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`. `gestures_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")

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

# Needed for GAMs
contrasts(gestures_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 aggreagated gestures count and maternal contingent talks (`all_tot`).

```r
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).

```r
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`.

Cultural background and development (within the 10-12 months sampling period) were tested separately with two series of models for each gesture (HoGs, reaches, pointing) and maternal scores (maternal utterances and maternal contingent talks). To test the significance of background and development we compared a full model including the relevant parameter with one in which the parameter is dropped, using `itsadug::compareML()`.

The full models testing background contain the following terms: a parametric term for background (`back_o`), a reference smooth over sampling period (`s(months)`, 10-12), a difference smooth over sampling period by background (`s(months, by = back_o)`), and a random smooth over sampling period by infant (`s(months, dyad)`, this corresponds to LME random smooths and intercepts). The reference smooth corresponds to the smooth of development in English infants, while the difference smooth models the difference between the smooth of English infants and those of Bengali and Chinese infants.

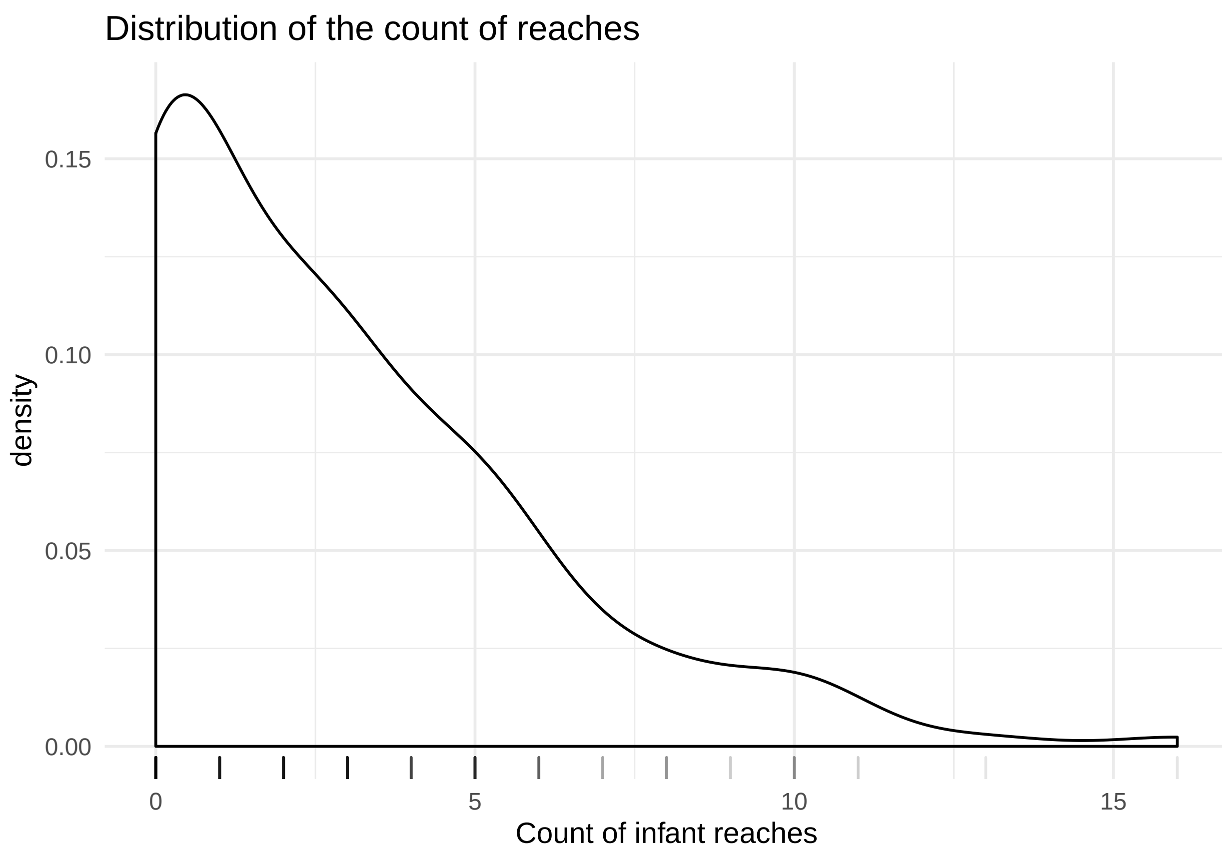
The full models testing development contain the following terms: a smooth over sampling period and a random smooth over sampling period by infant (`s(months, dyad)`, this corresponds to LME random smooths and intercepts).

The null models for background drop all terms including background (`back_o`) while the null models for development drop the smooth over sampling period (`s(months)`), but keep the random smooths (comparison can be done either on the fixed effect structure or the random effects structure at a time).

The warnings about repeated 1-d smooths do not indicate problems with the models, but they only inform the user about multiple smooths over the same variable (which are needed).

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 for infant reaches.

```
# 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 ~
    # parametric term
    back_o +
    # reference smooth
    s(months, k = 3) +
    # difference smoth
    s(months, k = 3, by = back_o) +
    # random smooths (random effect)
    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.
```

```r
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
```

```r
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)
)
```

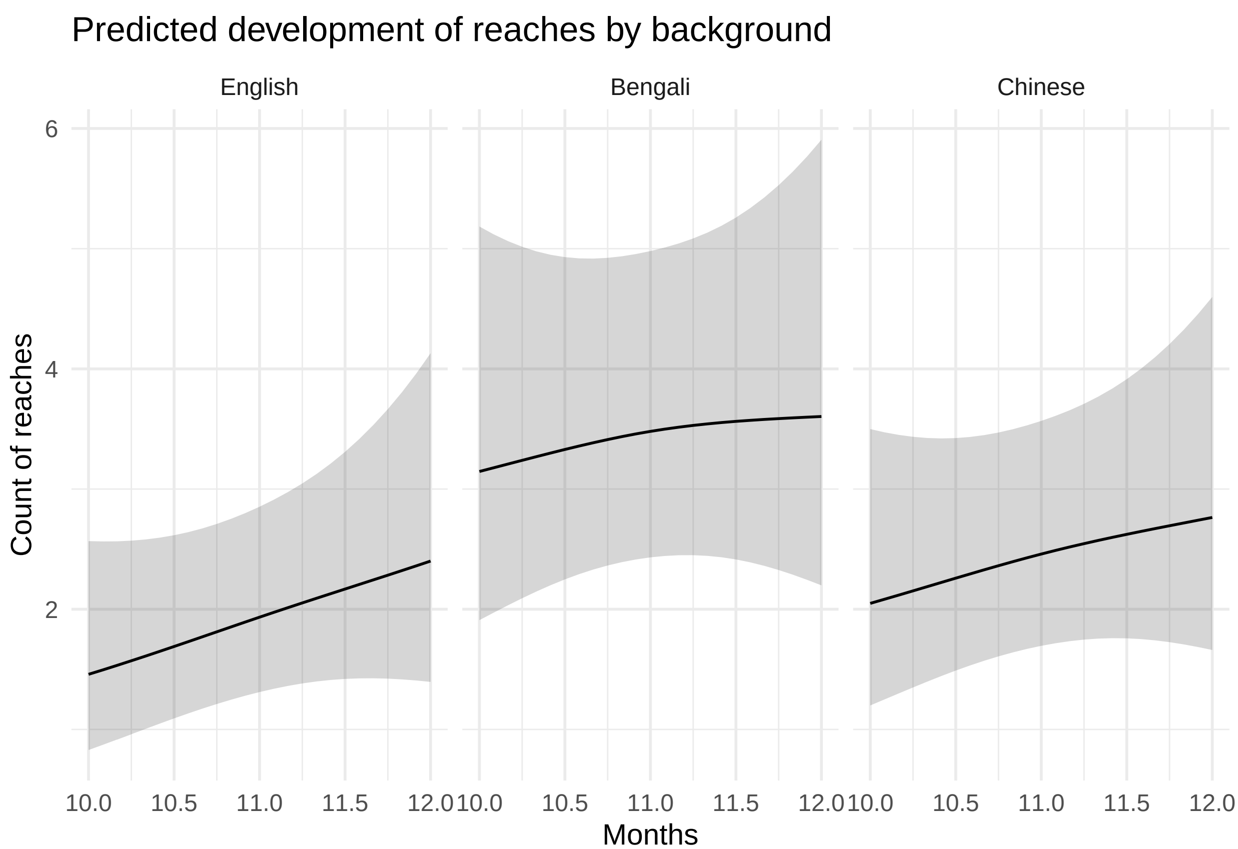
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```r
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) +
  labs(x = "Months", y = "Count of reaches", title = "Predicted development of reaches by background")
```



The following models test the development of infant reaches.

```
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)
)
```

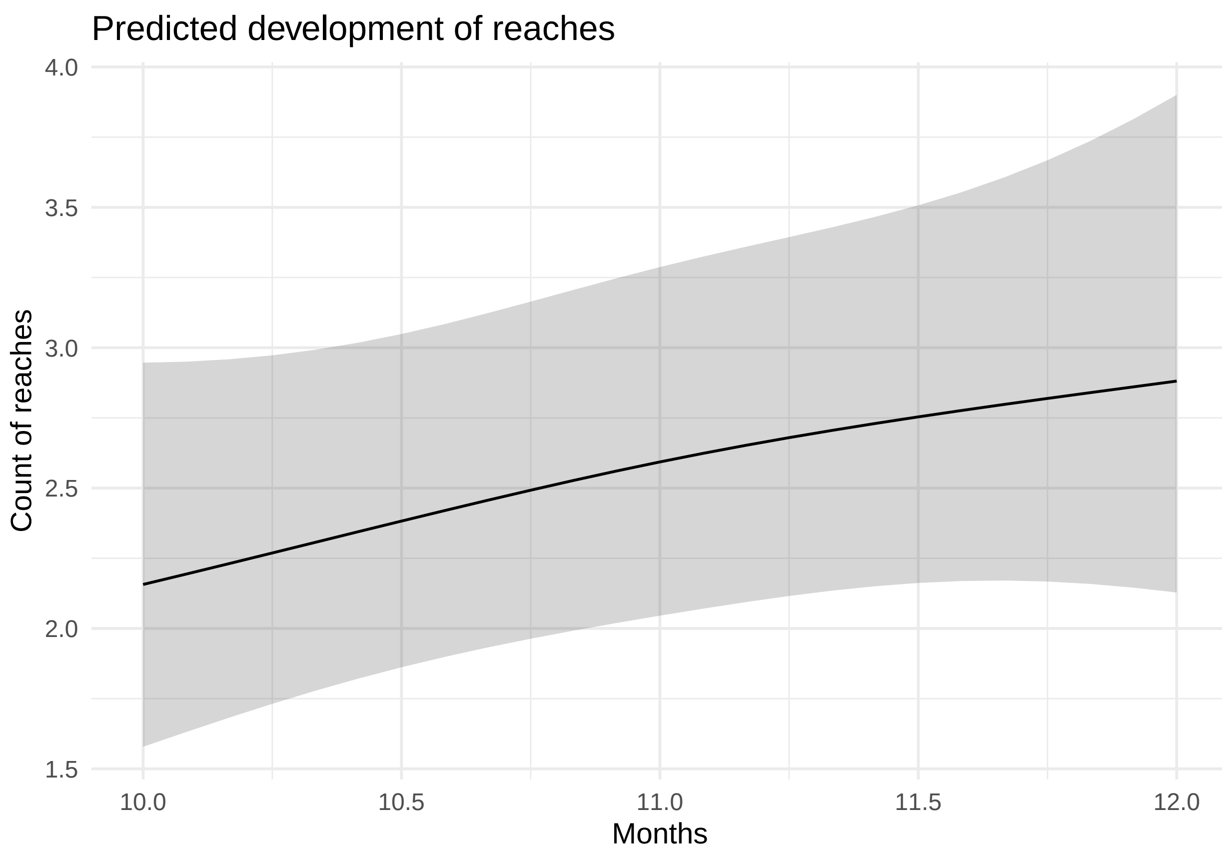
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
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.
```

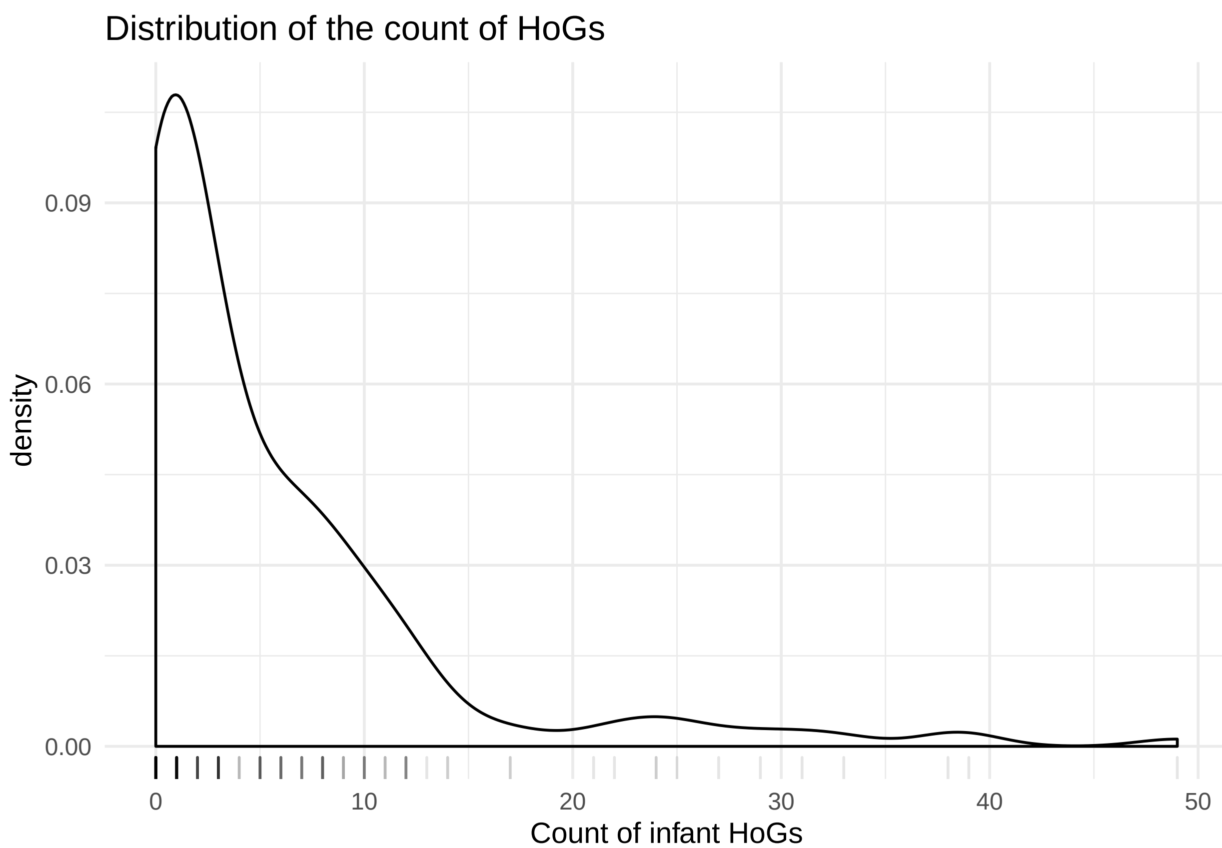
```
## Warning in compareML(reach_gam_2_null, reach_gam_2): Only small difference in ML...
```

```
plot_smooths(reach_gam_2, months, series_length = 25, transform = exp) +
  labs(x = "Months", y = "Count of reaches", title = "Predicted development of reaches")
```



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"
  )
```



The following models test cultural group differences for infant HoGs.

```
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.
```

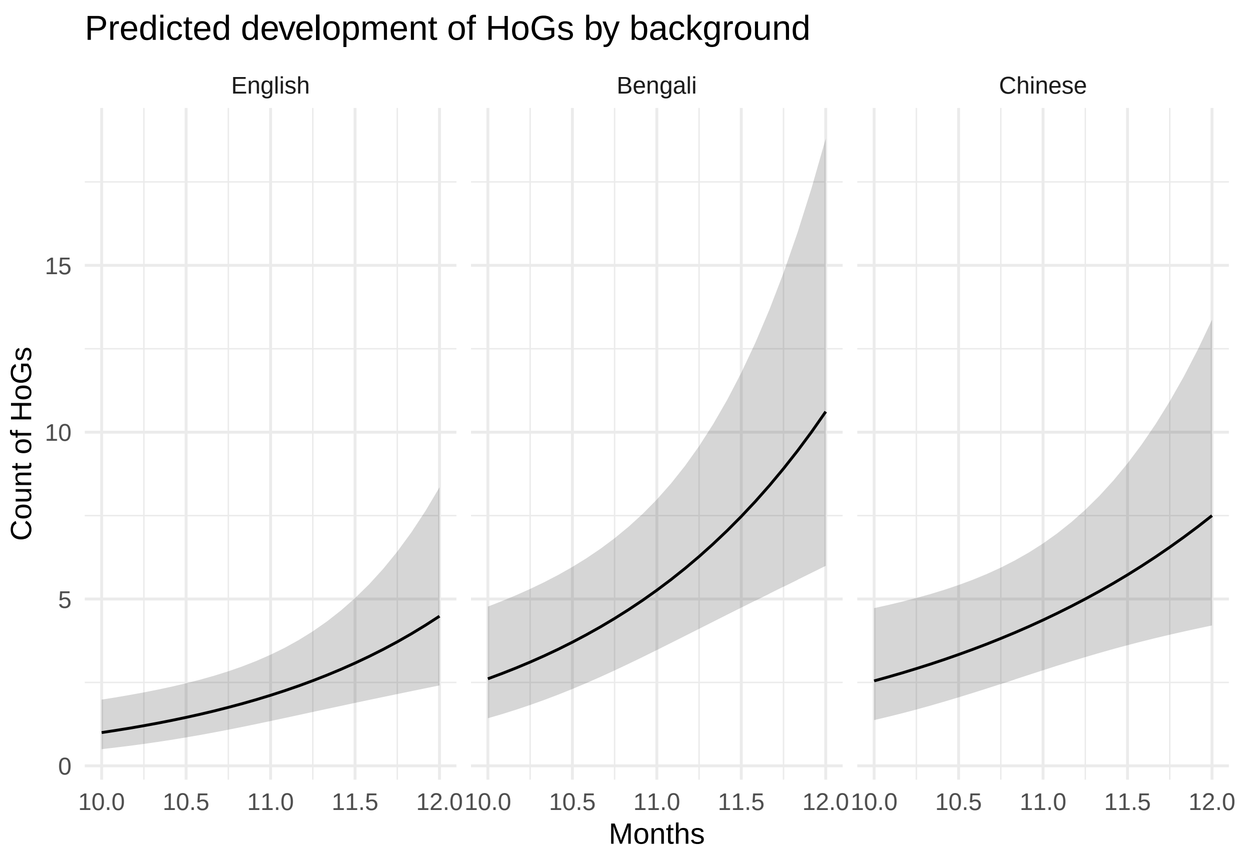
```
compareML(hg_gam_null, hg_gam)
```

```
## 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) +
  labs(x = "Months", y = "Count of HoGs", title = "Predicted development of HoGs by background")
```



The following models test development of infant HoGs.

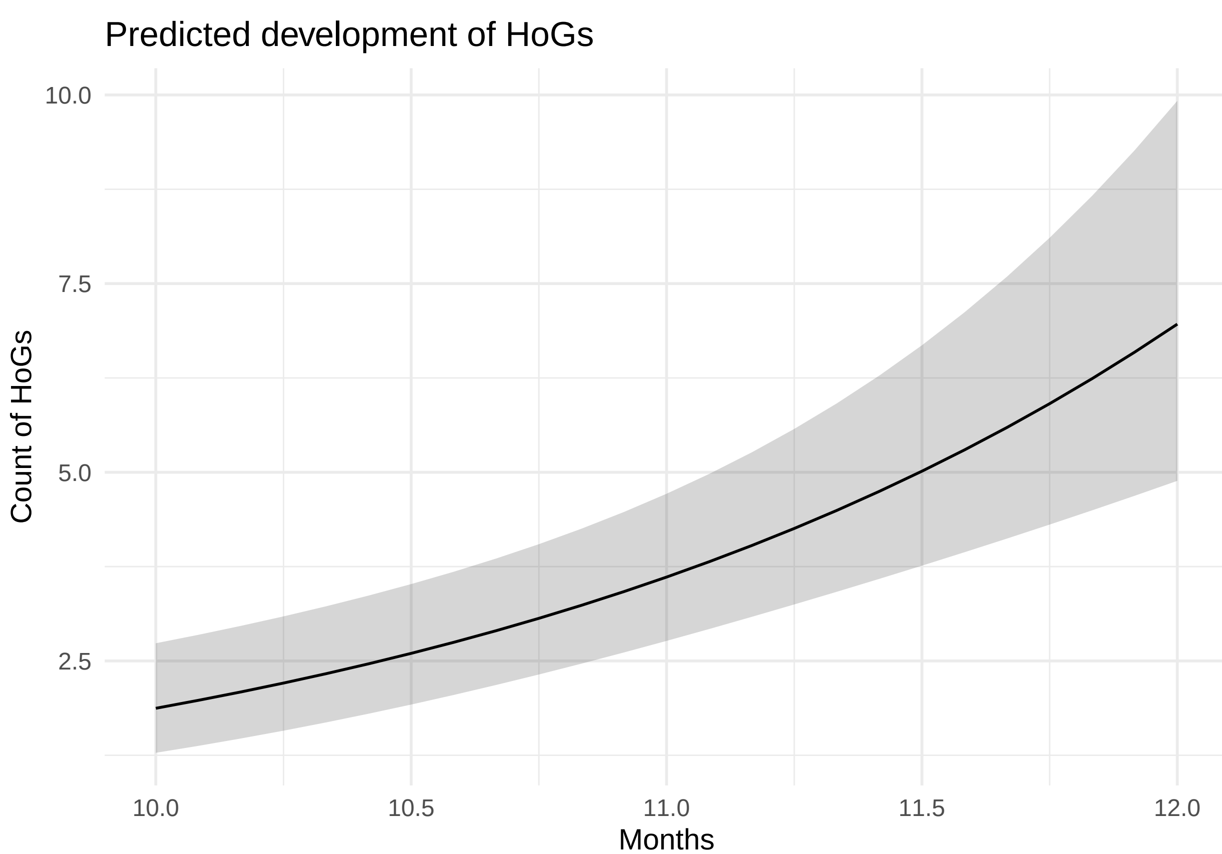
```
hg_gam_2 <- gam(
  count ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = hg_tot,
  method = "ML",
  family = negbin(theta_2)
)
```

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

```r
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)
```

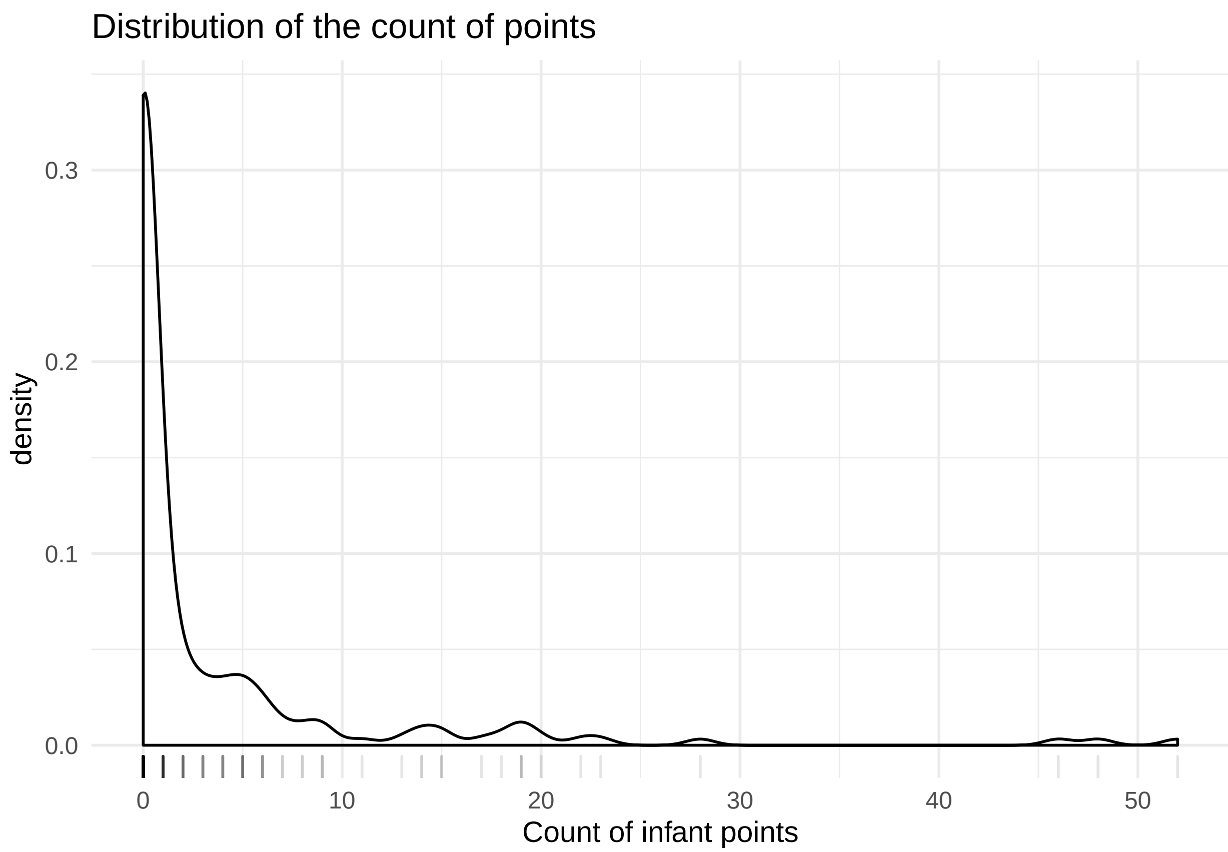
```
## hg_gam_2_null: count ~ s(months, dyad, k = 2, bs = "fs", m = 1)
##
## hg_gam_2: count ~ s(months, k = 3) + s(months, dyad, k = 2, bs = "fs",
##     m = 1)
##
## Chi-square test of ML scores
## -----
##           Model    Score Edf Difference    Df   p.value Sig.
## 1 hg_gam_2_null 467.6971   3
## 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.
```

```r
plot_smooths(hg_gam_2, months, series_length = 25, transform = exp) +
  labs(x = "Months", y = "Count of HoGs", title = "Predicted development of HoGs")
```



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 differences in infant pointing.

```
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",
```

```r
  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.
```

```r
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
```

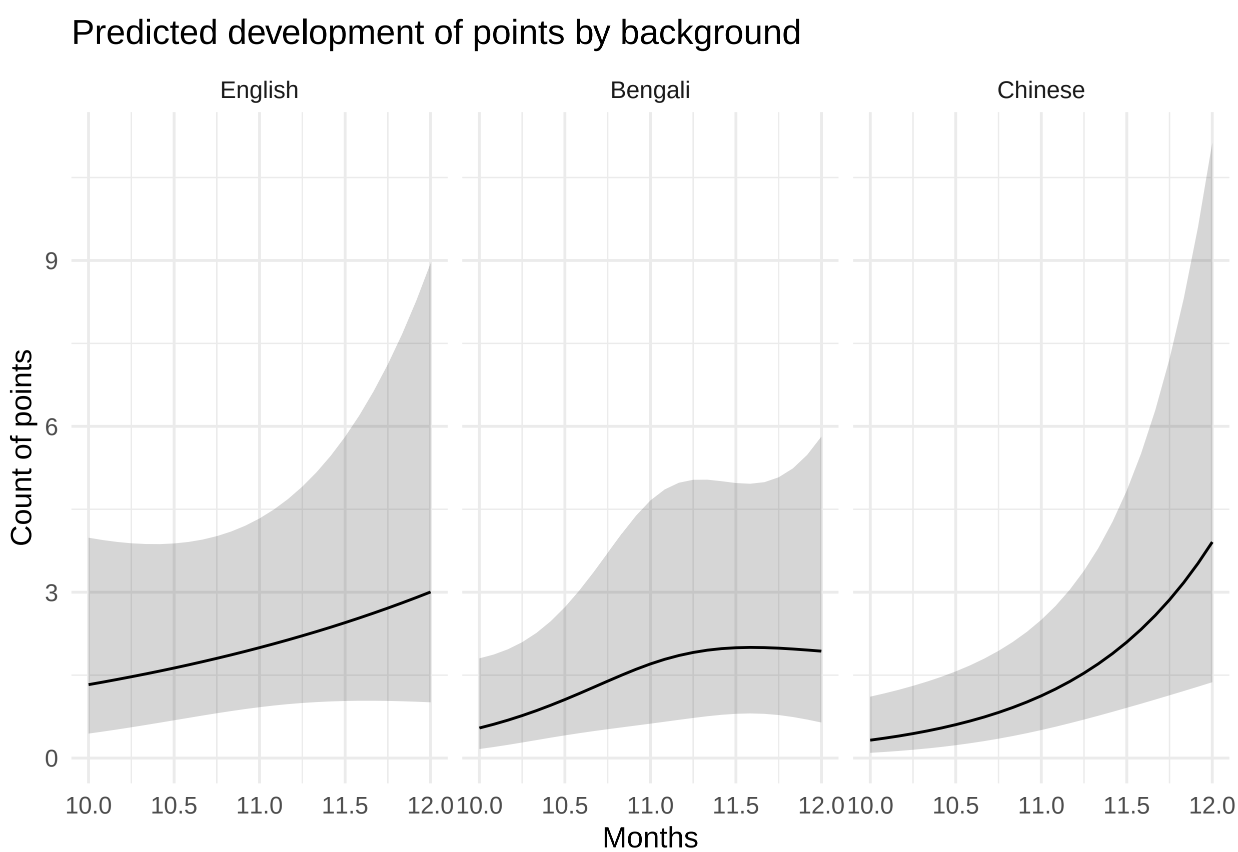
```r
point_gam_null <- gam(
  count ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = point_tot,
  method = "ML",
  family = negbin(theta_3)
)
```

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

```r
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...
```

```r
plot_smooths(point_gam, months, facet_terms = back_o, series_length = 25, transform = exp) +
  labs(x = "Months", y = "Count of points", title = "Predicted development of points by background")
```



The following models test development of infant pointing.

```r
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)
)
```

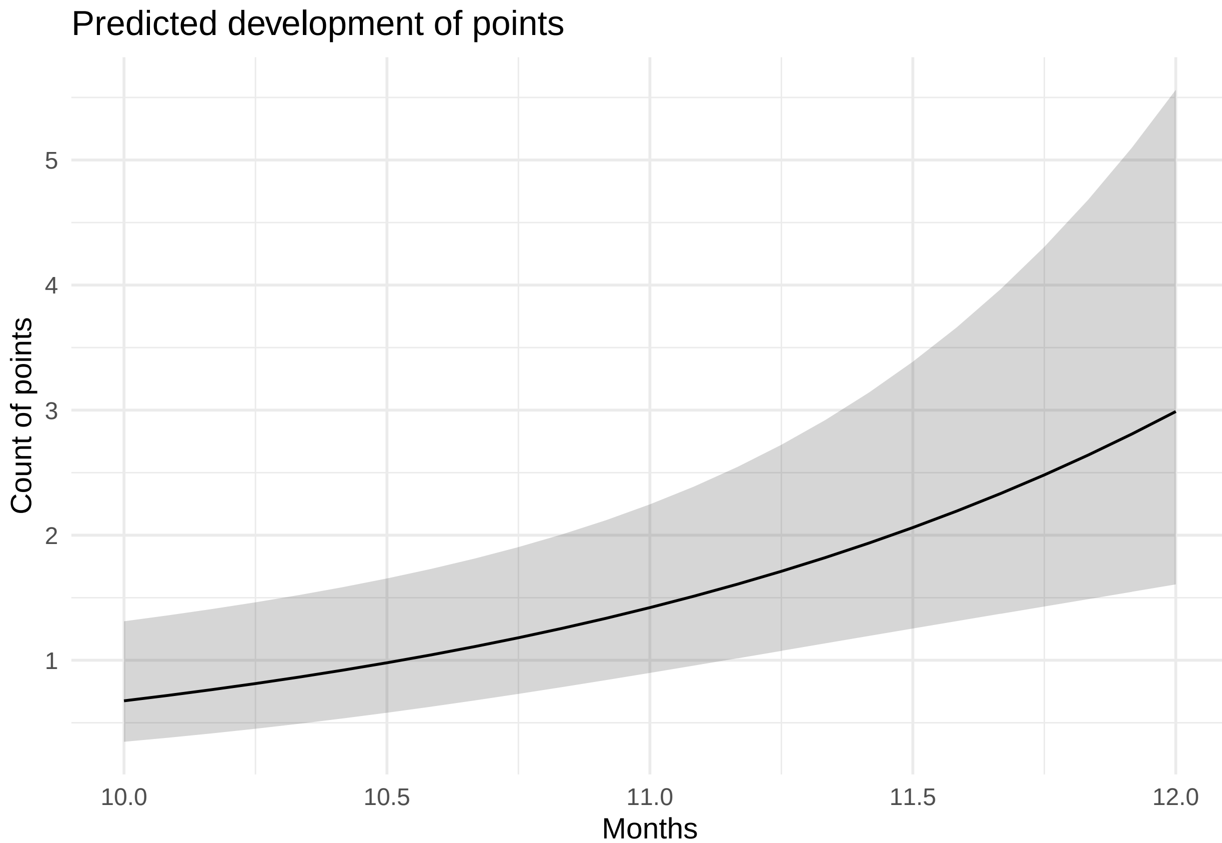
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```r
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...
```

```r
plot_smooths(point_gam_2, months, series_length = 25, transform = exp) +
  labs(x = "Months", y = "Count of points", title = "Predicted development of points")
```



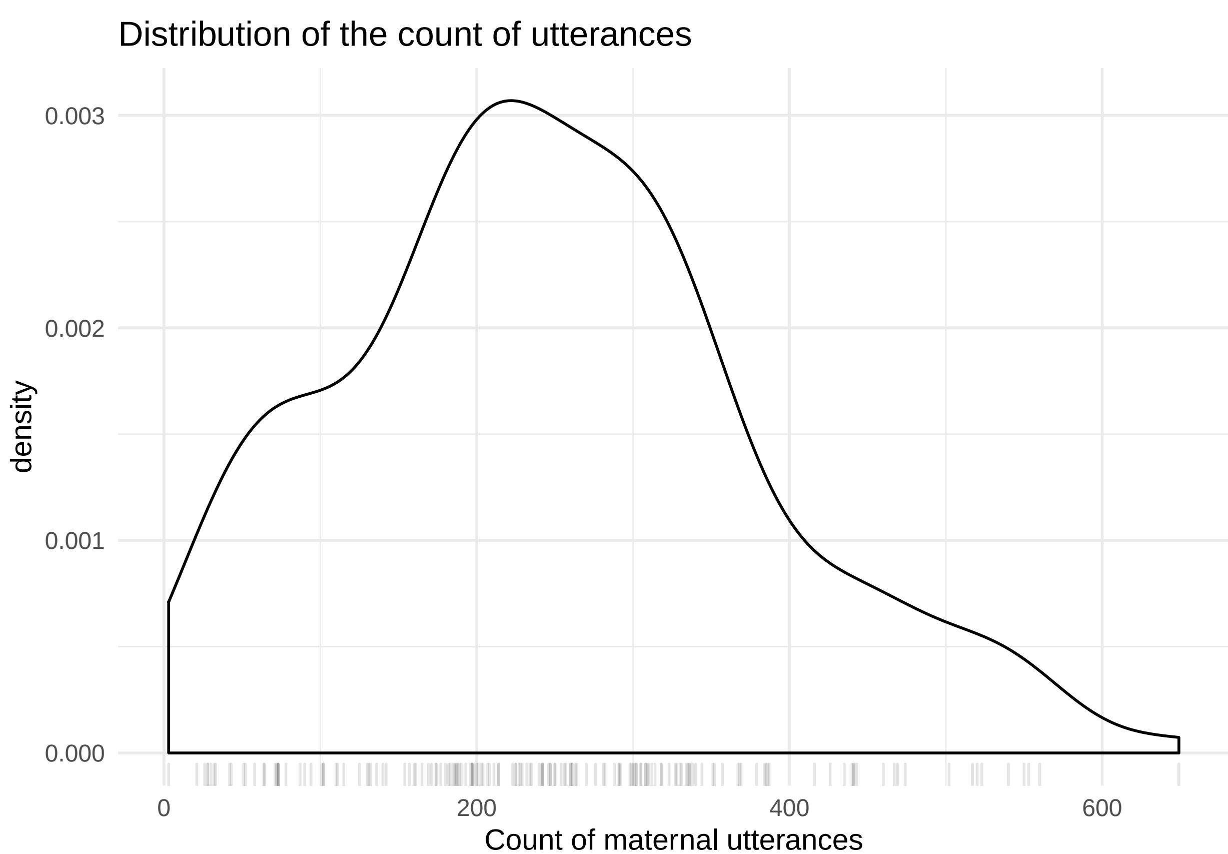
Predicted development of points
Count of points
5
4
3
2
1
10.0
10.5
11.0
11.5
12.0
Months

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 +
```

```r
    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.
```

```r
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
```

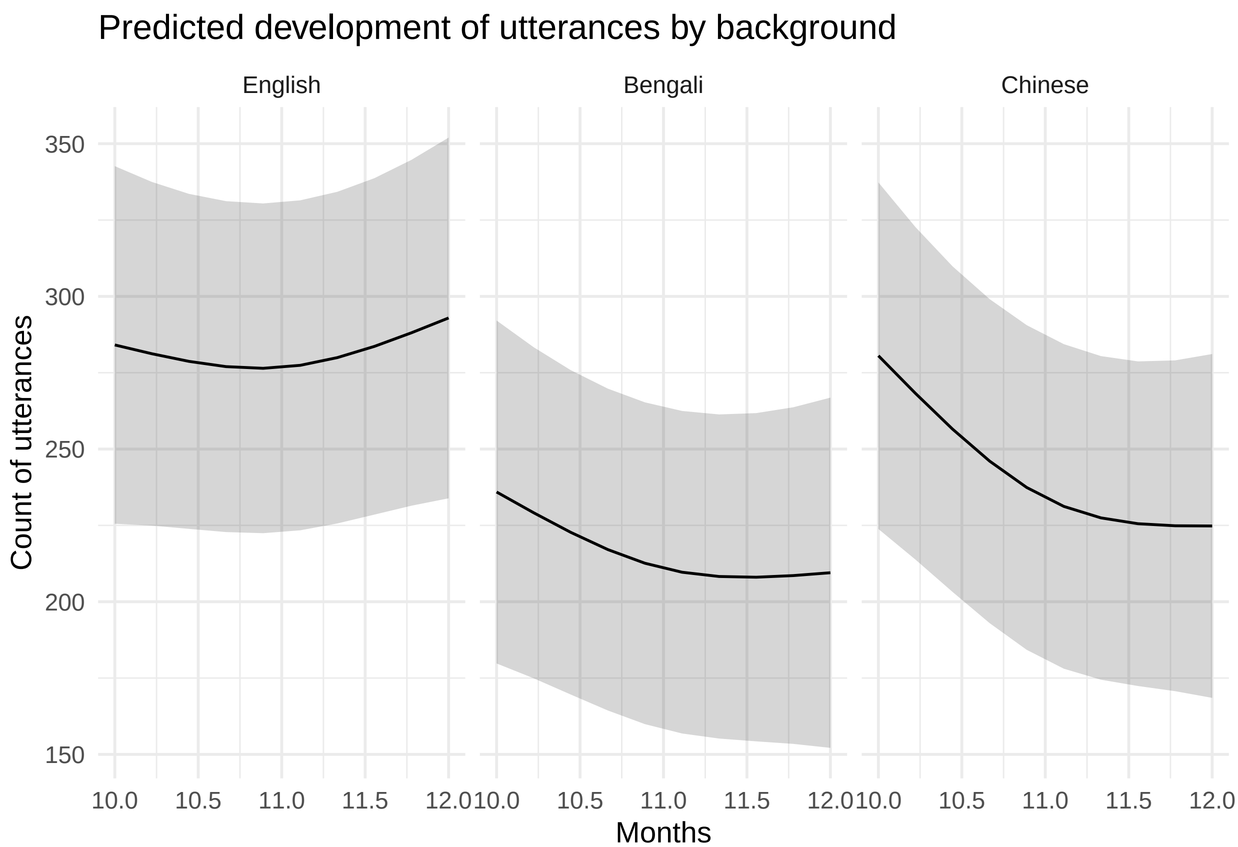
```r
utter_gam_null <- gam(
  utterances ~
    # back_o +
    s(months, k = 3) +
    # s(months, k = 3, by = back_o) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML"
)
```

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

```r
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...
```

```r
plot_smooths(utter_gam, months, facet_terms = back_o, series_length = 10) +
  labs(x = "Months", y = "Count of utterances", title = "Predicted development of utterances by backgrou
```



The following models test time sample.

```r
utter_gam_2 <- gam(
  utterances ~
    s(months, k = 3) +
    s(months, dyad, k = 2, bs = "fs", m = 1),
  data = utterances_tot,
  method = "ML"
)
```

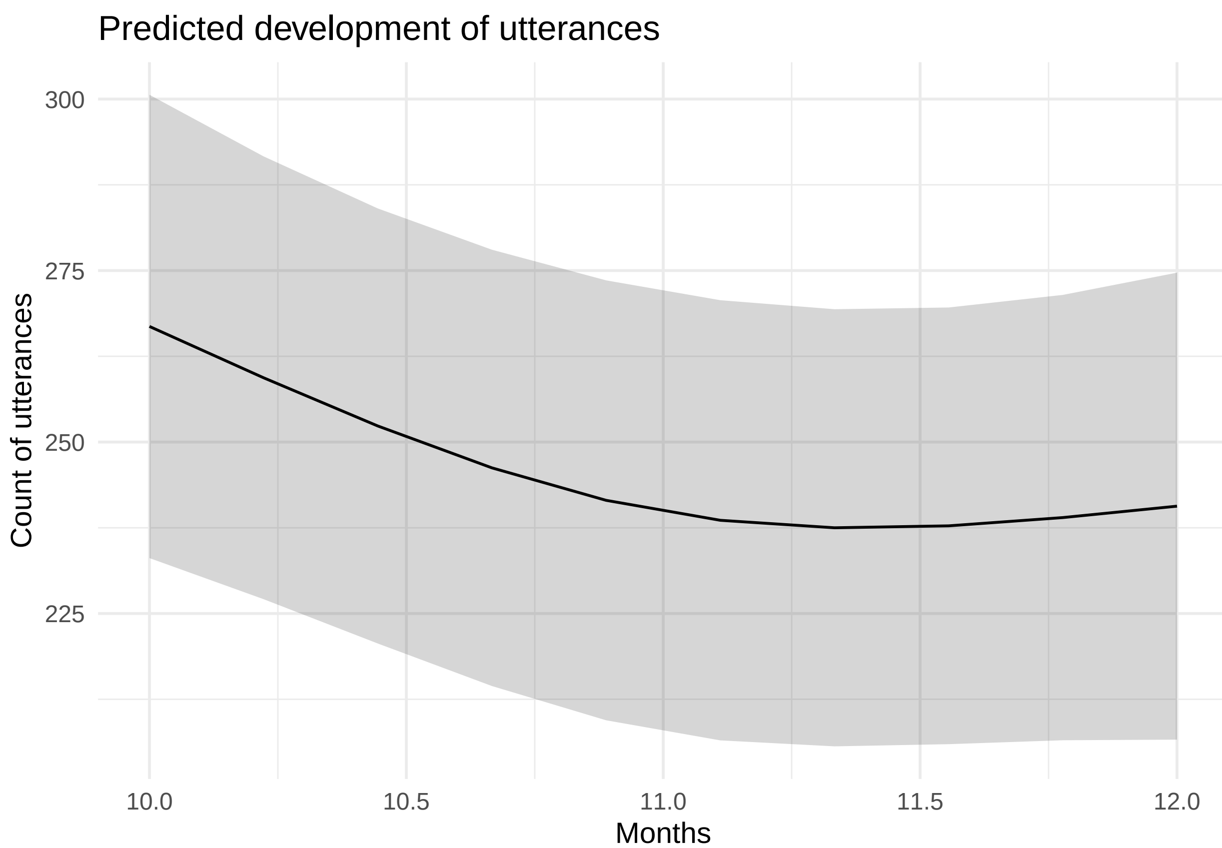
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```r
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.
```

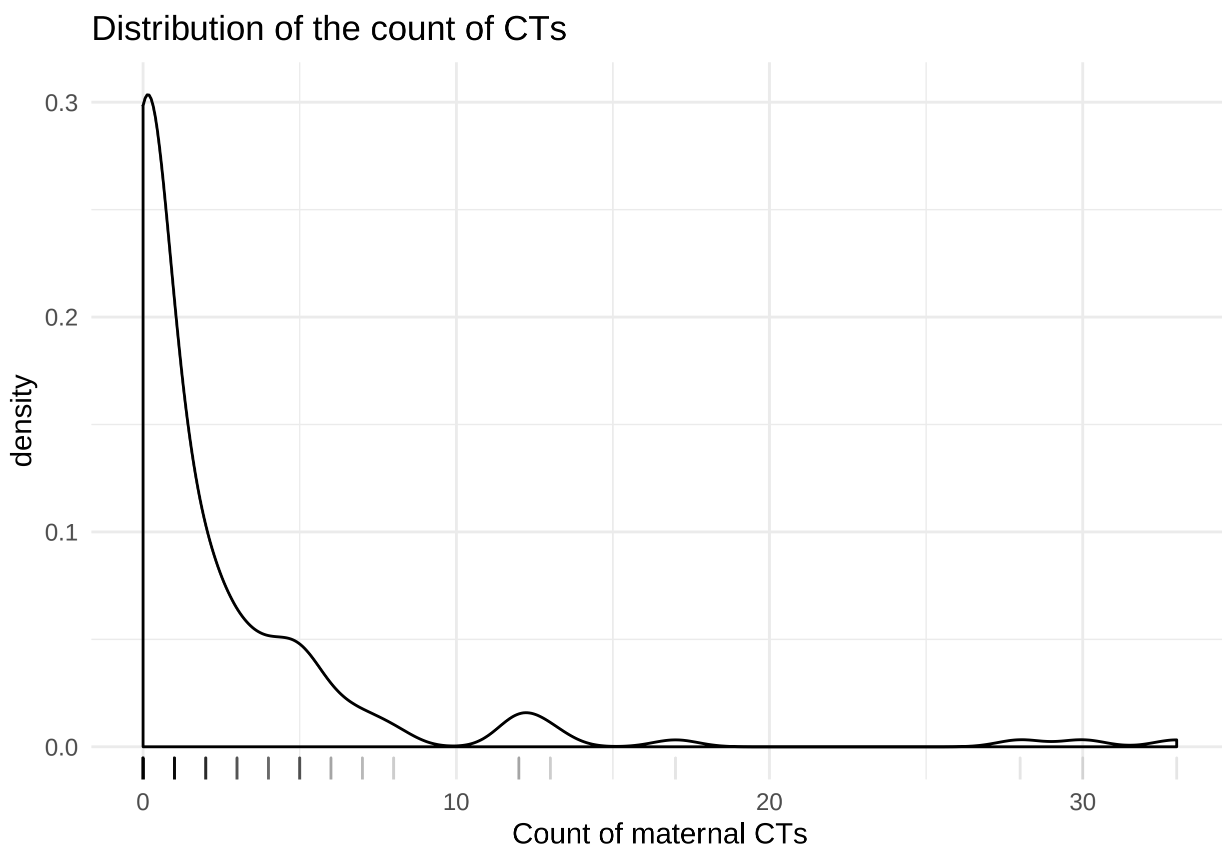
```
## Warning in compareML(utter_gam_2_null, utter_gam_2): Only small difference in ML...
```

```r
plot_smooths(utter_gam_2, months, series_length = 10) +
  labs(x = "Months", y = "Count of utterances", title = "Predicted development of utterances")
```



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"
  )
```



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.
```

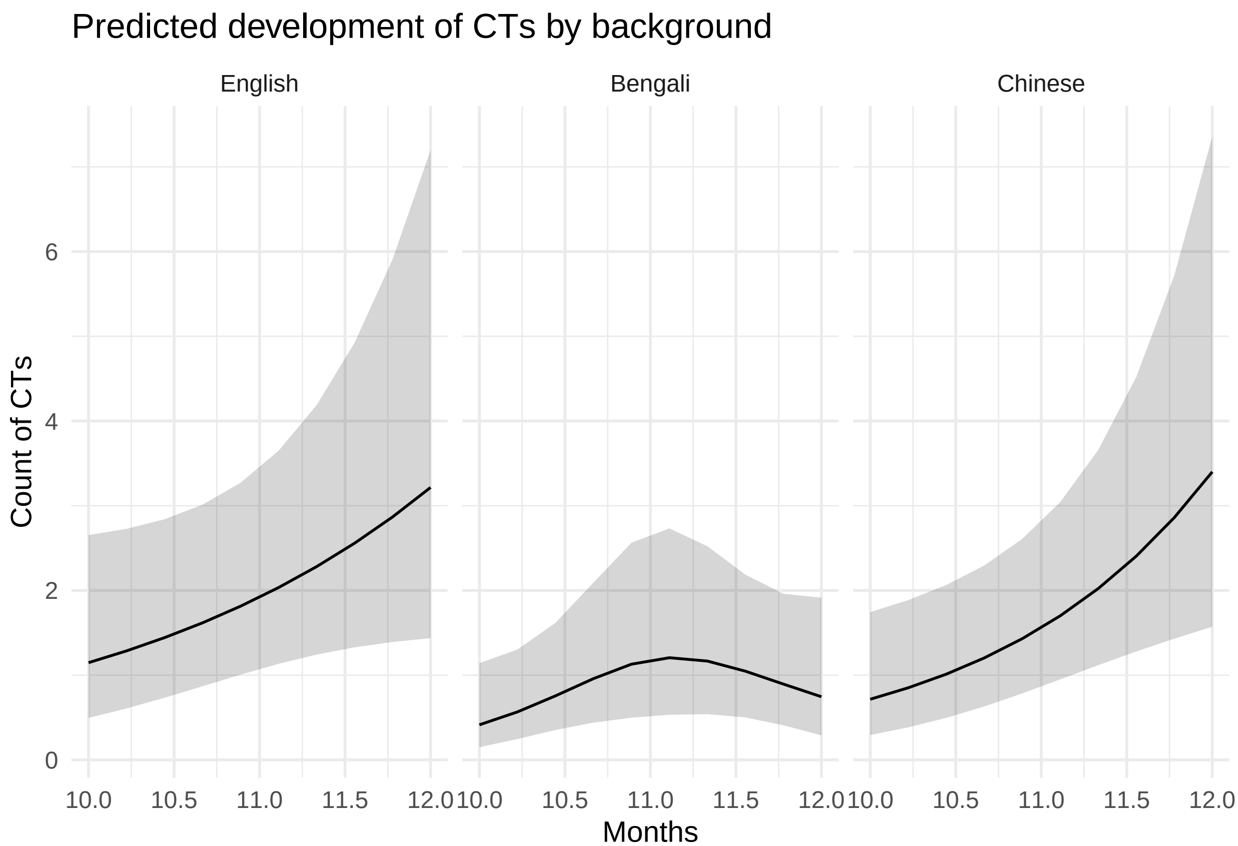
```
compareML(ct_gam_null, ct_gam)
```

```
## 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) +
  labs(x = "Months", y = "Count of CTs", title = "Predicted development of CTs by background")
```



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)
)
```

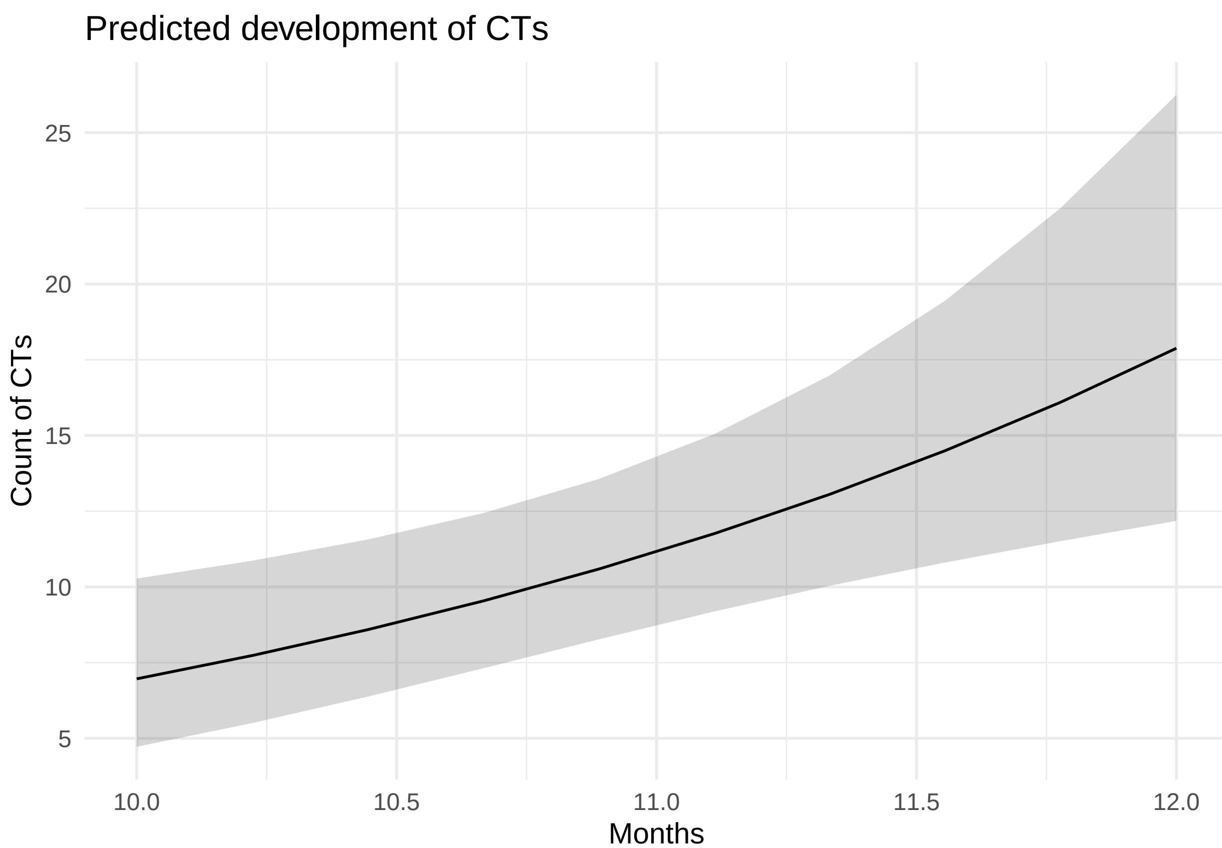
```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```r
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...
```

```r
plot_smooths(ct_gam_2, months, series_length = 10, transform = exp) +
  labs(x = "Months", y = "Count of CTs", title = "Predicted development of CTs")
```


Predicted development of CTs
Count of CTs
25
20
15
10
5
10.0
10.5
11.0
11.5
12.0
Months

4 Analysis 1c. Predictors of pointing

The following GLMMs test the relation between pointing and reaches/HoGs. The count of pointing refers to the one produced by the infant in the subsequent session: For example, the count of reaches at 10 months is matched with the count of points at 11 months, and that of reaches at 11 months is matched with the count of points at 12 months. This allows us to test whether gestures at a certain sampling time predict the production of pointing at the next sampling time. Data on pointing at 10 months is dropped, since there is no data on gestures prior to 10 months.

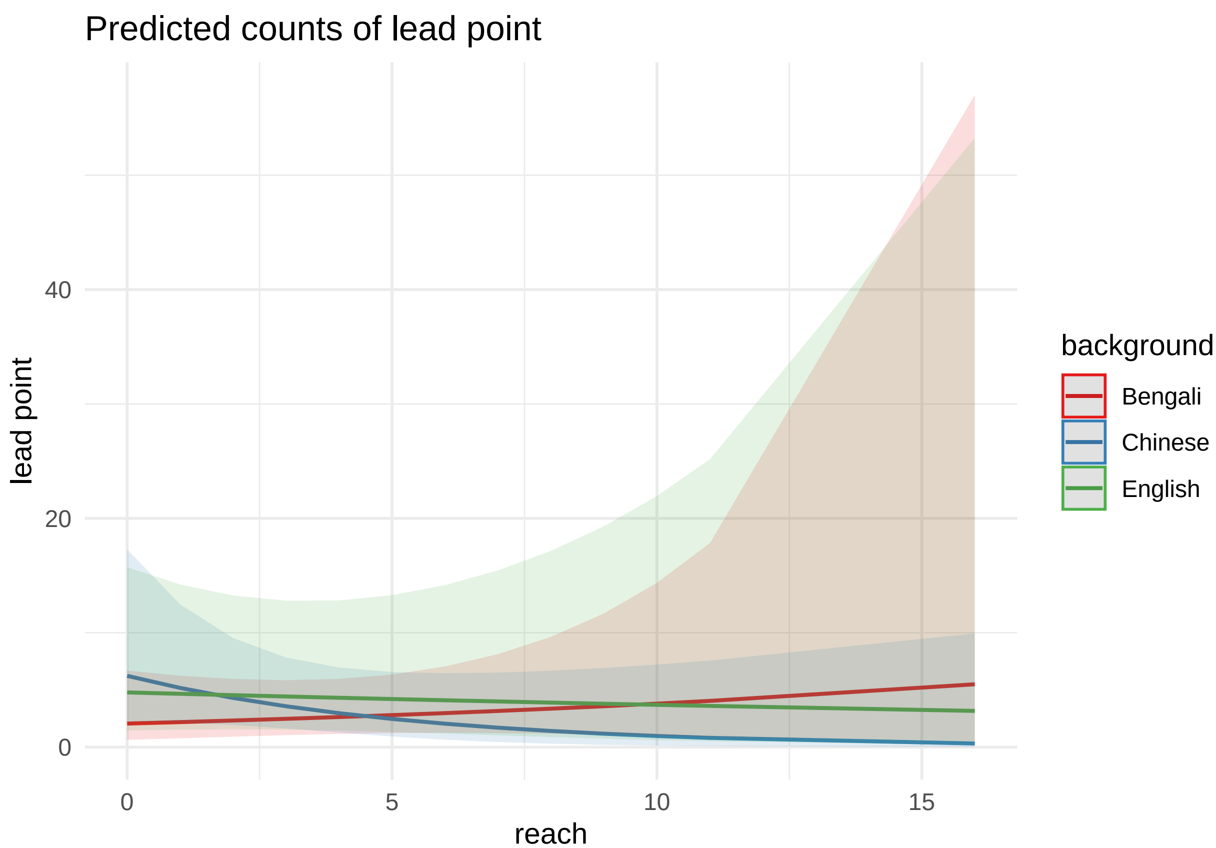
4.1 Reaches

```
reach_point_lead_nb <- glm.nb(lead_point ~ reach, data = reach_point_lead)
theta_5 <- summary(reach_point_lead_nb)[["theta"]]

reach_point_lm <- glmer(
  lead_point ~
    reach *
    background +
    (1|dyad),
  data = reach_point_lead,
  family = negbin(theta_5)
)
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 -0.4983 -0.3934  0.1438  3.0193
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  dyad   (Intercept) 0.157    0.3963
## Number of obs: 112, groups:  dyad, 57
##
## Fixed effects:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.72148    0.60137   1.200    0.230
## reach                    0.06137    0.09716   0.632    0.528
## backgroundChinese        1.10780    0.72839   1.521    0.128
## backgroundEnglish        0.84351    0.68166   1.237    0.216
## reach:backgroundChinese -0.24685    0.16104  -1.533    0.125
## reach:backgroundEnglish -0.08717    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 -0.558  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"))
```



4.2 HoGs

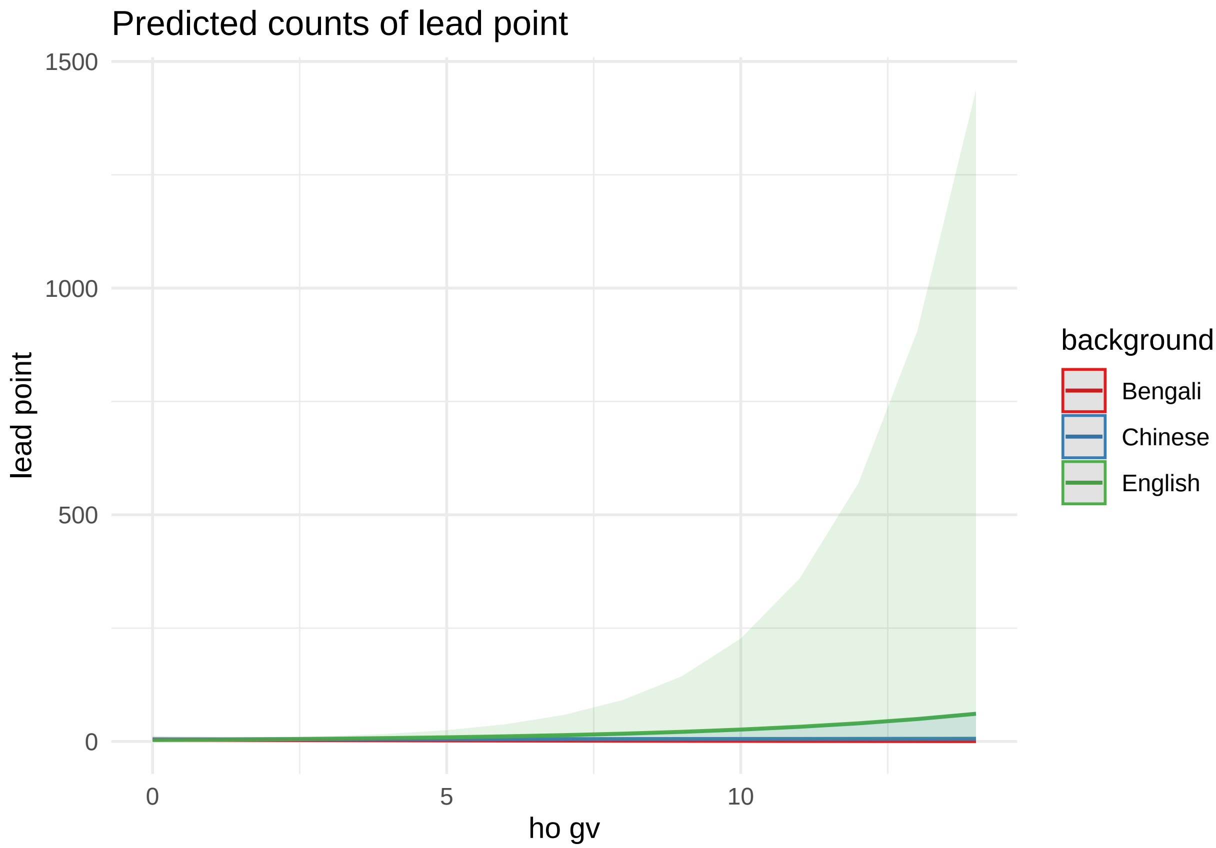
```
hg_point_lead_nb <- glm.nb(lead_point ~ ho_gv, data = filter(hg_point_lead, ho_gv < 20))
theta_6 <- summary(reach_point_lead_nb)[["theta"]]

hg_point_lm <- glmer(
  lead_point ~
    ho_gv *
    background +
    (1|dyad),
  data = filter(hg_point_lead, ho_gv < 20),
  family = negbin(theta_6)
)
```

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

```r
summary(hg_point_lm)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Negative Binomial(0.268)  ( log )
## Formula: lead_point ~ ho_gv * background + (1 | dyad)
##    Data: filter(hg_point_lead, ho_gv < 20)
##
##      AIC      BIC   logLik deviance df.resid
##    503.6    525.1   -243.8    487.6      101
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.5152 -0.5009 -0.4033  0.1257  6.1781
##
## Random effects:
##  Groups Name        Variance  Std.Dev.
##  dyad   (Intercept) 1.407e-10 1.186e-05
## Number of obs: 109, groups:  dyad, 57
##
## Fixed effects:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.37535    0.45787   3.004  0.00267 **
## ho_gv                   -0.10720    0.07934  -1.351  0.17665
## backgroundChinese        0.11398    0.67981   0.168  0.86685
## backgroundEnglish       -0.22597    0.62061  -0.364  0.71577
## ho_gv:backgroundChinese  0.12681    0.13692   0.926  0.35435
## ho_gv:backgroundEnglish  0.31874    0.15354   2.076  0.03790 *
## ---
## 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 -0.674  0.459
## bckgrndEngl -0.738  0.502  0.497
## h_gv:bckgrC  0.395 -0.579 -0.714 -0.291
## h_gv:bckgrE  0.352 -0.517 -0.237 -0.621  0.299
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
plot_model(hg_point_lm, type = "pred", terms = c("ho_gv", "background"))
```



Predicted counts of lead point
1500
1000
500
0
lead point
0
5
10
ho gv
background
Bengali
Chinese
English

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

5.1 Comprehension at 12 and 18 months

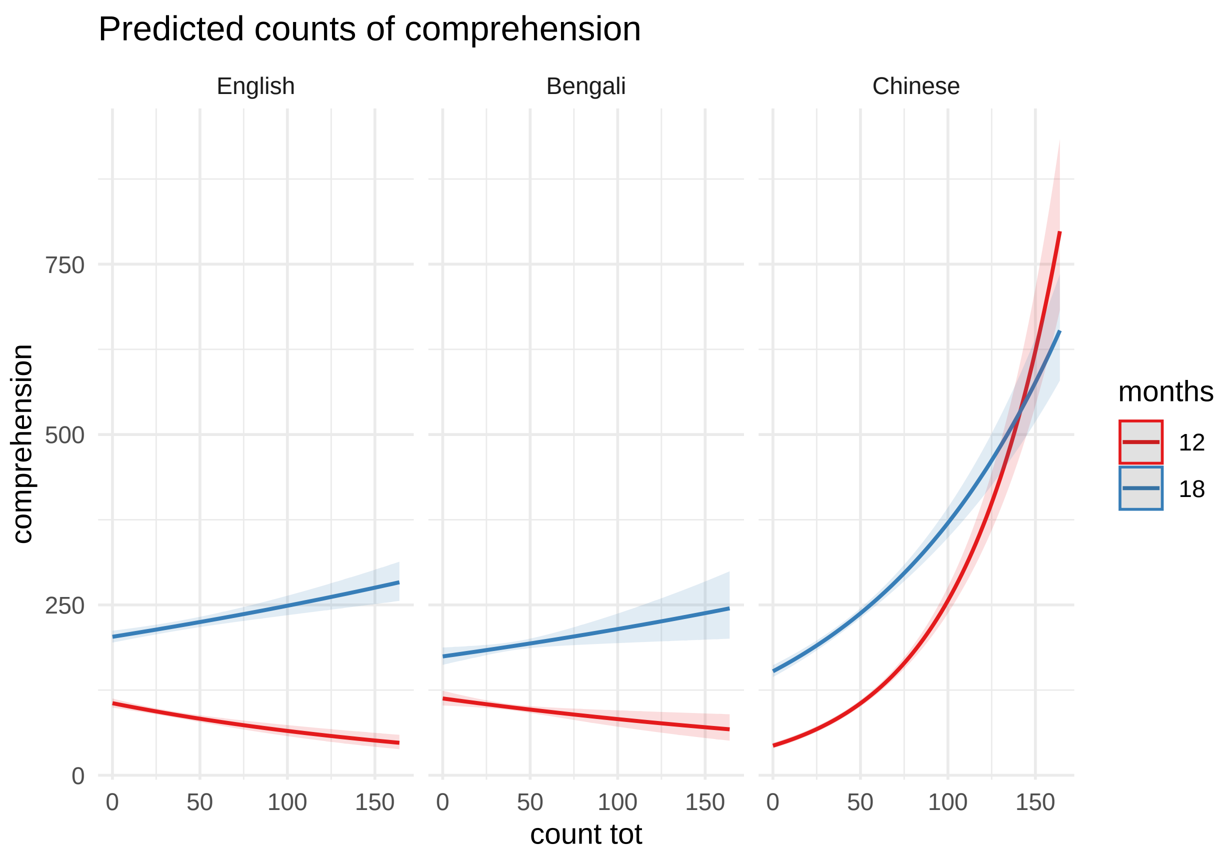
5.1.1 All gestures combined

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

```
##
## Call:
## glm(formula = comprehension ~ count_tot * months * background,
##     data = vocab)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -156.697   -39.955    -6.224    32.438   171.234
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          103.51505   21.68158   4.774 6.38e-06
## count_tot                             -0.36254    0.43271  -0.838  0.40419
## months18                              98.49938   31.30911   3.146  0.00220
## backgroundBengali                      8.58252   41.35601   0.208  0.83603
## backgroundChinese                    -87.92140   36.08175  -2.437  0.01664
## count_tot:months18                     0.84553    0.61592   1.373  0.17298
## count_tot:backgroundBengali            0.05165    0.90402   0.057  0.95456
## count_tot:backgroundChinese            2.51926    0.72934   3.454  0.00082
## months18:backgroundBengali           -36.68396   58.82787  -0.624  0.53437
## months18:backgroundChinese            22.72917   51.41852   0.442  0.65944
## count_tot:months18:backgroundBengali  -0.14125    1.28038  -0.110  0.91239
## count_tot:months18:backgroundChinese  -0.80004    1.03380  -0.774  0.44088
##
## (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 gaussian family taken to be 4962.375)
##
##     Null deviance: 988726  on 108  degrees of freedom
## Residual deviance: 481350  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1250.2
##
## Number of Fisher Scoring iterations: 2
```

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

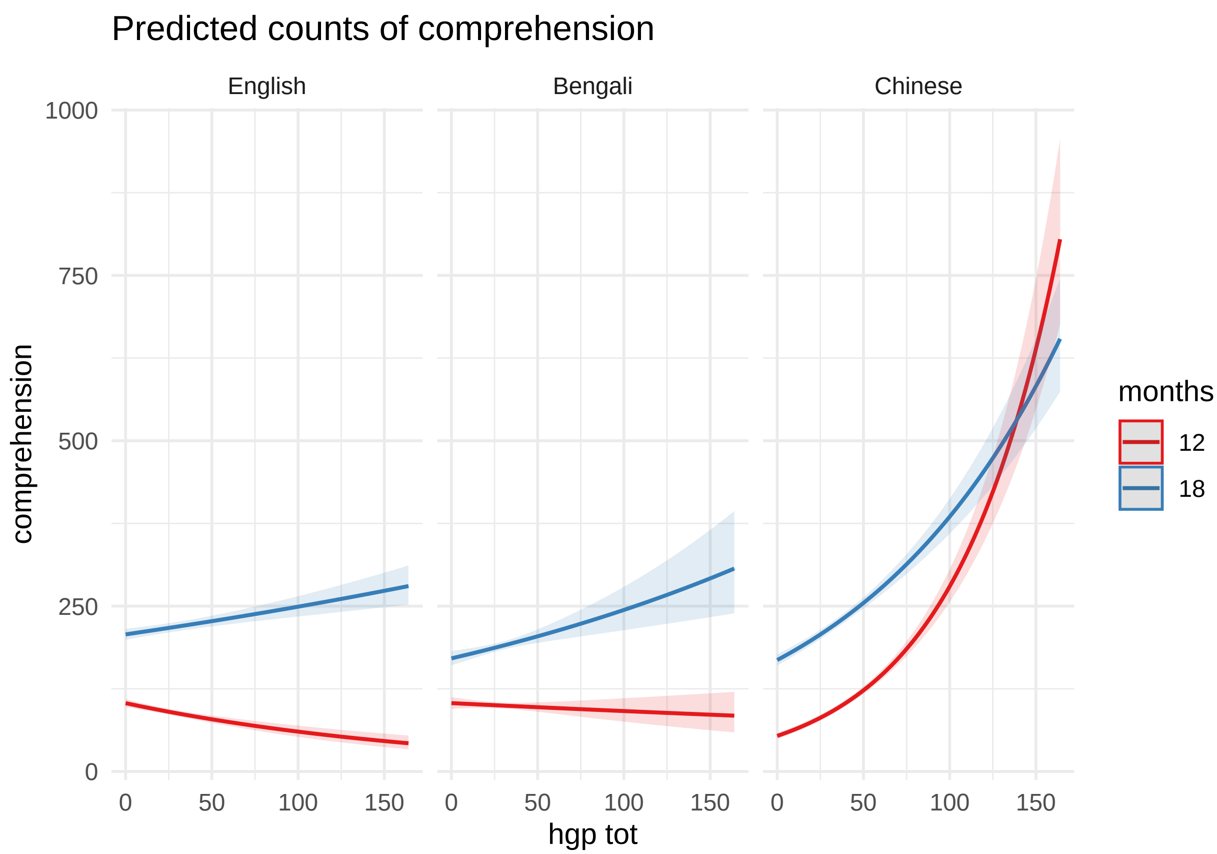


5.1.2 HoGs + points

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

```
##
## Call:
## glm(formula = comprehension ~ hgp_tot * months * background,
##     data = vocab)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -146.459   -46.622    -6.145    40.183   190.745
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       101.46876   20.27047   5.006 2.49e-06
## hgp_tot                            -0.38202    0.43931  -0.870 0.386675
## months18                          105.08054   29.26155   3.591 0.000519
## backgroundBengali                   1.82626   37.30280   0.049 0.961054
## backgroundChinese                 -61.92442   32.64270  -1.897 0.060798
## hgp_tot:months18                    0.82278    0.62495   1.317 0.191093
## hgp_tot:backgroundBengali           0.25939    1.04189   0.249 0.803913
## hgp_tot:backgroundChinese           2.36132    0.74734   3.160 0.002107
## months18:backgroundBengali        -38.30821   53.07968  -0.722 0.472208
## months18:backgroundChinese         16.00124   46.53544   0.344 0.731703
## hgp_tot:months18:backgroundBengali  -0.01282    1.47500  -0.009 0.993082
## hgp_tot:months18:backgroundChinese  -0.76045    1.05906  -0.718 0.474460
##
## (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 gaussian family taken to be 5186.06)
##
##     Null deviance: 988726  on 108  degrees of freedom
## Residual deviance: 503048  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1255
##
## Number of Fisher Scoring iterations: 2
```

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



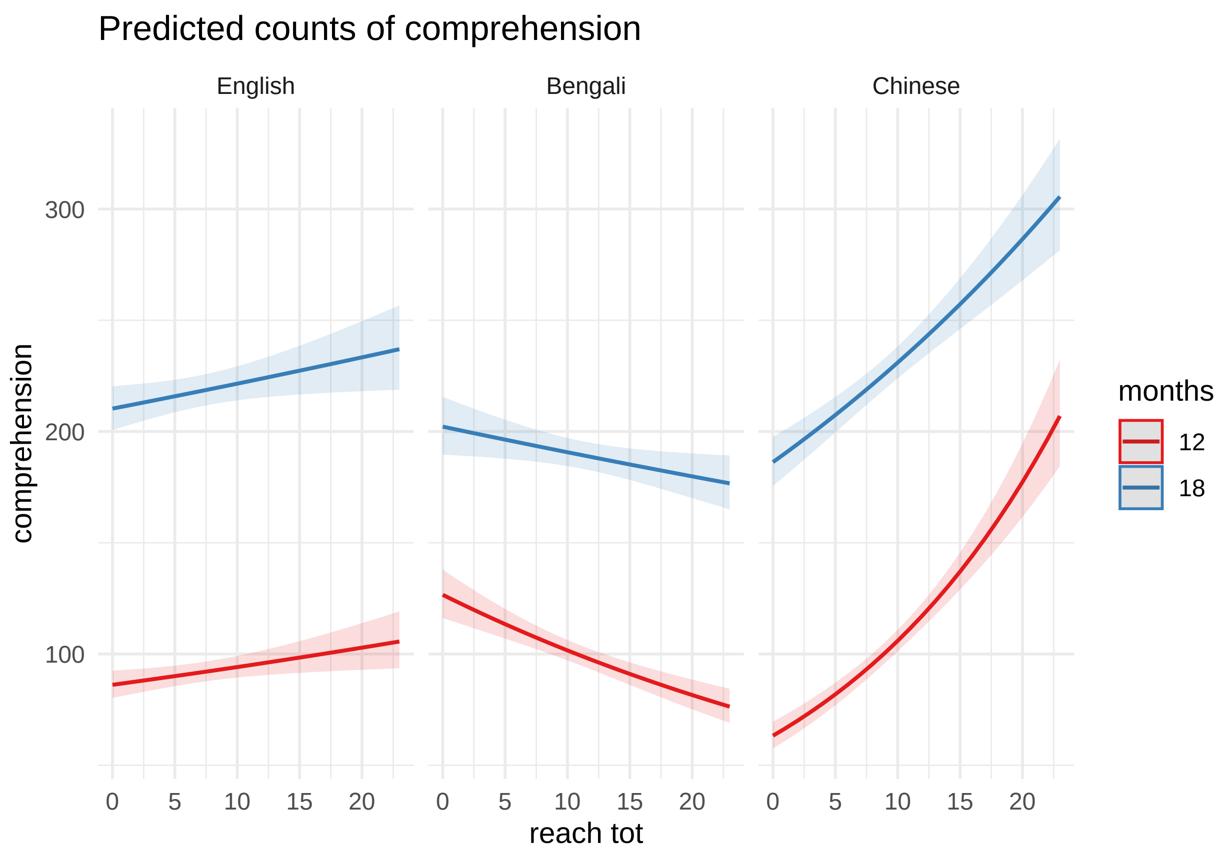
5.1.3 Reaches

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

```
##
## Call:
## glm(formula = comprehension ~ reach_tot * months * background,
##     data = vocab)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -197.044   -56.737    -5.917    49.522   209.415
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             85.9970    26.8595   3.202  0.00185
## reach_tot                                0.8338     2.7885   0.299  0.76558
```

```
## months18                              124.1729    38.3385   3.239  0.00164
## backgroundBengali                      37.9372    44.7391   0.848  0.39855
## backgroundChinese                     -28.6095    42.9986  -0.665  0.50740
## reach_tot:months18                      0.3172     3.9449   0.080  0.93609
## reach_tot:backgroundBengali            -2.9728     3.9147  -0.759  0.44946
## reach_tot:backgroundChinese             4.4409     4.3208   1.028  0.30660
## months18:backgroundBengali            -46.2654    63.4835  -0.729  0.46789
## months18:backgroundChinese              2.3506    61.0306   0.039  0.96936
## reach_tot:months18:backgroundBengali    0.7191     5.5372   0.130  0.89694
## reach_tot:months18:backgroundChinese   -0.7252     6.1114  -0.119  0.90579
##
## (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 gaussian family taken to be 6127.427)
##
##     Null deviance: 988726  on 108  degrees of freedom
## Residual deviance: 594360  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1273.2
##
## Number of Fisher Scoring iterations: 2
```

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



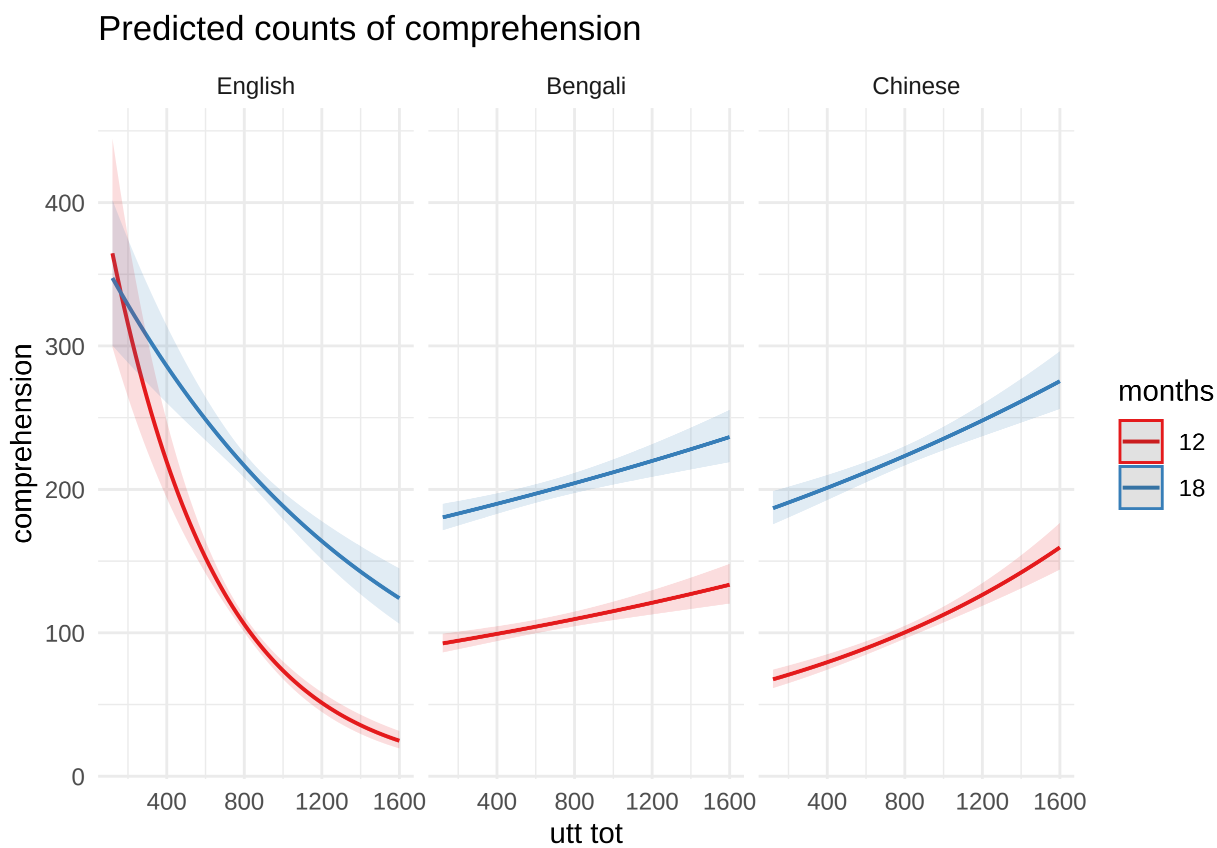
5.1.4 Maternal utterances

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

```
##
## Call:
## glm(formula = comprehension ~ utt_tot * months * background,
##     data = vocab)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -172.80   -49.42   -14.09    45.39   203.25
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         248.65611   93.77792   2.652  0.00952
## utt_tot                              -0.17106    0.10757  -1.590  0.11541
```

```
## months18                              83.13026  136.64378   0.608  0.54453
## backgroundBengali                   -160.15054   99.33997  -1.612  0.11055
## backgroundChinese                   -194.59502  102.69295  -1.895  0.06142
## utt_tot:months18                       0.02911    0.15525   0.187  0.85171
## utt_tot:backgroundBengali              0.19834    0.11603   1.709  0.09093
## utt_tot:backgroundChinese              0.23186    0.11874   1.953  0.05407
## months18:backgroundBengali             3.48893  144.29063   0.024  0.98076
## months18:backgroundChinese            39.71408  148.91143   0.267  0.79033
## utt_tot:months18:backgroundBengali    -0.01889    0.16699  -0.113  0.91022
## utt_tot:months18:backgroundChinese    -0.03052    0.17076  -0.179  0.85859
##
## (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 gaussian family taken to be 6298.419)
##
##     Null deviance: 888719  on 98  degrees of freedom
## Residual deviance: 547962  on 87  degrees of freedom
##   (21 observations deleted due to missingness)
## AIC: 1160.2
##
## Number of Fisher Scoring iterations: 2
```

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



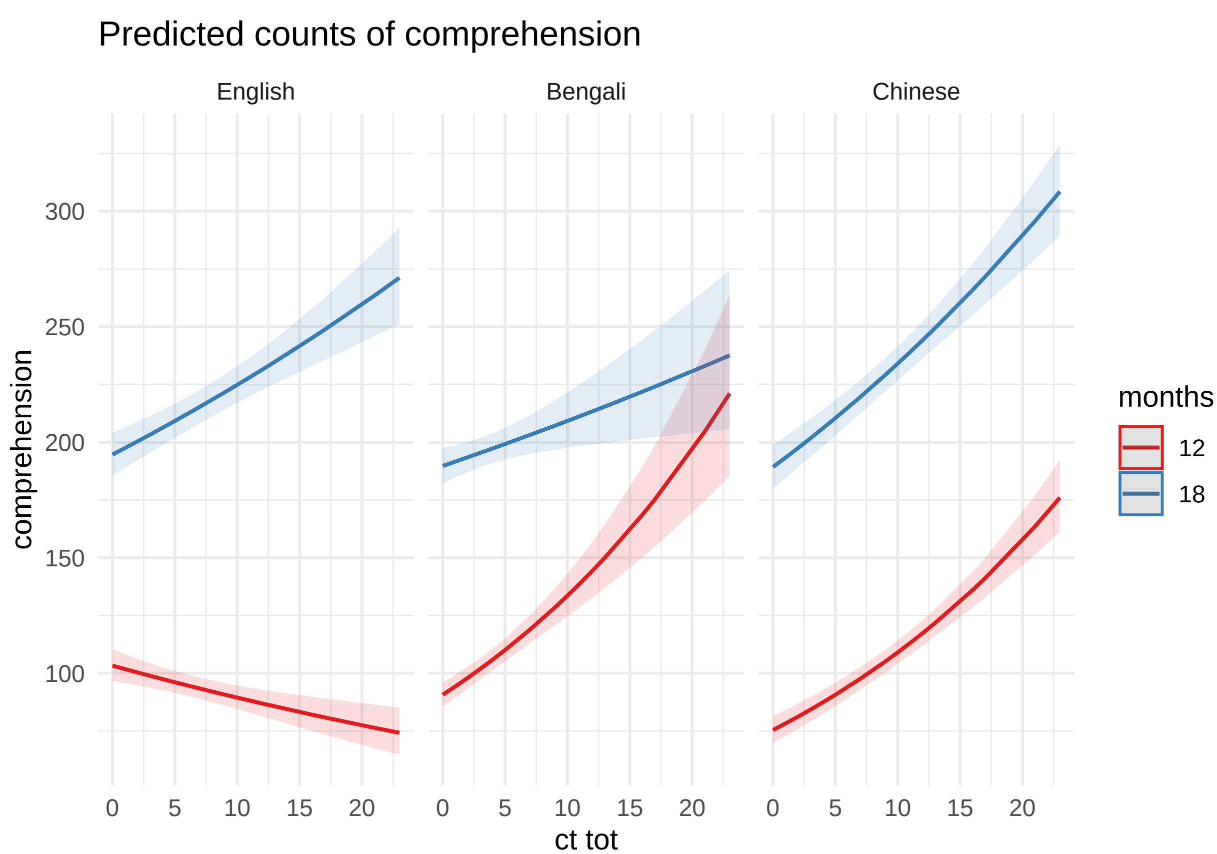
5.1.5 Contingent talks

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

```
##
## Call:
## glm(formula = comprehension ~ ct_tot * months * background, data = filter(vocab,
##     ct_tot < 30))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -160.410   -51.658    -2.591    43.598   233.163
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          102.801     26.426   3.890 0.000188 ***
## ct_tot                                -1.280      2.765  -0.463 0.644532
```

```
## months18                             90.747     37.941   2.392 0.018779 *
## backgroundBengali                   -13.281     34.624  -0.384 0.702170
## backgroundChinese                   -31.149     38.939  -0.800 0.425778
## ct_tot:months18                       4.557      3.919   1.163 0.247902
## ct_tot:backgroundBengali              5.872      4.990   1.177 0.242335
## ct_tot:backgroundChinese              5.465      3.793   1.441 0.153010
## months18:backgroundBengali            9.323     49.402   0.189 0.850722
## months18:backgroundChinese           24.136     55.456   0.435 0.664399
## ct_tot:months18:backgroundBengali    -7.169      7.063  -1.015 0.312684
## ct_tot:months18:backgroundChinese    -3.679      5.371  -0.685 0.495090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6129.831)
##
##     Null deviance: 962265  on 104  degrees of freedom
## Residual deviance: 570074  on  93  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1226.9
##
## Number of Fisher Scoring iterations: 2
```

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



5.2 Production at 12 and 18 months

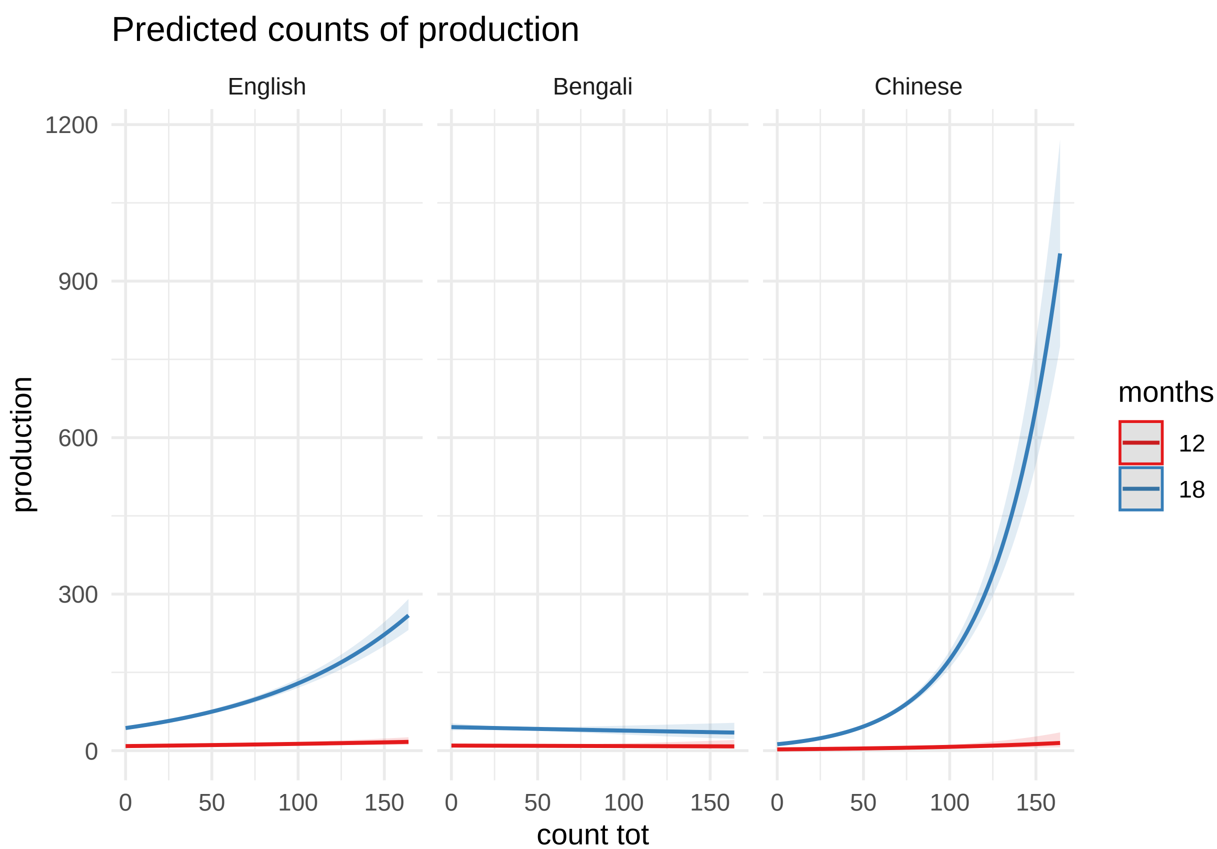
5.2.1 All gestures combined

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

```
##
## Call:
## glm(formula = production ~ count_tot * months * background, data = vocab)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -75.168  -19.248   -3.349    3.923  291.354
##
## Coefficients:
##                                         Estimate Std. Error t value
## (Intercept)                             8.506909  14.469464   0.588
## count_tot                               0.048061   0.288774   0.166
## months18                               20.534744  20.894515   0.983
## backgroundBengali                       1.191880  27.599437   0.043
## backgroundChinese                      -6.482250  24.079595  -0.269
## count_tot:months18                      1.193286   0.411042   2.903
## count_tot:backgroundBengali            -0.057777   0.603308  -0.096
## count_tot:backgroundChinese             0.001117   0.486731   0.002
## months18:backgroundBengali             14.827213  39.259493   0.378
## months18:backgroundChinese            -38.547502  34.314775  -1.123
## count_tot:months18:backgroundBengali   -1.251867   0.854480  -1.465
## count_tot:months18:backgroundChinese    0.393477   0.689919   0.570
##                                       Pr(>|t|)
## (Intercept)                            0.55795
## count_tot                              0.86816
## months18                               0.32816
## backgroundBengali                      0.96564
## backgroundChinese                      0.78835
## count_tot:months18                     0.00457 **
## count_tot:backgroundBengali            0.92390
## count_tot:backgroundChinese            0.99817
## months18:backgroundBengali             0.70650
## months18:backgroundChinese             0.26406
## count_tot:months18:backgroundBengali   0.14614
## count_tot:months18:backgroundChinese   0.56978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2210.104)
##
```

```
##     Null deviance: 358584  on 108  degrees of freedom
## Residual deviance: 214380  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1162
##
## Number of Fisher Scoring iterations: 2
```

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



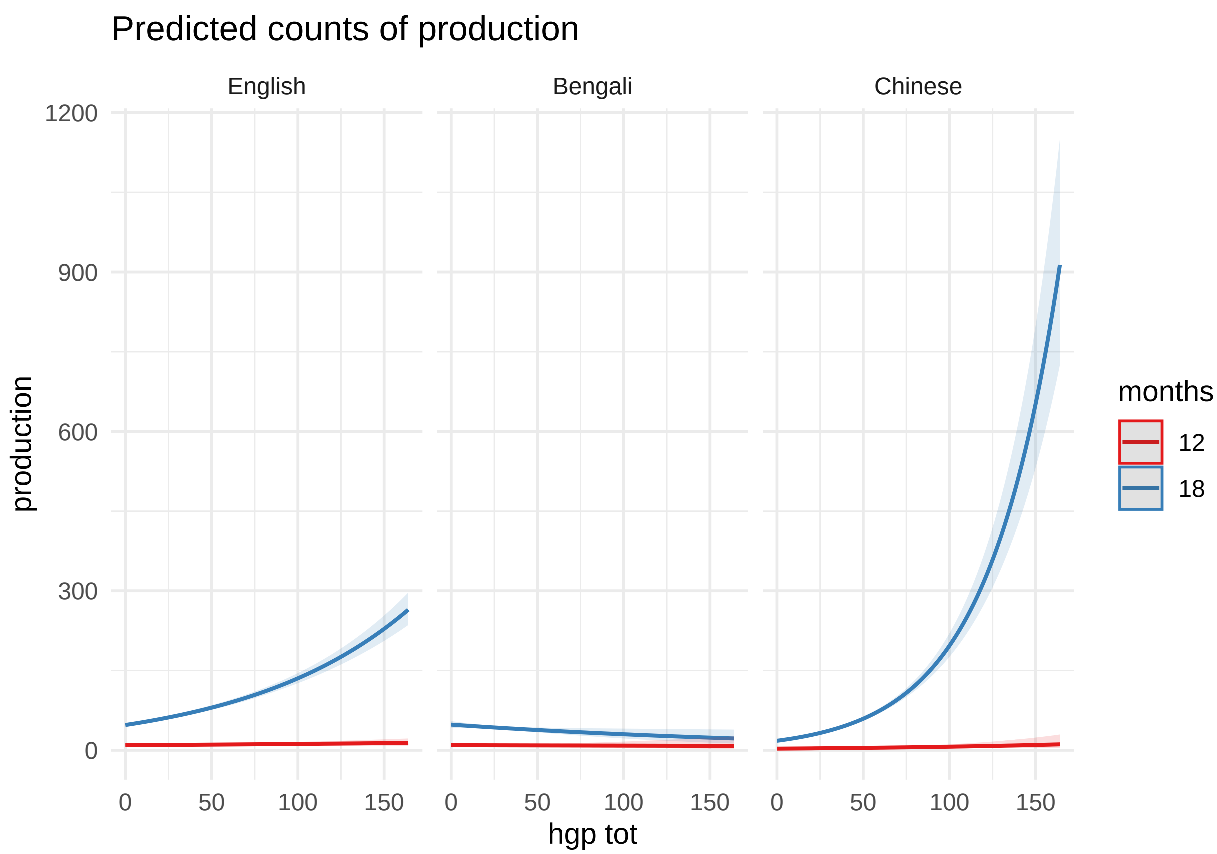
5.2.2 HoGs + point

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

```
##
## Call:
## glm(formula = production ~ hgp_tot * months * background, data = vocab)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -83.961  -19.381   -3.427    3.809  292.162
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         9.395902  13.456080   0.698  0.48668
## hgp_tot                             0.026154   0.291629   0.090  0.92873
## months18                           28.242655  19.424602   1.454  0.14918
## backgroundBengali                   0.157087  24.762597   0.006  0.99495
## backgroundChinese                  -6.494013  21.669100  -0.300  0.76505
## hgp_tot:months18                    1.221406   0.414859   2.944  0.00405
## hgp_tot:backgroundBengali          -0.034578   0.691632  -0.050  0.96023
## hgp_tot:backgroundChinese           0.008524   0.496106   0.017  0.98633
## months18:backgroundBengali         10.068667  35.235712   0.286  0.77568
## months18:backgroundChinese        -27.512484  30.891468  -0.891  0.37534
## hgp_tot:months18:backgroundBengali -1.408163   0.979143  -1.438  0.15361
## hgp_tot:months18:backgroundChinese  0.199396   0.703033   0.284  0.77730
##
## (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 gaussian family taken to be 2285.32)
##
##     Null deviance: 358584  on 108  degrees of freedom
## Residual deviance: 221676  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1165.6
##
## Number of Fisher Scoring iterations: 2
```

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



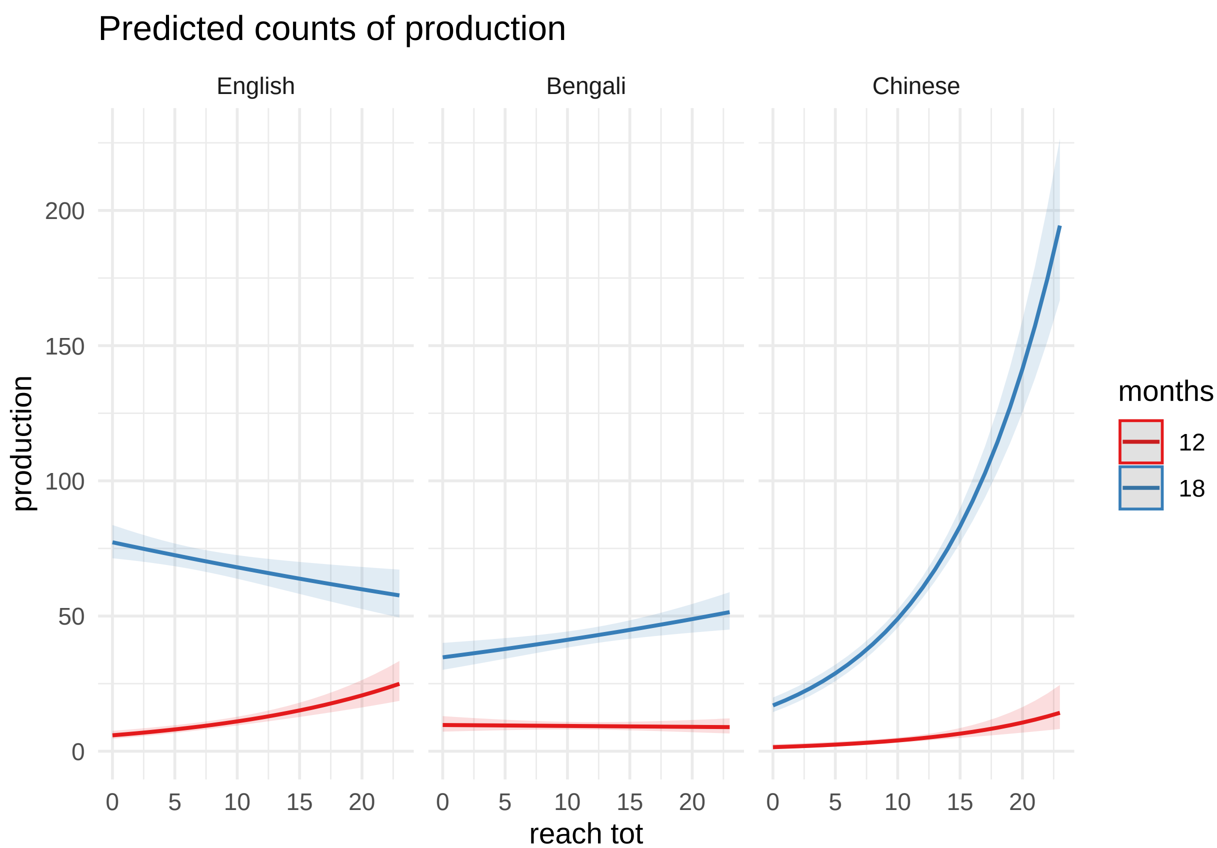
5.2.3 Reaches

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

```
##
## Call:
## glm(formula = production ~ reach_tot * months * background, data = vocab)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -75.884  -24.285   -3.257    4.118  288.621
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                               4.9832    18.3147   0.272  0.78613
## reach_tot                                 0.7246     1.9014   0.381  0.70397
## months18                                 72.0425    26.1419   2.756  0.00699
```

```
## backgroundBengali                        4.7153    30.5063   0.155  0.87748
## backgroundChinese                       -4.2472    29.3195  -0.145  0.88512
## reach_tot:months18                      -1.5949     2.6899  -0.593  0.55461
## reach_tot:backgroundBengali             -0.7586     2.6693  -0.284  0.77687
## reach_tot:backgroundChinese             -0.3406     2.9462  -0.116  0.90820
## months18:backgroundBengali             -47.6450    43.2876  -1.101  0.27377
## months18:backgroundChinese             -66.9012    41.6150  -1.608  0.11117
## reach_tot:months18:backgroundBengali     2.3634     3.7756   0.626  0.53282
## reach_tot:months18:backgroundChinese     6.3692     4.1672   1.528  0.12966
##
## (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 gaussian family taken to be 2848.935)
##
##     Null deviance: 358584  on 108  degrees of freedom
## Residual deviance: 276347  on  97  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1189.7
##
## Number of Fisher Scoring iterations: 2
```

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



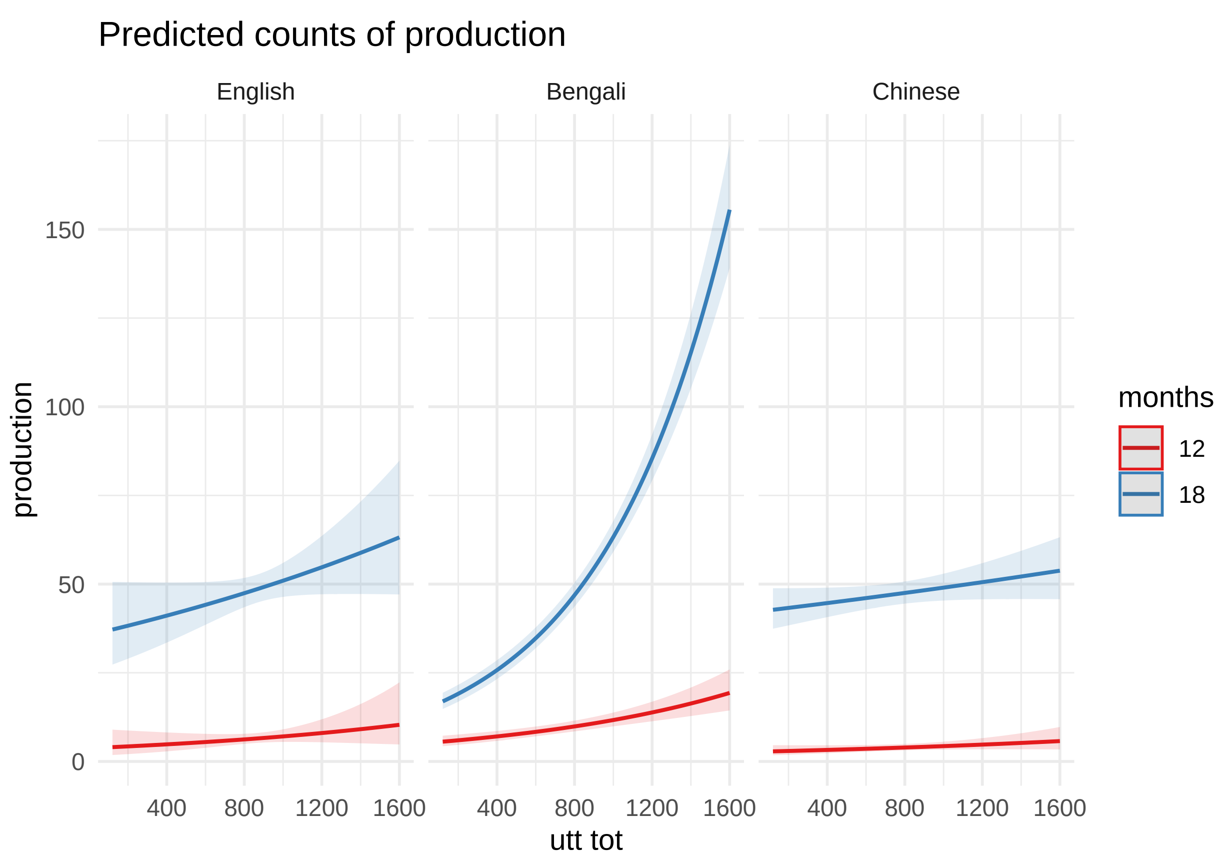
5.2.4 Maternal utterances

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

```
##
## Call:
## glm(formula = production ~ utt_tot * months * background, data = vocab)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -116.668   -17.794    -3.113     3.738   215.516
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            2.891384  57.096048   0.051    0.960
## utt_tot                                0.004213   0.065491   0.064    0.949
## months18                              30.553534  83.194642   0.367    0.714
```

```
## backgroundBengali                       0.773351  60.482465   0.013    0.990
## backgroundChinese                      -0.437357  62.523906  -0.007    0.994
## utt_tot:months18                        0.013408   0.094525   0.142    0.888
## utt_tot:backgroundBengali               0.004590   0.070642   0.065    0.948
## utt_tot:backgroundChinese              -0.002290   0.072293  -0.032    0.975
## months18:backgroundBengali            -40.559113  87.850374  -0.462    0.645
## months18:backgroundChinese              8.639304  90.663712   0.095    0.924
## utt_tot:months18:backgroundBengali      0.061666   0.101674   0.607    0.546
## utt_tot:months18:backgroundChinese    -0.007907   0.103969  -0.076    0.940
##
## (Dispersion parameter for gaussian family taken to be 2334.761)
##
##     Null deviance: 268071  on 98  degrees of freedom
## Residual deviance: 203124  on 87  degrees of freedom
##   (21 observations deleted due to missingness)
## AIC: 1062
##
## Number of Fisher Scoring iterations: 2
```

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

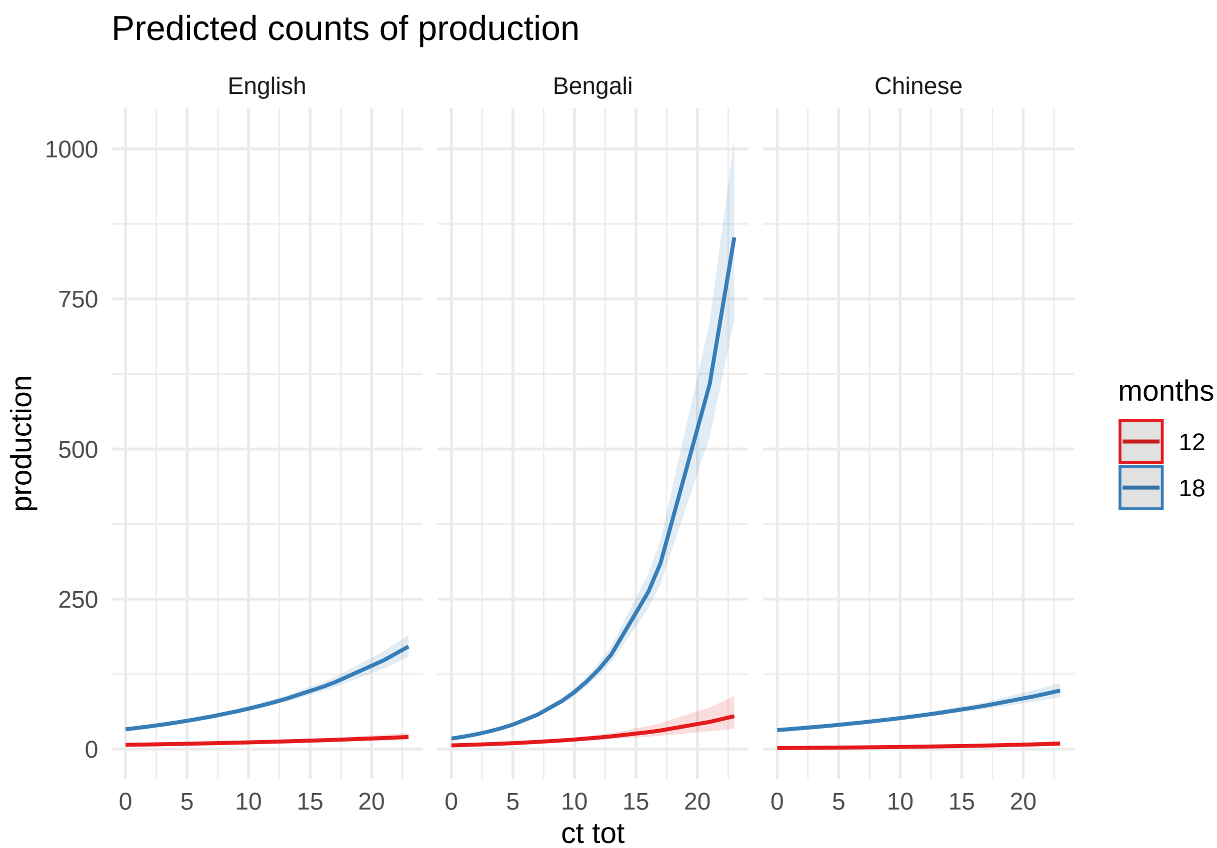


5.2.5 Contingent talks

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

```
##
## Call:
## glm(formula = production ~ ct_tot * months * background, data = filter(vocab,
##     ct_tot < 30))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -109.012   -18.528    -2.762     4.880   197.197
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         6.5784    15.2861   0.430    0.668
## ct_tot                              0.5413     1.5993   0.338    0.736
## months18                           18.1570    21.9472   0.827    0.410
## backgroundBengali                  -1.0840    20.0286  -0.054    0.957
## backgroundChinese                  -5.2748    22.5242  -0.234    0.815
## ct_tot:months18                     4.9654     2.2671   2.190    0.031 *
## ct_tot:backgroundBengali            0.6490     2.8867   0.225    0.823
## ct_tot:backgroundChinese           -0.2496     2.1941  -0.114    0.910
## months18:backgroundBengali        -17.5642    28.5769  -0.615    0.540
## months18:backgroundChinese          9.0669    32.0784   0.283    0.778
## ct_tot:months18:backgroundBengali   5.6368     4.0853   1.380    0.171
## ct_tot:months18:backgroundChinese  -2.5640     3.1068  -0.825    0.411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2051.084)
##
##     Null deviance: 325810  on 104  degrees of freedom
## Residual deviance: 190751  on  93  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1112
##
## Number of Fisher Scoring iterations: 2
```

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



Predicted counts of production
English
Bengali
Chinese
1000
750
500
250
0
production
0
5
10
15
20
ct tot
months
12
18

6 Number of observations

The following sections report the number of observations (excluding NAs) used in the models above.

6.1 Analysis 1a

6.1.1 Reaches

```
reach_tot %>%
  group_by(back_o, months) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 9 x 3
## # Groups:   back_o [3]
##   back_o  months     n
##   <ord>    <dbl> <int>
## 1 English     10    19
## 2 English     11    19
## 3 English     12    18
## 4 Bengali     10    20
## 5 Bengali     11    19
## 6 Bengali     12    19
## 7 Chinese     10    18
## 8 Chinese     11    19
## 9 Chinese     12    20
```

6.1.2 HoGs

```
hg_tot %>%
  group_by(back_o, months) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 9 x 3
## # Groups:   back_o [3]
##   back_o  months     n
##   <ord>    <dbl> <int>
## 1 English     10    19
## 2 English     11    19
## 3 English     12    18
## 4 Bengali     10    20
## 5 Bengali     11    19
## 6 Bengali     12    19
## 7 Chinese     10    18
## 8 Chinese     11    19
## 9 Chinese     12    20
```

6.1.3 Points

```r
point_tot %>%
  group_by(back_o, months) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 9 x 3
## # Groups:   back_o [3]
##   back_o  months     n
##   <ord>    <dbl> <int>
## 1 English     10    19
## 2 English     11    19
## 3 English     12    18
## 4 Bengali     10    20
## 5 Bengali     11    19
## 6 Bengali     12    19
## 7 Chinese     10    18
## 8 Chinese     11    19
## 9 Chinese     12    20
```

6.2 Analysis 1b

6.2.1 Maternal utterances

```r
utterances_tot %>%
  group_by(back_o, months) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 9 x 3
## # Groups:   back_o [3]
##   back_o  months     n
##   <ord>    <dbl> <int>
## 1 English     10    17
## 2 English     11    18
## 3 English     12    16
## 4 Bengali     10    20
## 5 Bengali     11    19
## 6 Bengali     12    18
## 7 Chinese     10    19
## 8 Chinese     11    20
## 9 Chinese     12    20
```

6.2.2 Maternal CTs

```r
all_tot %>%
  group_by(back_o, months) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 9 x 3
## # Groups:   back_o [3]
##   back_o  months     n
##   <ord>    <dbl> <int>
## 1 English     10    19
## 2 English     11    19
## 3 English     12    18
## 4 Bengali     10    20
## 5 Bengali     11    19
## 6 Bengali     12    19
## 7 Chinese     10    18
## 8 Chinese     11    19
## 9 Chinese     12    20
```

6.3 Analysis 1c

6.3.1 Reaches

```
reach_point_lead %>%
  group_by(back_o) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 3 x 2
##   back_o      n
##   <ord>   <int>
## 1 English    37
## 2 Bengali    38
## 3 Chinese    37
```

6.3.2 HoGs

```
hg_point_lead %>%
  group_by(back_o) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 3 x 2
##   back_o      n
##   <ord>   <int>
## 1 English    37
## 2 Bengali    38
## 3 Chinese    37
```

6.4 Analysis 2

The counts apply both to the comprehension and production analyses.

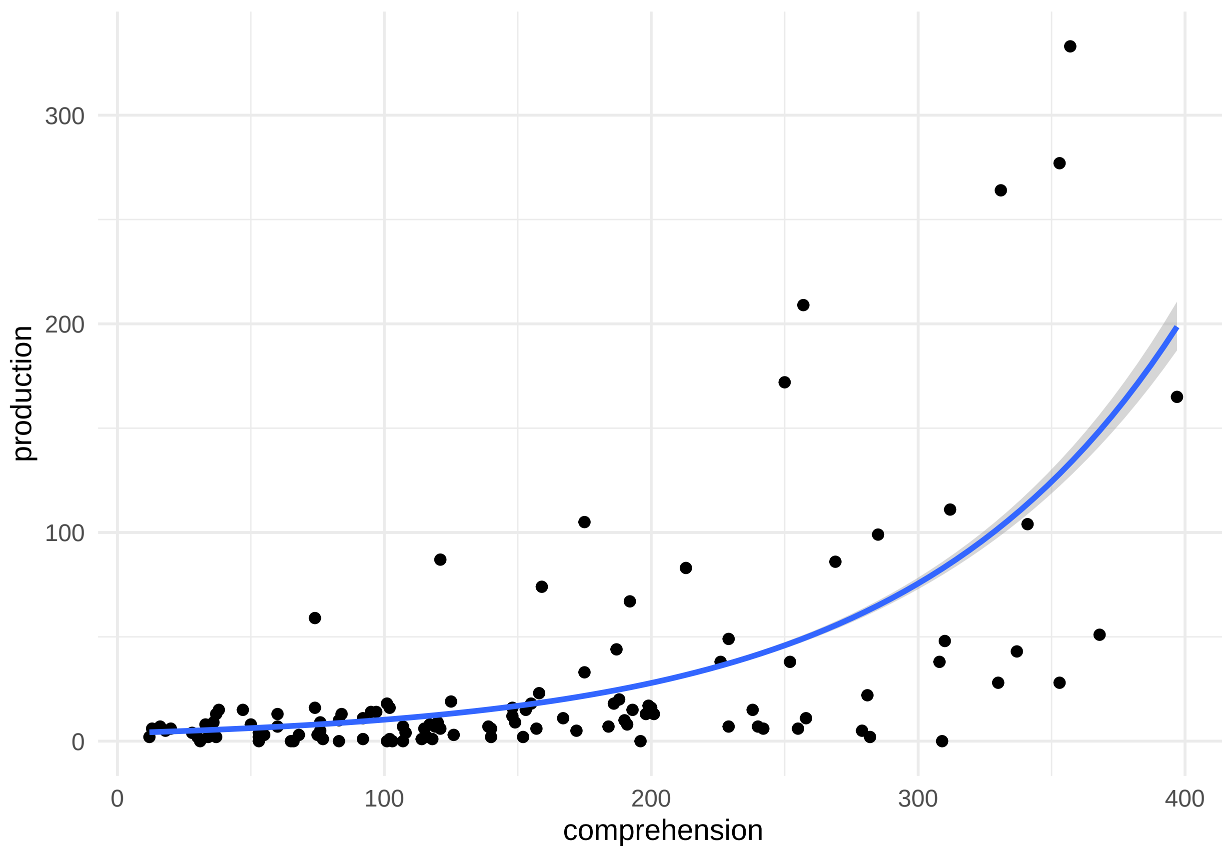
```r
vocab %>%
  group_by(background) %>%
  na.omit() %>%
  summarise(n = n())
```

```
## # A tibble: 3 x 2
##   background     n
##   <fct>      <int>
## 1 English       25
## 2 Bengali       34
## 3 Chinese       34
```

7 Correlation of vocabulary scores and maternal scores

```r
vocab %>%
  ggplot(aes(comprehension, production)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = poisson))
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).

## Warning: Removed 3 rows containing missing values (geom_point).
```

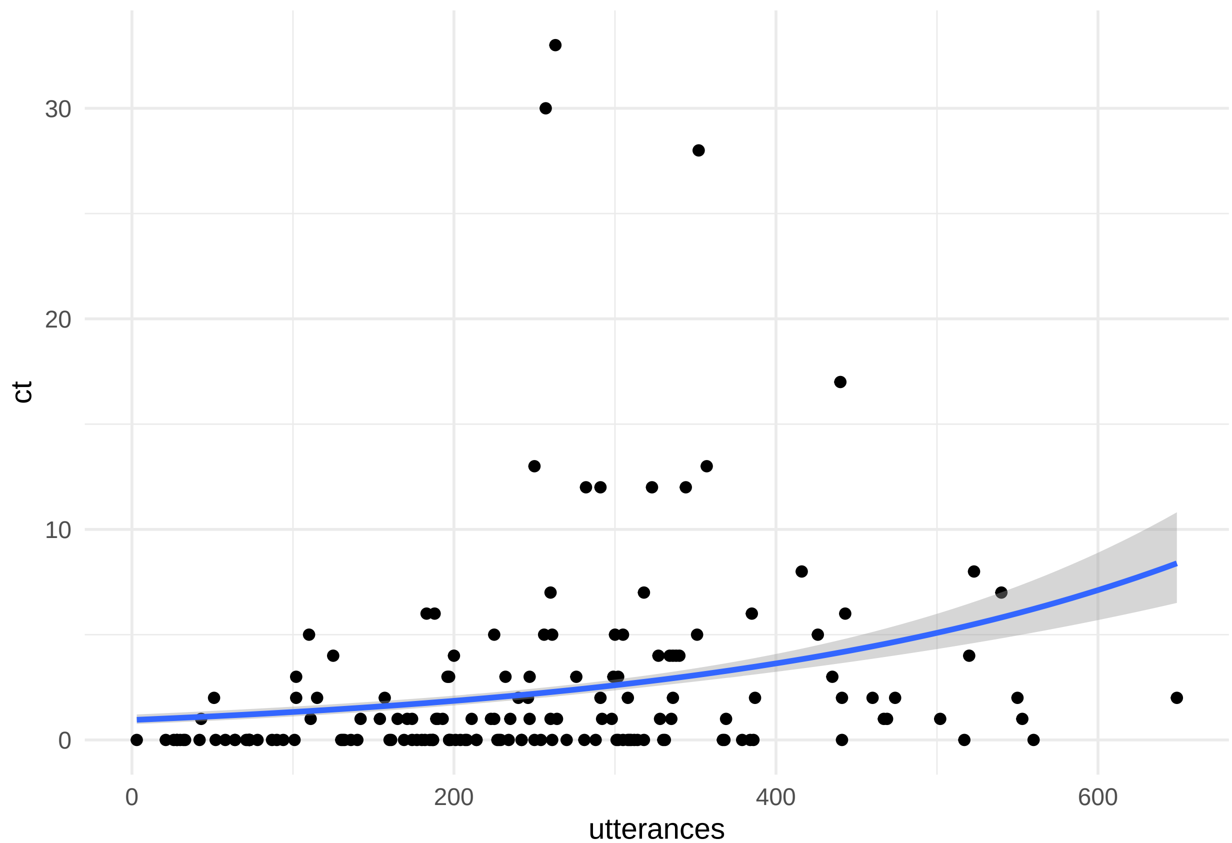


```
all_tot %>%
  left_join(utterances_tot) %>%
  ggplot(aes(utterances, ct)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = poisson))
```

```
## Joining, by = c("dyad", "back_o", "months")
```

```
## Warning: Removed 12 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```



8 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
```