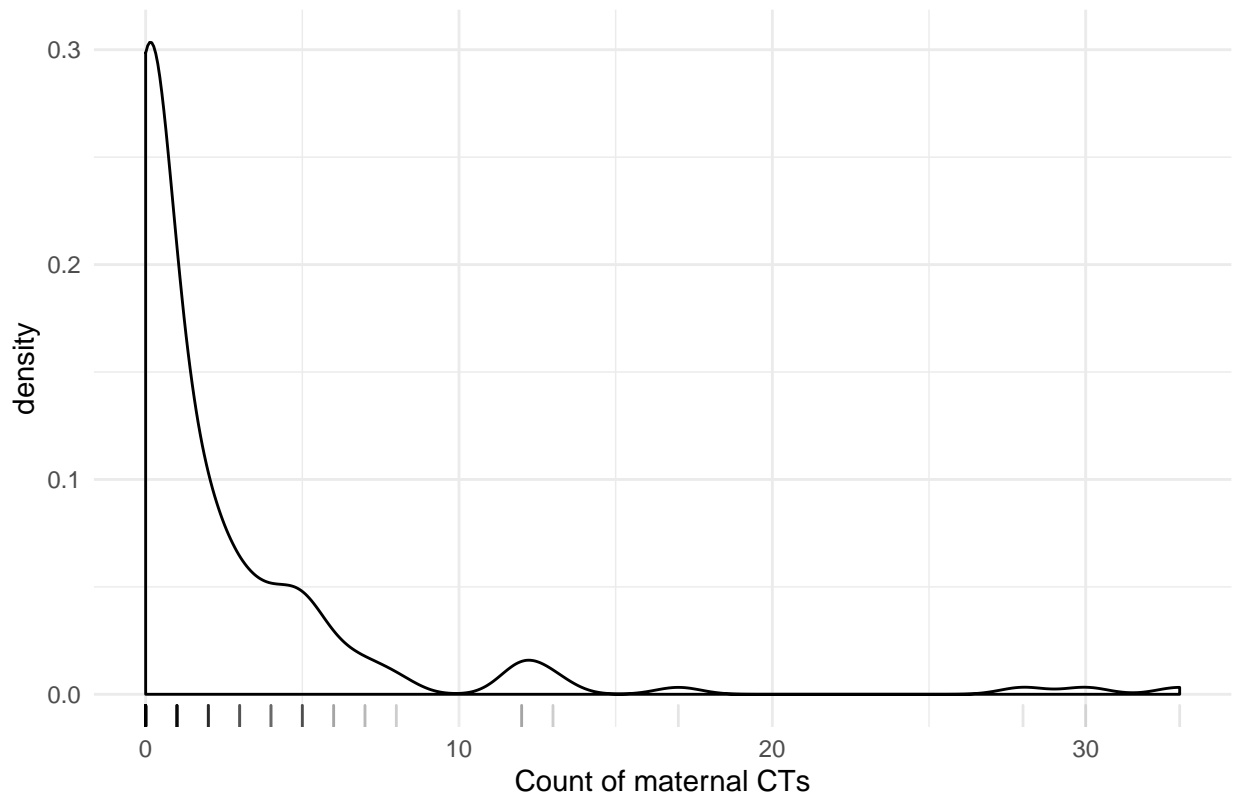
3.2 Contingent talks development

```

all_tot %>%
  ggplot(aes(ct)) + geom_density() + geom_rug(alpha = 0.1) +
  labs(
    title = "Distribution of the count of CTs",
    x = "Count of maternal CTs"
  )

```

Distribution of the count of CTs



The following models test cultural group.

```
ct_nb <- glm.nb(ct ~ months, data = all_tot)
theta_4 <- summary(ct_nb)[["theta"]]

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",
  family = negbin(theta_4)
)
```

```
## 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: Negative Binomial(0.385)
## Link function: log
##
```

```
## 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 z value Pr(>|z|)
## (Intercept)    0.6527    0.2977   2.192  0.0283 *
## back_oBengali -0.9863    0.4347  -2.269  0.0233 *
## back_oChinese -0.2083    0.4226  -0.493  0.6222
## ---
## 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)         1.00     1.000   3.039 0.08129 .
## s(months):back_oBengali 1.75     1.937   3.064 0.24022
## s(months):back_oChinese 1.00     1.000   0.391 0.53191
## s(months,dyad)      18.38    112.000  27.602 0.00937 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.394   Deviance explained = 43.7%
## -ML = 315.49   Scale est. = 1           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",
  family = negbin(theta_4)
)
```

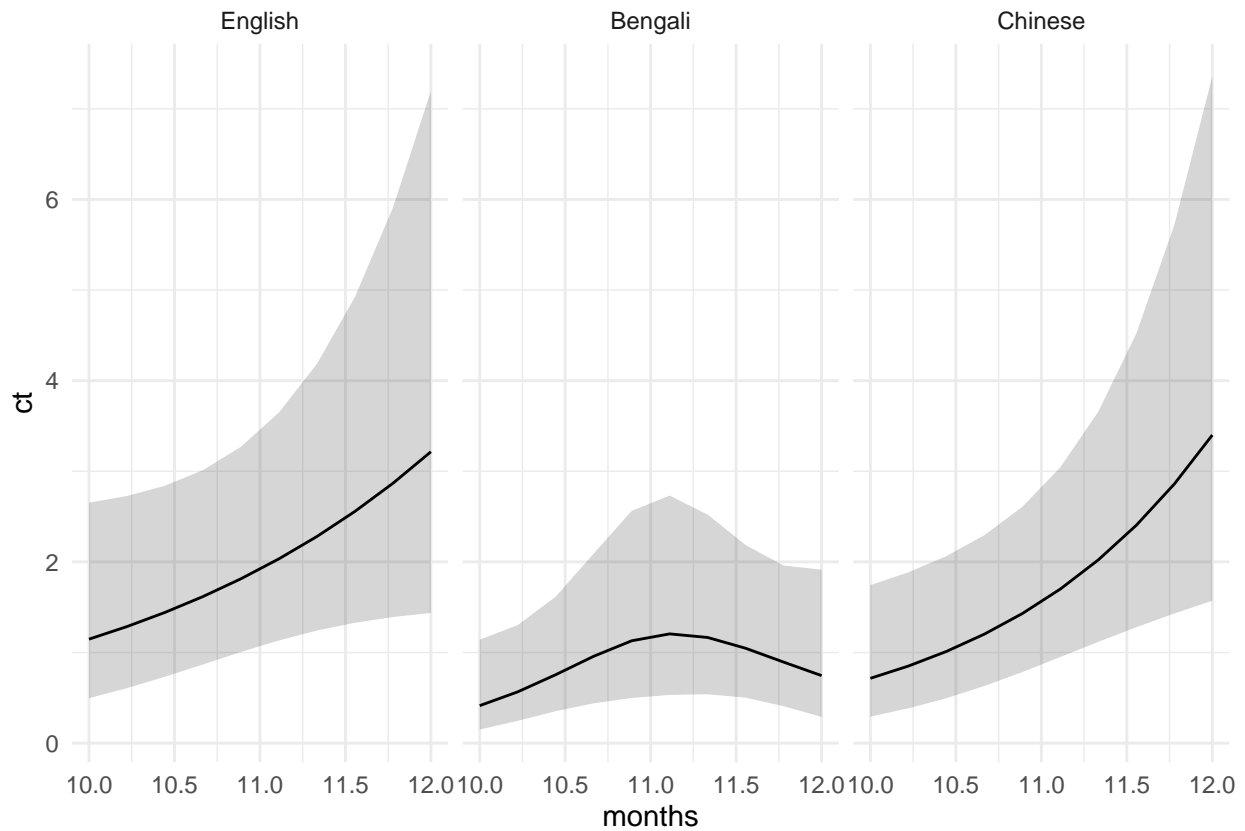
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
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 318.9134    5
## 2      ct_gam 315.4851   11      3.428 6.000    0.334
##
## AIC difference: 0.60, model ct_gam has lower AIC.
```

```
## Warning in compareML(ct_gam_null, ct_gam): Only small difference in ML...
```

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



The following models test time sample.

```
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(theta_4)
)
```

```
## 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(theta_4)
)
```



```
)
```

```
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 641.7134    3
```

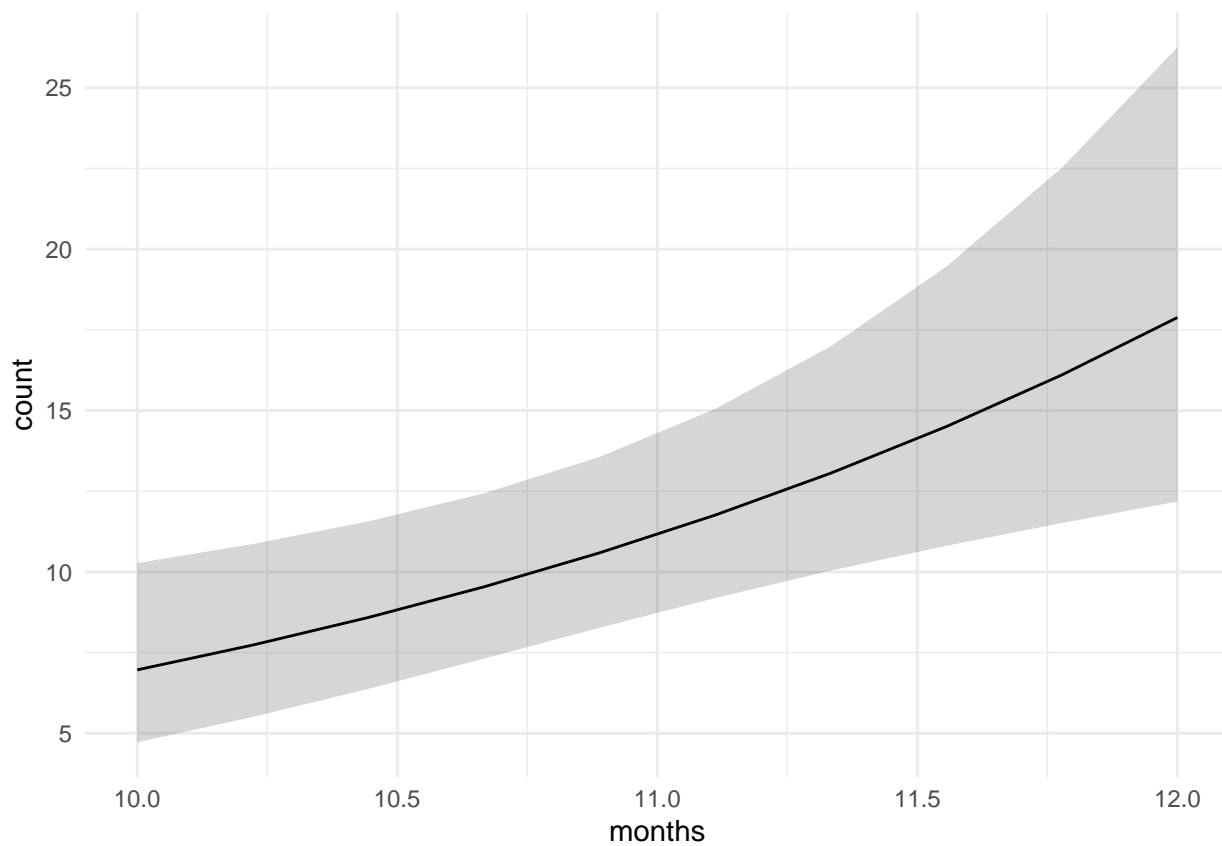
```
## 2      ct_gam_2 637.2323    5      4.481 2.000  0.011  *
```

```
##
```

```
## AIC difference: 6.96, 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_2, months, series_length = 10, transform = exp)
```



4 Analysis 1c. Predictors of pointing at 12 months

The following GLMMs test the relation between pointing as the outcome variable and reaches/HoGs.

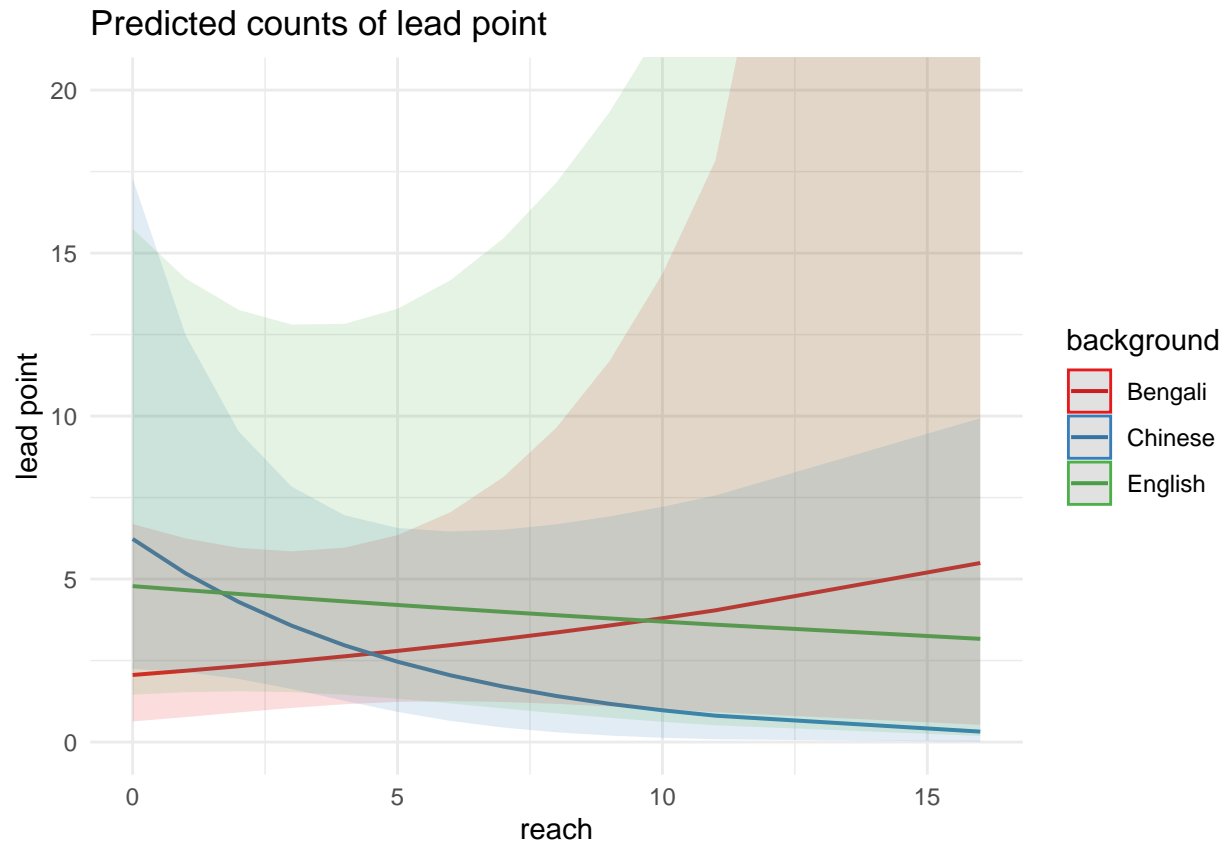
4.1 Reaches

```
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_model(reach_point_lm, type = "pred", terms = c("reach", "background")) + coord_cartesian(ylim = c(0, 20))
```



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)
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(hg_point_lm)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(0.261) ( log )
```

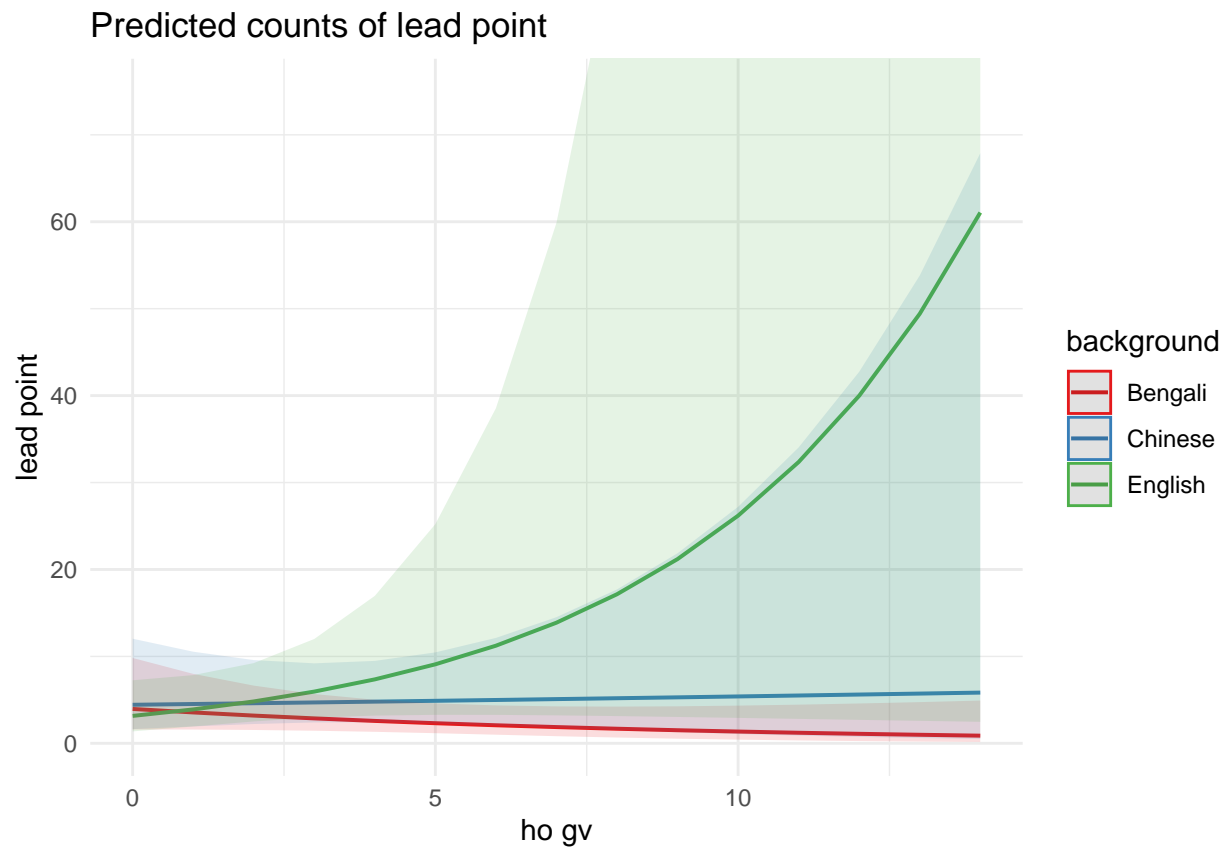
```
## Formula: lead_point ~ ho_gv * background + (1 | dyad)
## Data: filter(hg_point_lead, ho_gv < 20)
##
##      AIC      BIC    logLik deviance df.resid
##    503.8    525.3   -243.9   487.8     101
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.5080 -0.4942 -0.3979  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
## boundary (singular) fit: see ?isSingular
```

```
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_model(hg_point_lm, type = "pred", terms = c("ho_gv", "background")) + coord_cartesian(ylim = c(0,
```



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

5.1 Comprehension at 12 and 18 months

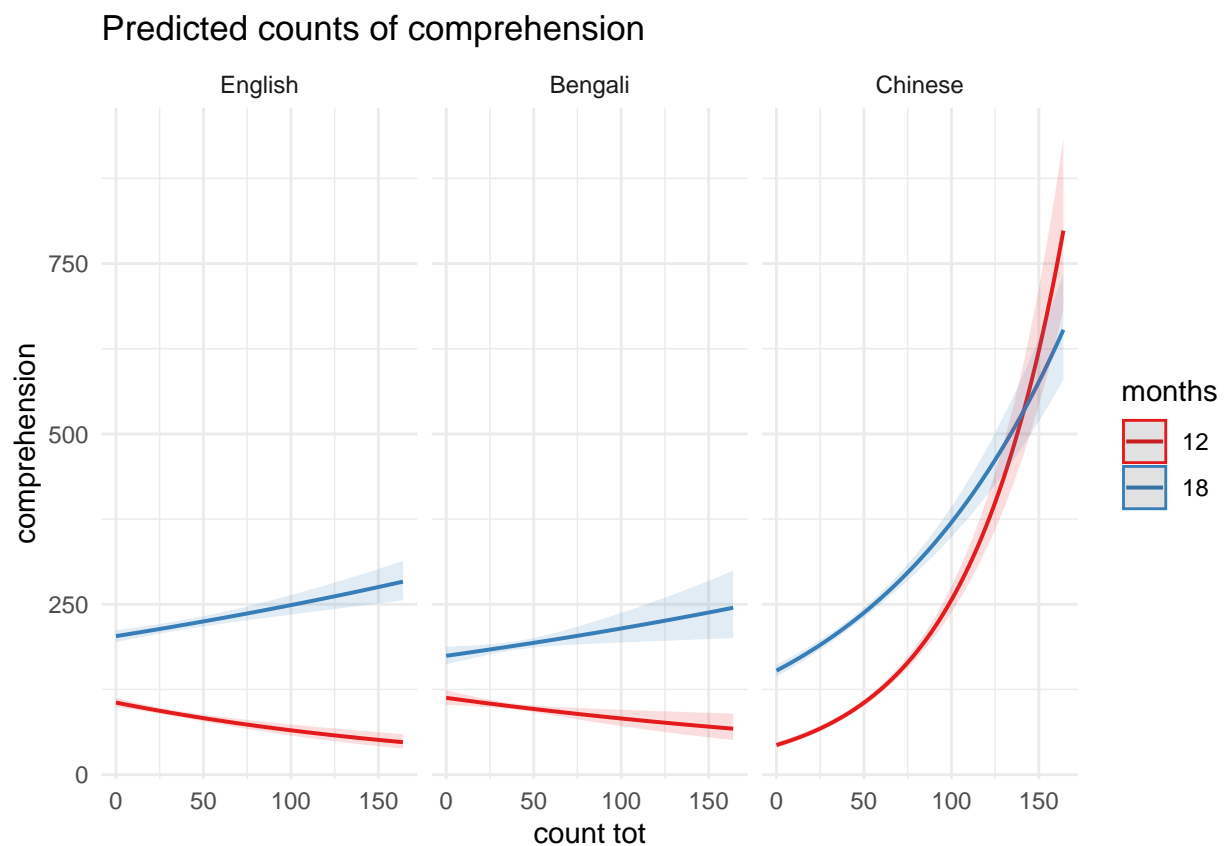
5.1.1 All gestures combined

```
all_gest_lm <- glm(
  comprehension ~
    count_tot *
    months *
    background,
  data = vocab,
  family = poisson
)
summary(all_gest_lm)
```

```
##
## Call:
## glm(formula = comprehension ~ count_tot * months * background,
##      family = poisson, data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -12.534   -3.589   -0.588    2.339   16.033
##
## Coefficients:
##                                Estimate Std. Error z value
## (Intercept)                   4.6615113  0.0323369 144.155
## count_tot                    -0.0048625  0.0007949  -6.117
## months18                      0.6528243  0.0389823  16.747
## backgroundBengali             0.0635196  0.0586909   1.082
## backgroundChinese            -0.8899846  0.0570165 -15.609
## count_tot:months18            0.0068861  0.0008836   7.793
## count_tot:backgroundBengali    0.0017305  0.0013854   1.249
## count_tot:backgroundChinese    0.0226132  0.0010689  21.155
## months18:backgroundBengali    -0.2163911  0.0726471  -2.979
## months18:backgroundChinese     0.6034030  0.0675504   8.933
## count_tot:months18:backgroundBengali -0.0016836  0.0016536  -1.018
## count_tot:months18:backgroundChinese -0.0157740  0.0012443 -12.677
##                                Pr(>|z|)
## (Intercept)                   < 2e-16 ***
## count_tot                     9.53e-10 ***
## months18                      < 2e-16 ***
## backgroundBengali             0.2791
## backgroundChinese             < 2e-16 ***
## count_tot:months18            6.52e-15 ***
## count_tot:backgroundBengali    0.2116
## count_tot:backgroundChinese    < 2e-16 ***
## months18:backgroundBengali     0.0029 **
## months18:backgroundChinese     < 2e-16 ***
## count_tot:months18:backgroundBengali 0.3086
## count_tot:months18:backgroundChinese < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 6642.1  on 108  degrees of freedom
## Residual deviance: 3326.7  on  97  degrees of freedom
## (11 observations deleted due to missingness)
## AIC: 4072.7
##
## Number of Fisher Scoring iterations: 5
```

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



5.1.2 HoGs + points

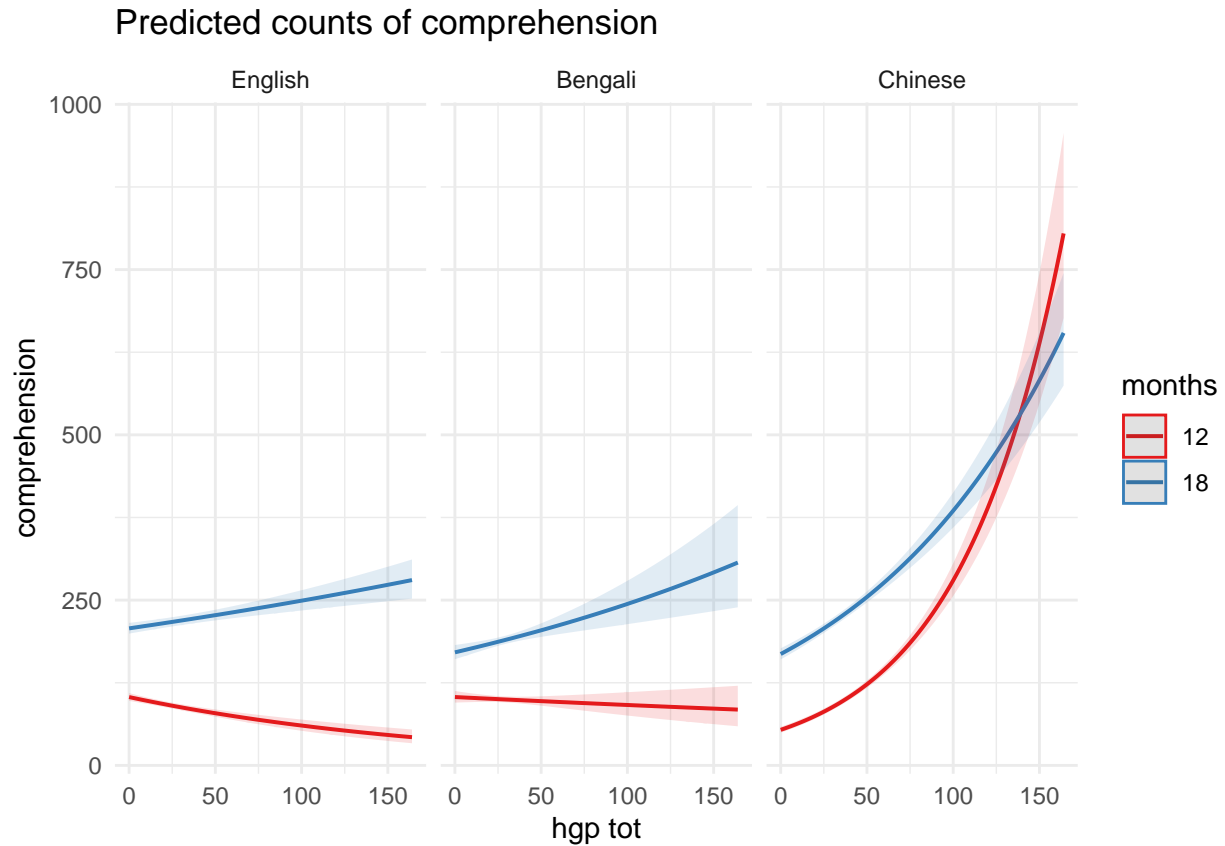
```
hgp_lm <- glm(
  comprehension ~
    hgp_tot *
    months *
    background,
  data = vocab,
  family = poisson())
```

```
)
summary(hgp_lm)
```

```
##
## Call:
## glm(formula = comprehension ~ hgp_tot * months * background,
##      family = poisson(), data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0912   -4.0304   -0.3296    2.6629   17.5976
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.6377993   0.0292714 158.441 < 2e-16
## hgp_tot          -0.0053955   0.0008347  -6.464 1.02e-10
## months18          0.6962163   0.0354287  19.651 < 2e-16
## backgroundBengali  0.0003589   0.0522614   0.007 0.99452
## backgroundChinese -0.6535607   0.0506378 -12.907 < 2e-16
## hgp_tot:months18   0.0072361   0.0009178   7.885 3.16e-15
## hgp_tot:backgroundBengali 0.0041596   0.0015628   2.662 0.00778
## hgp_tot:backgroundChinese 0.0218970   0.0011069  19.782 < 2e-16
## months18:backgroundBengali -0.1927418   0.0644893  -2.989 0.00280
## months18:backgroundChinese  0.4461919   0.0600587   7.429 1.09e-13
## hgp_tot:months18:backgroundBengali -0.0024355   0.0018614  -1.308 0.19074
## hgp_tot:months18:backgroundChinese -0.0154648   0.0012786 -12.095 < 2e-16
##
## (Intercept)          ***
## hgp_tot              ***
## months18            ***
## backgroundBengali
## backgroundChinese   ***
## hgp_tot:months18    ***
## hgp_tot:backgroundBengali **
## hgp_tot:backgroundChinese ***
## months18:backgroundBengali **
## months18:backgroundChinese ***
## hgp_tot:months18:backgroundBengali
## hgp_tot:months18:backgroundChinese ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 6642.1  on 108  degrees of freedom
## Residual deviance: 3459.2  on  97  degrees of freedom
## (11 observations deleted due to missingness)
## AIC: 4205.2
##
## Number of Fisher Scoring iterations: 5
```



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



5.1.3 Reaches

```
reach_lm <- glm(
  comprehension ~
    reach_tot *
    months *
    background,
  data = vocab,
  family = poisson()
)
summary(reach_lm)
```

```
##
## Call:
## glm(formula = comprehension ~ reach_tot * months * background,
##      family = poisson(), data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0397  -4.2179  -0.4216   3.4316  17.2738
##
```

```

## Coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.456258   0.036398 122.433 < 2e-16
## reach_tot        0.008853   0.003627   2.441 0.01465
## months18         0.892274   0.043543  20.492 < 2e-16
## backgroundBengali 0.384841   0.057007   6.751 1.47e-11
## backgroundChinese -0.308676   0.060521  -5.100 3.39e-07
## reach_tot:months18 -0.003657   0.004337  -0.843 0.39916
## reach_tot:backgroundBengali -0.030842   0.005119  -6.024 1.70e-09
## reach_tot:backgroundChinese 0.042672   0.005541   7.701 1.35e-14
## months18:backgroundBengali -0.424152   0.069985  -6.061 1.36e-09
## months18:backgroundChinese 0.187235   0.071705   2.611 0.00902
## reach_tot:months18:backgroundBengali 0.019788   0.006201   3.191 0.00142
## reach_tot:months18:backgroundChinese -0.026341   0.006649  -3.961 7.45e-05
##
## (Intercept)      ***
## reach_tot        *
## months18         ***
## backgroundBengali ***
## backgroundChinese ***
## reach_tot:months18
## reach_tot:backgroundBengali ***
## reach_tot:backgroundChinese ***
## months18:backgroundBengali ***
## months18:backgroundChinese **
## reach_tot:months18:backgroundBengali **
## reach_tot:months18:backgroundChinese ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##   Null deviance: 6642.1  on 108  degrees of freedom
## Residual deviance: 4011.7  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 4757.8
##
## Number of Fisher Scoring iterations: 5

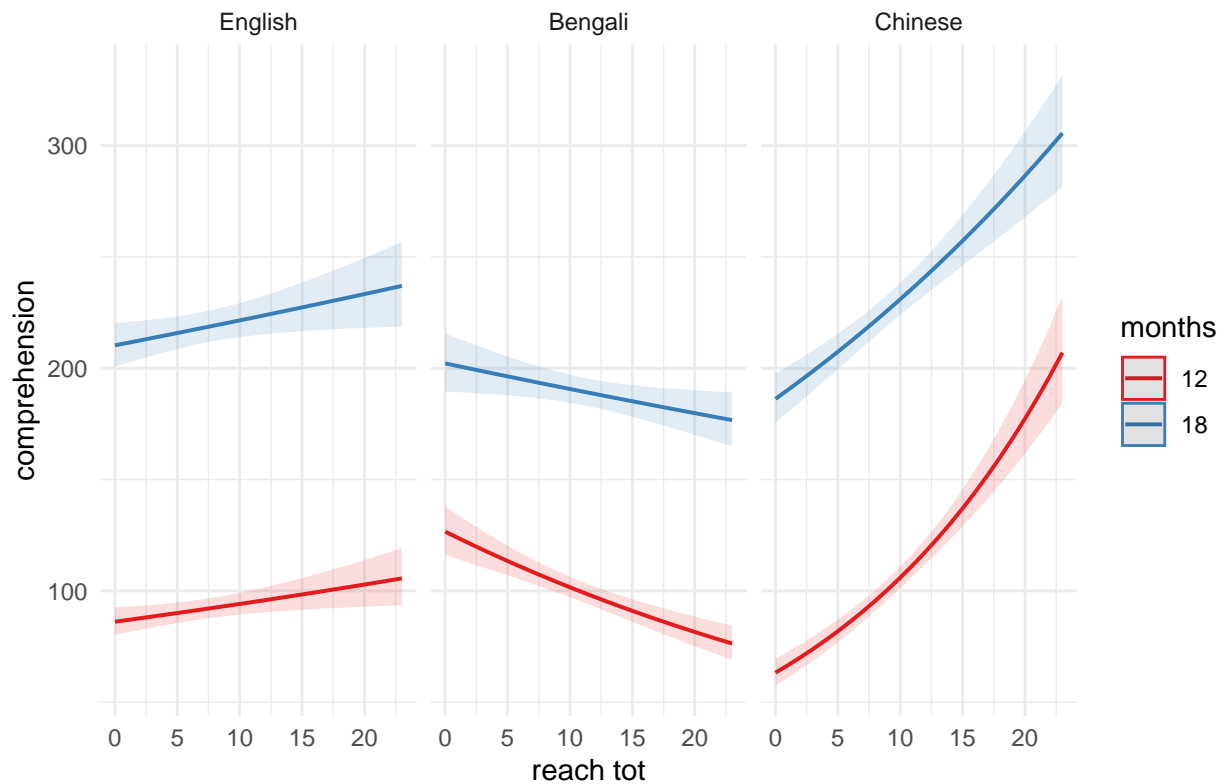
```

```

plot_model(reach_lm, type = "pred", terms = c("reach_tot", "months", "background"))

```

Predicted counts of comprehension



5.1.4 Maternal utterances

```
utt_lm <- glm(
  comprehension ~
    utt_tot *
    months *
    background,
  data = vocab,
  family = poisson()
)
summary(utt_lm)
```

```
##
## Call:
## glm(formula = comprehension ~ utt_tot * months * background,
##      family = poisson(), data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -14.146   -4.714   -1.202    3.528   16.309
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.1169916  0.1184273  51.652  < 2e-16
```

```

## utt_tot -0.0018179 0.0001481 -12.277 < 2e-16
## months18 -0.1830862 0.1463324 -1.251 0.211
## backgroundBengali -1.6180843 0.1253368 -12.910 < 2e-16
## backgroundChinese -1.9734461 0.1306492 -15.105 < 2e-16
## utt_tot:months18 0.0011224 0.0001787 6.281 3.37e-10
## utt_tot:backgroundBengali 0.0020650 0.0001566 13.186 < 2e-16
## utt_tot:backgroundChinese 0.0023984 0.0001599 14.996 < 2e-16
## months18:backgroundBengali 0.8579418 0.1548619 5.540 3.02e-08
## months18:backgroundChinese 1.2385528 0.1605040 7.717 1.19e-14
## utt_tot:months18:backgroundBengali -0.0011869 0.0001896 -6.259 3.87e-10
## utt_tot:months18:backgroundChinese -0.0014408 0.0001932 -7.457 8.82e-14
##
## (Intercept) ***
## utt_tot ***
## months18
## backgroundBengali ***
## backgroundChinese ***
## utt_tot:months18 ***
## utt_tot:backgroundBengali ***
## utt_tot:backgroundChinese ***
## months18:backgroundBengali ***
## months18:backgroundChinese ***
## utt_tot:months18:backgroundBengali ***
## utt_tot:months18:backgroundChinese ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 5914.0 on 98 degrees of freedom
## Residual deviance: 3642.8 on 87 degrees of freedom
## (21 observations deleted due to missingness)
## AIC: 4324.5
##
## Number of Fisher Scoring iterations: 5

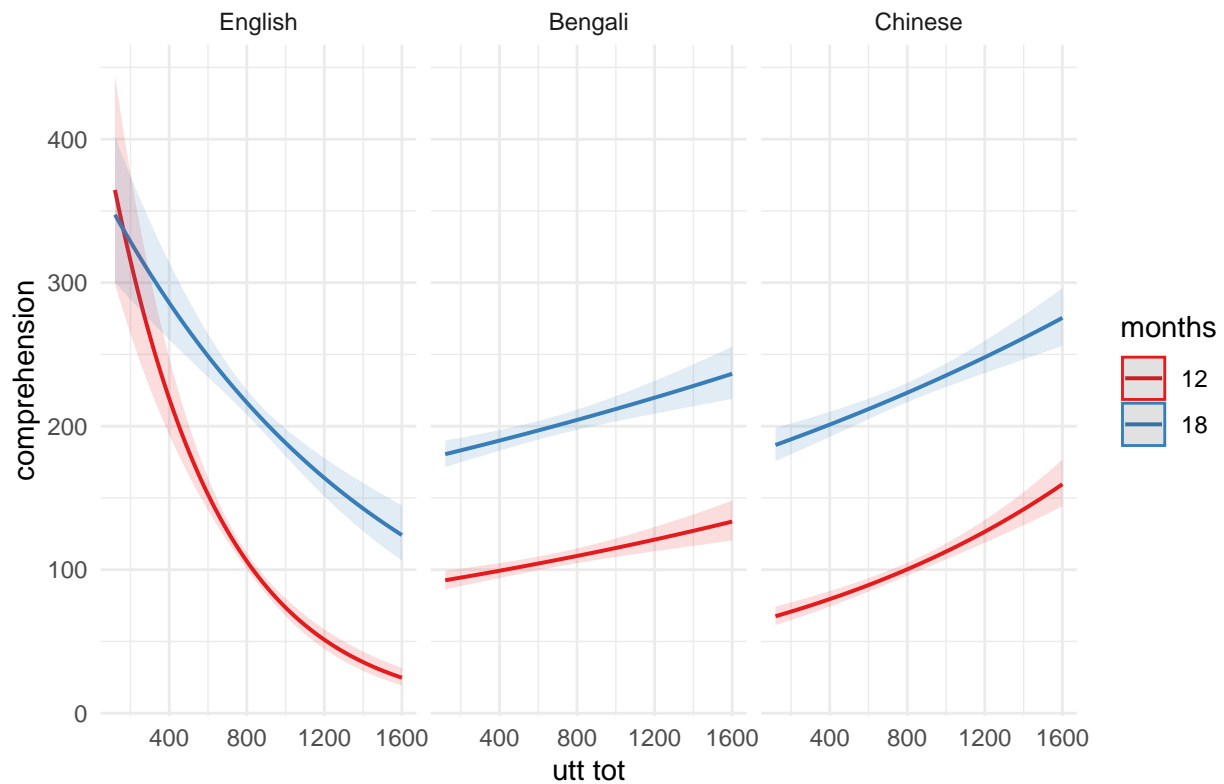
```

```

plot_model(utt_lm, type = "pred", terms = c("utt_tot", "months", "background"))

```

Predicted counts of comprehension



5.1.5 Contingent talks

```
ct_lm <- glm(
  comprehension ~
    ct_tot *
    months *
    background,
  data = filter(vocab, ct_tot < 30),
  family = poisson()
)
summary(ct_lm)
```

```
##
## Call:
## glm(formula = comprehension ~ ct_tot * months * background, family = poisson(),
##      data = filter(vocab, ct_tot < 30))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4501  -4.6077  -0.2327   3.4079  19.6808
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.637232   0.034058 136.157  < 2e-16
```

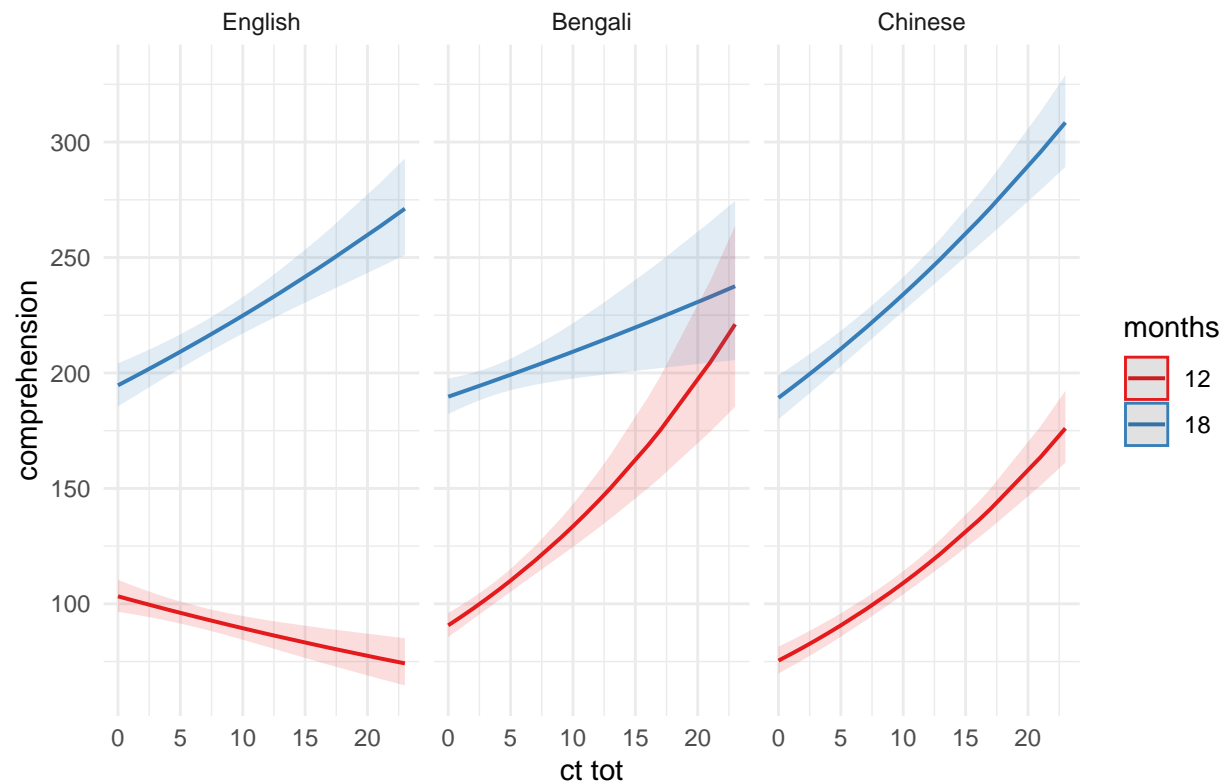
```

## ct_tot -0.014368 0.003855 -3.727 0.000194
## months18 0.633921 0.041849 15.148 < 2e-16
## backgroundBengali -0.130032 0.044814 -2.902 0.003713
## backgroundChinese -0.314289 0.051969 -6.048 1.47e-09
## ct_tot:months18 0.028783 0.004487 6.415 1.41e-10
## ct_tot:backgroundBengali 0.053121 0.005997 8.857 < 2e-16
## ct_tot:backgroundChinese 0.051221 0.004907 10.439 < 2e-16
## months18:backgroundBengali 0.104409 0.054990 1.899 0.057603
## months18:backgroundChinese 0.286084 0.062807 4.555 5.24e-06
## ct_tot:months18:backgroundBengali -0.057756 0.007397 -7.808 5.80e-15
## ct_tot:months18:backgroundChinese -0.044388 0.005814 -7.635 2.26e-14
##
## (Intercept) ***
## ct_tot ***
## months18 ***
## backgroundBengali **
## backgroundChinese ***
## ct_tot:months18 ***
## ct_tot:backgroundBengali ***
## ct_tot:backgroundChinese ***
## months18:backgroundBengali .
## months18:backgroundChinese ***
## ct_tot:months18:backgroundBengali ***
## ct_tot:months18:backgroundChinese ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 6335.5 on 104 degrees of freedom
## Residual deviance: 3804.5 on 93 degrees of freedom
## (1 observation deleted due to missingness)
## AIC: 4526.8
##
## Number of Fisher Scoring iterations: 5

```

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

Predicted counts of comprehension



5.2 Production at 12 and 18 months

5.2.1 All gestures combined

```
all_gest_prod <- glm(
  production ~
    count_tot *
    months *
    background,
  data = vocab,
  family = poisson()
)
summary(all_gest_prod)
```

```
##
## Call:
## glm(formula = production ~ count_tot * months * background, family = poisson(),
##      data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -9.4957  -4.2027  -1.2293   0.9099  28.3218
##
```

```

## Coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.166316   0.098032  22.098 < 2e-16
## count_tot        0.003990   0.001628   2.451  0.0142
## months18         1.601423   0.106336  15.060 < 2e-16
## backgroundBengali 0.106356   0.189875   0.560  0.5754
## backgroundChinese -1.270756   0.241938  -5.252 1.50e-07
## count_tot:months18 0.006925   0.001700   4.074 4.62e-05
## count_tot:backgroundBengali -0.005035   0.004041  -1.246  0.2127
## count_tot:backgroundChinese 0.006937   0.004082   1.699  0.0892
## months18:backgroundBengali -0.064505   0.208611  -0.309  0.7572
## months18:backgroundChinese 0.011565   0.256854   0.045  0.9641
## count_tot:months18:backgroundBengali -0.007497   0.004425  -1.694  0.0902
## count_tot:months18:backgroundChinese 0.008679   0.004237   2.048  0.0405
##
## (Intercept)      ***
## count_tot        *
## months18         ***
## backgroundBengali
## backgroundChinese ***
## count_tot:months18 ***
## count_tot:backgroundBengali
## count_tot:backgroundChinese .
## months18:backgroundBengali
## months18:backgroundChinese
## count_tot:months18:backgroundBengali .
## count_tot:months18:backgroundChinese *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 6616.0  on 108  degrees of freedom
## Residual deviance: 3247.6  on  97  degrees of freedom
## (11 observations deleted due to missingness)
## AIC: 3711.3
##
## Number of Fisher Scoring iterations: 6

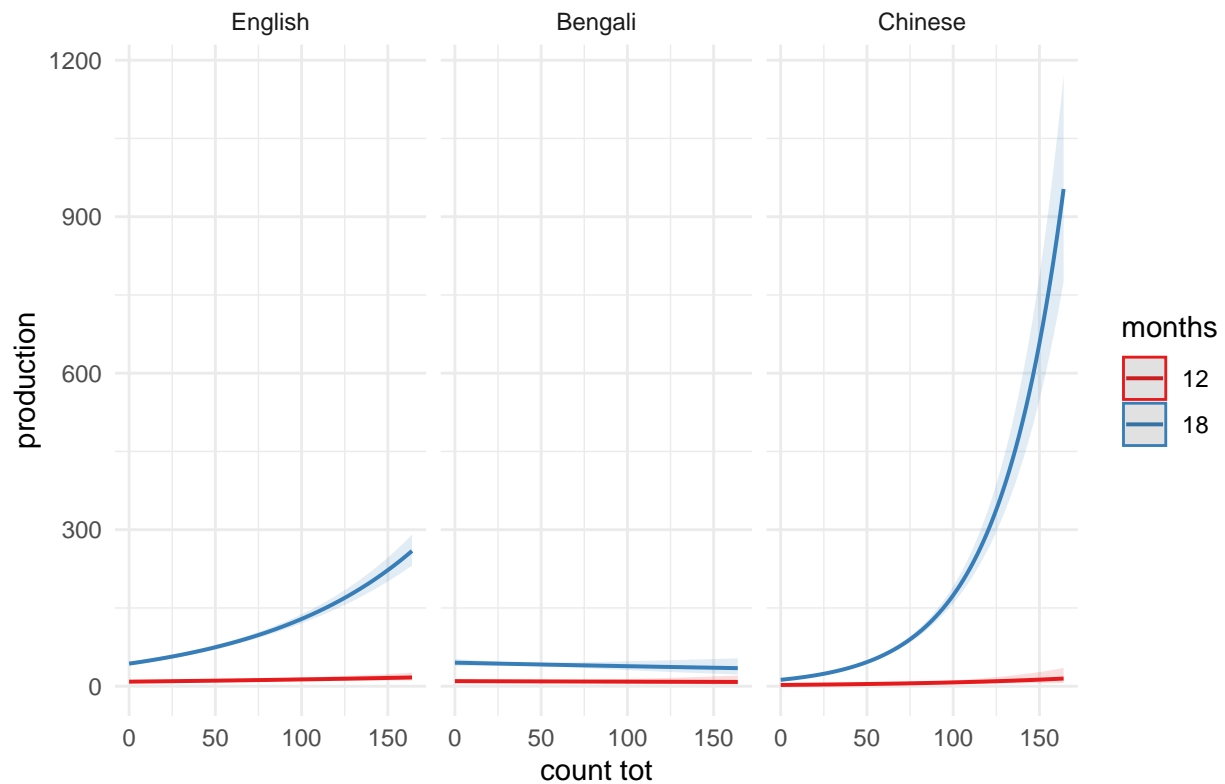
```

```

plot_model(all_gest_prod, type = "pred", terms = c("count_tot", "months", "background"))

```


Predicted counts of production



5.2.2 HoGs + point

```
hgp_prod <- glm(
  production ~
    hgp_tot *
    months *
    background,
  data = vocab,
  poisson()
)
summary(hgp_prod)
```

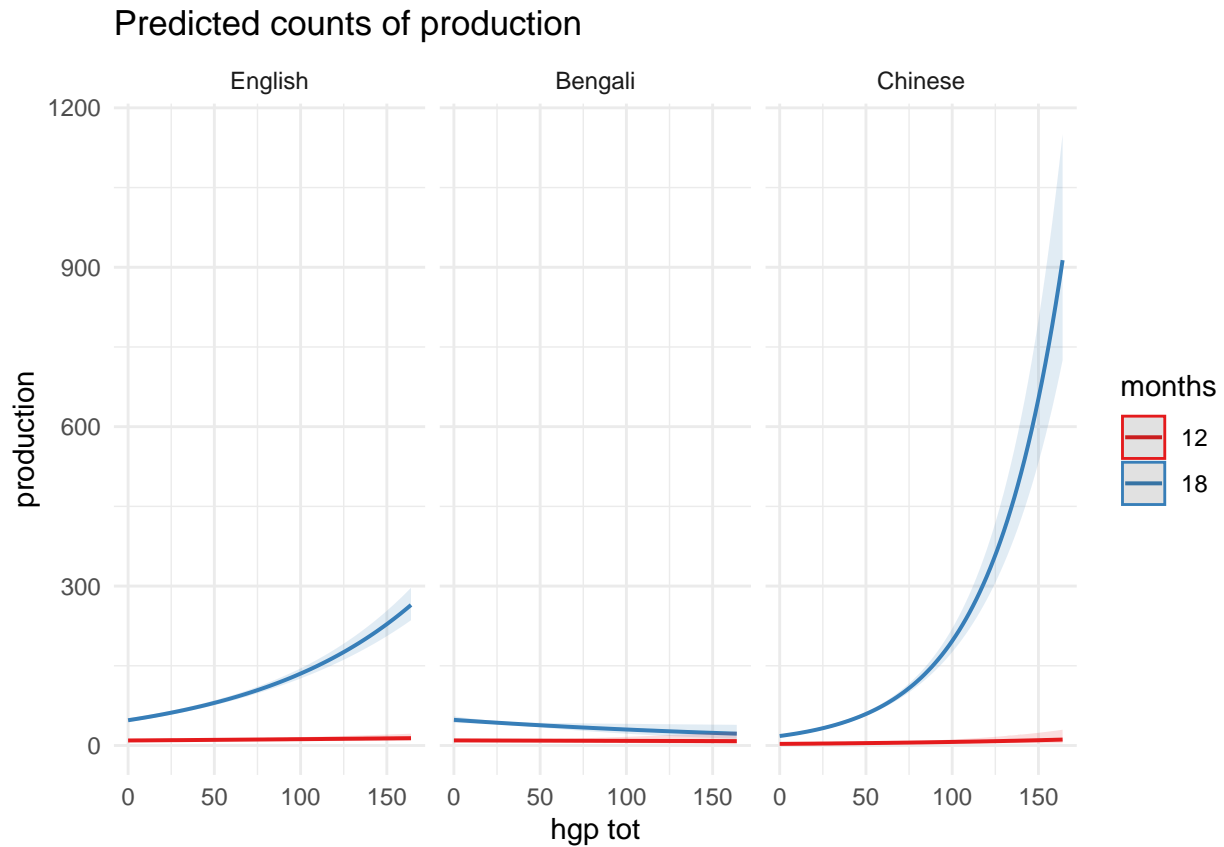
```
##
## Call:
## glm(formula = production ~ hgp_tot * months * background, family = poisson(),
##      data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1715  -4.3463  -1.2156   0.8963  28.5549
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.245524   0.089517  25.085  < 2e-16
```

```

## hgp_tot          0.002311    0.001714    1.348    0.1776
## months18         1.615544    0.097321   16.600 < 2e-16
## backgroundBengali 0.011640    0.167801    0.069    0.9447
## backgroundChinese -1.137042    0.209942   -5.416 6.09e-08
## hgp_tot:months18  0.008156    0.001776    4.592 4.38e-06
## hgp_tot:backgroundBengali -0.003219    0.004646   -0.693    0.4883
## hgp_tot:backgroundChinese 0.005629    0.004230    1.331    0.1833
## months18:backgroundBengali 0.003230    0.184145    0.018    0.9860
## months18:backgroundChinese 0.158991    0.223274    0.712    0.4764
## hgp_tot:months18:backgroundBengali -0.011967    0.005105   -2.344    0.0191
## hgp_tot:months18:backgroundChinese 0.007893    0.004380    1.802    0.0715
##
## (Intercept)          ***
## hgp_tot
## months18             ***
## backgroundBengali
## backgroundChinese     ***
## hgp_tot:months18      ***
## hgp_tot:backgroundBengali
## hgp_tot:backgroundChinese
## months18:backgroundBengali
## months18:backgroundChinese
## hgp_tot:months18:backgroundBengali *
## hgp_tot:months18:backgroundChinese .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 6616.0 on 108 degrees of freedom
## Residual deviance: 3391.2 on 97 degrees of freedom
## (11 observations deleted due to missingness)
## AIC: 3854.9
##
## Number of Fisher Scoring iterations: 6

```

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



5.2.3 Reaches

```
reach_prod <- glm(
  production ~
    reach_tot *
    months *
    background,
  data = vocab,
  family = poisson()
)
summary(reach_prod)
```

```
##
## Call:
## glm(formula = production ~ reach_tot * months * background, family = poisson(),
##      data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -11.152   -4.326   -1.274    1.089   27.720
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.77394    0.12626  14.050 < 2e-16
```

```

## reach_tot                0.06271    0.01005    6.239 4.40e-10
## months18                 2.57343    0.13262   19.405 < 2e-16
## backgroundBengali        0.49872    0.19502    2.557 0.010549
## backgroundChinese       -1.36596    0.30542   -4.472 7.73e-06
## reach_tot:months18      -0.07546    0.01097   -6.877 6.11e-12
## reach_tot:backgroundBengali -0.06637    0.01531   -4.336 1.45e-05
## reach_tot:backgroundChinese  0.03494    0.02410    1.450 0.147099
## months18:backgroundBengali -1.29841    0.21214   -6.121 9.32e-10
## months18:backgroundChinese -0.15125    0.31853   -0.475 0.634915
## reach_tot:months18:backgroundBengali 0.09620    0.01679    5.729 1.01e-08
## reach_tot:months18:backgroundChinese 0.08389    0.02529    3.317 0.000909
##
## (Intercept)              ***
## reach_tot                 ***
## months18                  ***
## backgroundBengali         *
## backgroundChinese         ***
## reach_tot:months18        ***
## reach_tot:backgroundBengali ***
## reach_tot:backgroundChinese
## months18:backgroundBengali ***
## months18:backgroundChinese
## reach_tot:months18:backgroundBengali ***
## reach_tot:months18:backgroundChinese ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 6616.0  on 108  degrees of freedom
## Residual deviance: 3934.9  on  97  degrees of freedom
##    (11 observations deleted due to missingness)
## AIC: 4398.6
##
## Number of Fisher Scoring iterations: 6

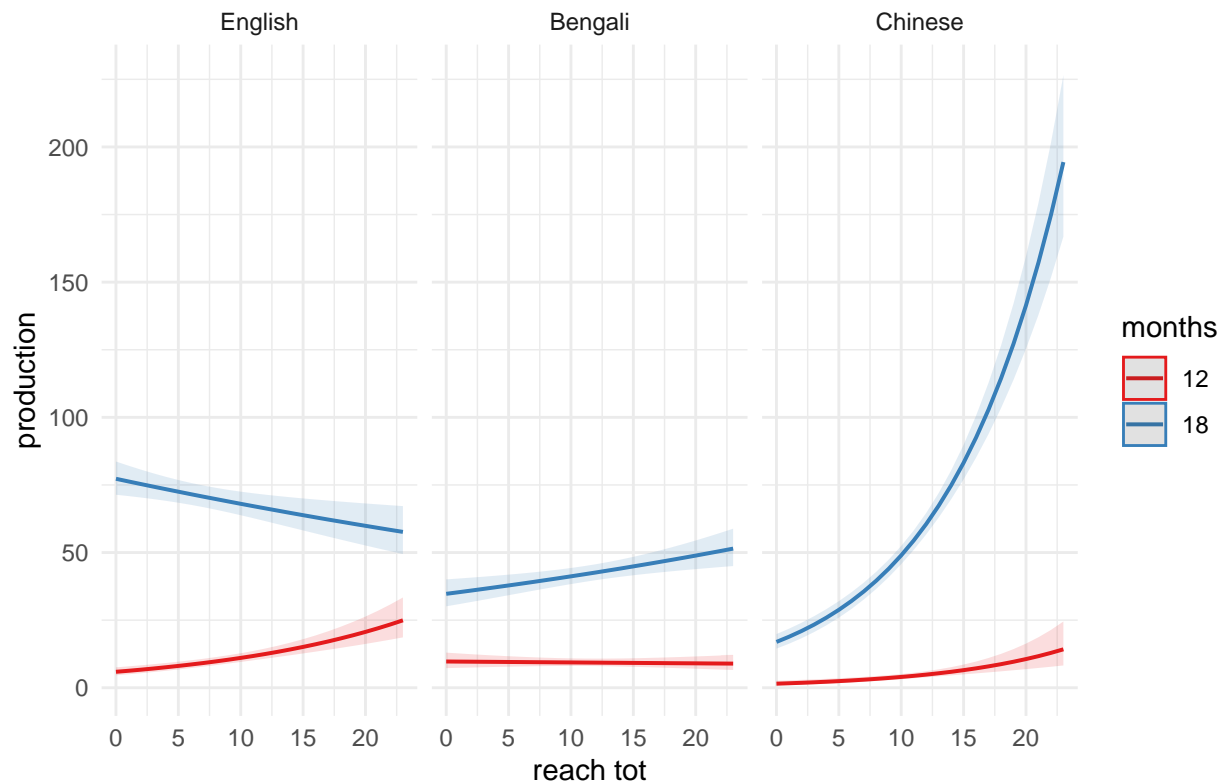
```

```

plot_model(reach_prod, type = "pred", terms = c("reach_tot", "months", "background"))

```

Predicted counts of production



5.2.4 Maternal utterances

```
utt_prod <- glm(
  production ~
    utt_tot *
    months *
    background,
  data = vocab,
  family = poisson()
)
summary(utt_prod)
```

```
##
## Call:
## glm(formula = production ~ utt_tot * months * background, family = poisson(),
##      data = vocab)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -15.241   -3.563   -1.206    1.294   21.551
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.3192300   0.4676905    2.821  0.00479
```

```

## utt_tot          0.0006349  0.0005199   1.221  0.22202
## months18         2.2536527  0.5013805   4.495 6.96e-06
## backgroundBengali 0.2990578  0.4911280   0.609  0.54258
## backgroundChinese -0.3297520  0.5434137  -0.607  0.54397
## utt_tot:months18 -0.0002767  0.0005571  -0.497  0.61938
## utt_tot:backgroundBengali 0.0002044  0.0005441   0.376  0.70713
## utt_tot:backgroundChinese -0.0001612  0.0006045  -0.267  0.78978
## months18:backgroundBengali -1.2230964  0.5287705  -2.313  0.02072
## months18:backgroundChinese 0.4936444  0.5778999   0.854  0.39299
## utt_tot:months18:backgroundBengali 0.0009362  0.0005839   1.603  0.10887
## utt_tot:months18:backgroundChinese -0.0000419  0.0006432  -0.065  0.94806
##
## (Intercept)          **
## utt_tot
## months18             ***
## backgroundBengali
## backgroundChinese
## utt_tot:months18
## utt_tot:backgroundBengali
## utt_tot:backgroundChinese
## months18:backgroundBengali      *
## months18:backgroundChinese
## utt_tot:months18:backgroundBengali
## utt_tot:months18:backgroundChinese
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##   Null deviance: 5308.0  on 98  degrees of freedom
## Residual deviance: 3069.1  on 87  degrees of freedom
##   (21 observations deleted due to missingness)
## AIC: 3480.5
##
## Number of Fisher Scoring iterations: 6

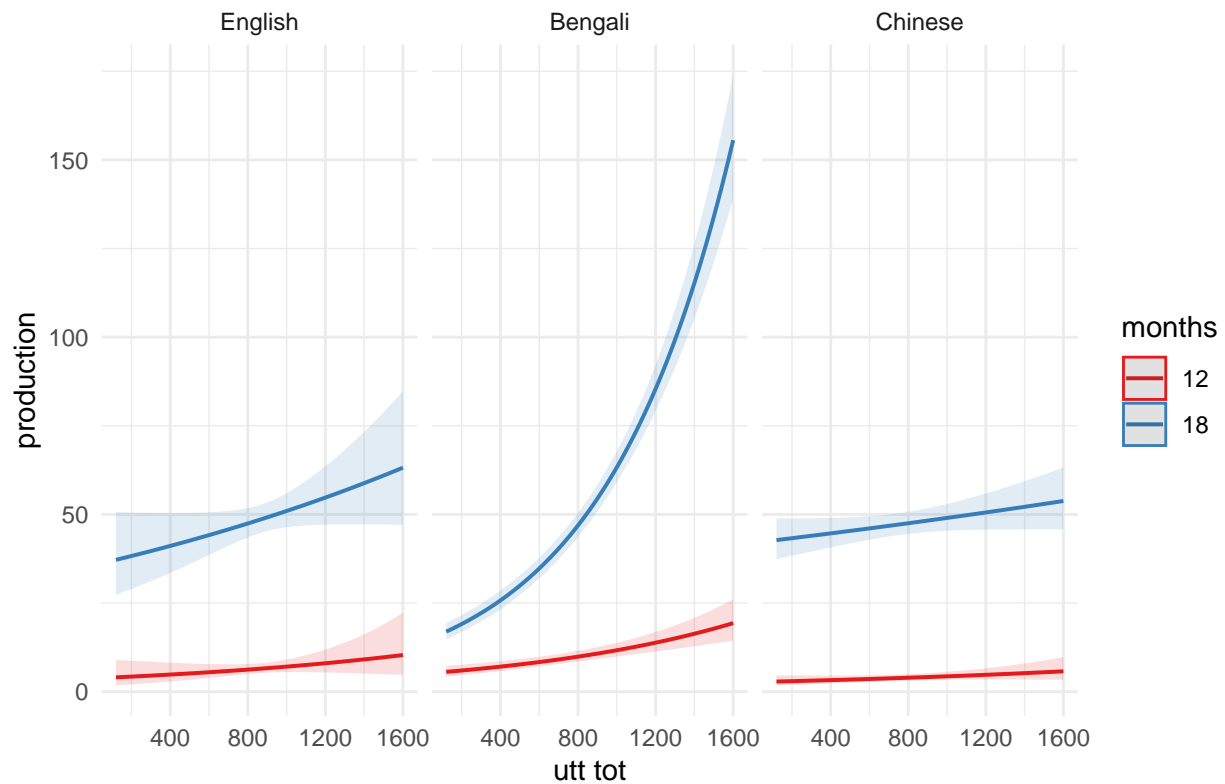
```

```

plot_model(utt_prod, type = "pred", terms = c("utt_tot", "months", "background"))

```

Predicted counts of production



5.2.5 Contingent talks

```
ct_prod <- glm(
  production ~
    ct_tot *
    months *
    background,
  data = filter(vocab, ct_tot < 30),
  family = poisson()
)
summary(ct_prod)
```

```
##
## Call:
## glm(formula = production ~ ct_tot * months * background, family = poisson(),
##      data = filter(vocab, ct_tot < 30))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -10.399   -3.871   -1.082    1.232   19.069
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.964853   0.116021  16.935  < 2e-16
```

```

## ct_tot          0.045305    0.009572    4.733 2.21e-06
## months18        1.530771    0.127526   12.004 < 2e-16
## backgroundBengali -0.140679    0.157079   -0.896 0.37047
## backgroundChinese -1.389104    0.264167   -5.258 1.45e-07
## ct_tot:months18    0.026242    0.010293    2.550 0.01078
## ct_tot:backgroundBengali 0.049489    0.016404    3.017 0.00255
## ct_tot:backgroundChinese 0.026008    0.018783    1.385 0.16617
## months18:backgroundBengali -0.487944    0.175556   -2.779 0.00545
## months18:backgroundChinese 1.348792    0.275822    4.890 1.01e-06
## ct_tot:months18:backgroundBengali 0.047711    0.017752    2.688 0.00719
## ct_tot:months18:backgroundChinese -0.048643    0.019651   -2.475 0.01331
##
## (Intercept)          ***
## ct_tot               ***
## months18             ***
## backgroundBengali
## backgroundChinese    ***
## ct_tot:months18      *
## ct_tot:backgroundBengali **
## ct_tot:backgroundChinese
## months18:backgroundBengali **
## months18:backgroundChinese ***
## ct_tot:months18:backgroundBengali **
## ct_tot:months18:backgroundChinese *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 6123.3  on 104  degrees of freedom
## Residual deviance: 2765.6  on  93  degrees of freedom
##    (1 observation deleted due to missingness)
## AIC: 3209.7
##
## Number of Fisher Scoring iterations: 6

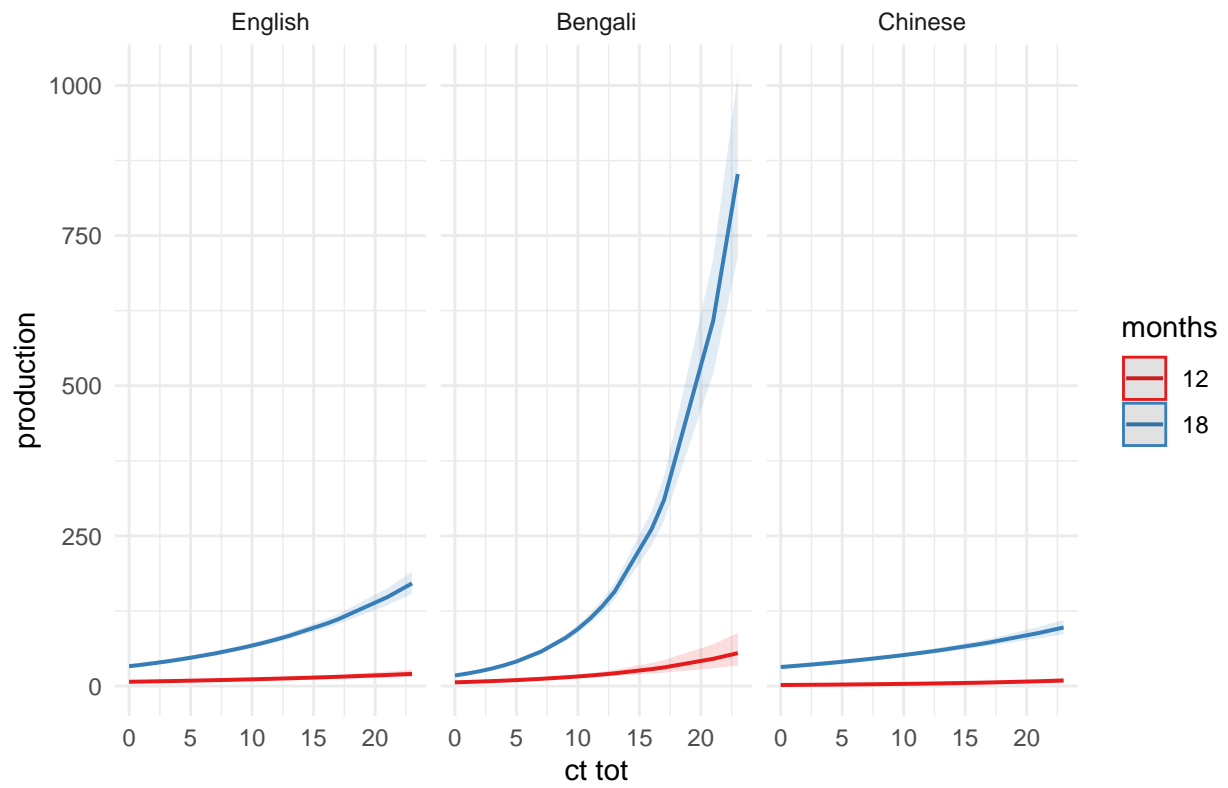
```

```

plot_model(ct_prod, type = "pred", terms = c("ct_tot", "months", "background"))

```


Predicted counts of production



6 R session

```
sessionInfo()
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.5
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] sjPlot_2.6.3      simr_1.0.5        effects_4.1-1
## [4] carData_3.0-2     lmerTest_3.1-0    lme4_1.1-21
## [7] Matrix_1.2-17     tidymv_2.2.0      itsadug_2.3
## [10] plotfunctions_1.3 mgcv_1.8-28       nlme_3.1-140
## [13] forcats_0.4.0     stringr_1.4.0     dplyr_0.8.2
## [16] purrr_0.3.2       readr_1.3.1       tidyr_0.8.3
## [19] tibble_2.1.3      ggplot2_3.2.0     tidyverse_1.2.1
## [22] MASS_7.3-51.4
##
## loaded via a namespace (and not attached):
## [1] TH.data_1.0-10    minqa_1.2.4        colorspace_1.4-1
## [4] rio_0.5.16        sjlabelled_1.1.0    snakecase_0.11.0
## [7] estimability_1.3  rstudioapi_0.10     glmmTMB_0.2.3
## [10] mvtnorm_1.0-11    lubridate_1.7.4     xml2_1.2.0
## [13] codetools_0.2-16  splines_3.5.3       mnormt_1.5-5
## [16] knitr_1.23        sjmisc_2.8.1        jsonlite_1.6
## [19] nloptr_1.2.1      ggeffects_0.11.0    pbkrtest_0.4-7
## [22] broom_0.5.2       binom_1.1-1         compiler_3.5.3
## [25] httr_1.4.0        sjstats_0.17.5      emmeans_1.3.5.1
## [28] backports_1.1.4   assertthat_0.2.1    lazyeval_0.2.2
## [31] survey_3.36       cli_1.1.0           htmltools_0.3.6
## [34] tools_3.5.3       coda_0.19-2         gtable_0.3.0
## [37] glue_1.3.1        Rcpp_1.0.1          cellranger_1.1.0
## [40] iterators_1.0.10  psych_1.8.12        insight_0.4.0
## [43] xfun_0.8          openxlsx_4.1.0.1    rvest_0.3.4
## [46] zoo_1.8-6         scales_1.0.0        hms_0.4.2
## [49] parallel_3.5.3    sandwich_2.5-1      RColorBrewer_1.1-2
## [52] TMB_1.7.15        yaml_2.2.0          curl_3.3
## [55] stringi_1.4.3     bayestestR_0.2.2    plotrix_3.7-6
## [58] boot_1.3-22       zip_2.0.3           rlang_0.4.0
## [61] pkgconfig_2.0.2   evaluate_0.14       lattice_0.20-38
## [64] labeling_0.3      tidyselect_0.2.5    plyr_1.8.4
## [67] magrittr_1.5      R6_2.4.0            generics_0.0.2
```

## [70] multcomp_1.4-10	RLRsim_3.1-3	DBI_1.0.0
## [73] pillar_1.4.2	haven_2.1.0	foreign_0.8-71
## [76] withr_2.1.2	survival_2.44-1.1	abind_1.4-5
## [79] nnet_7.3-12	performance_0.2.0	modelr_0.1.4
## [82] crayon_1.3.4	car_3.0-3	rmarkdown_1.13
## [85] grid_3.5.3	readxl_1.3.1	data.table_1.12.2
## [88] digest_0.6.19	xtable_1.8-4	numDeriv_2016.8-1.1
## [91] munsell_0.5.0	mitools_2.4	