Model_comparisons.R

gioia

2020-05-21

```
################################
### Model comparison
################################

library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(bayesplot)
```

```
## This is bayesplot version 1.7.1
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(loo)
```

```
## This is loo version 2.2.0
## - Online documentation and vignettes at mc-stan.org/loo
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' ar
##
## Attaching package: 'loo'
## The following object is masked from 'package:rstan':
##
##     loo
```

```
theme_set(bayesplot::theme_default())
rstan_options(auto_write = TRUE)
# data
pest_data <- readRDS('data/pest_data.RDS')
str(pest_data)
```

```
## 'data.frame':    120 obs. of  14 variables:
##  $ mus                 : num  0.369 0.359 0.282 0.129 0.452 ...
##  $ building_id         : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ wk_ind              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ date                : Date, format: "2017-01-15" "2017-02-14" ...
##  $ traps               : num  8 8 9 10 11 11 10 10 9 9 ...
##  $ floors              : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ sq_footage_p_floor  : num  5149 5149 5149 5149 5149 ...
##  $ live_in_super       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ monthly_average_rent: num  3847 3847 3847 3847 3847 ...
##  $ average_tenant_age  : num  53.9 53.9 53.9 53.9 53.9 ...
##  $ age_of_building     : num  47 47 47 47 47 47 47 47 47 47 ...
##  $ total_sq_foot       : num  41192 41192 41192 41192 41192 ...
##  $ month               : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ complaints          : num  1 3 0 1 0 0 4 3 2 2 ...
```

```
N_buildings <- length(unique(pest_data$building_id))
N_buildings
```

```
## [1] 10
```

```
N_months <- length(unique(pest_data$date))

## Data acquisition

stan_dat_simple <- list(
  N = nrow(pest_data),
  complaints = pest_data$complaints,
  traps = pest_data$traps
)

pest_data <- pest_data %>%
  mutate(
```

```
    building_fac = factor(building_id, levels = unique(building_id)),
    building_idx = as.integer(building_fac),
    ids = rep(1:N_months, N_buildings),
    mo_idx = lubridate::month(date)
  )

# Center and rescale the building specific data
building_data <- pest_data %>%
  select(
    building_idx,
    live_in_super,
    age_of_building,
    total_sq_foot,
    average_tenant_age,
    monthly_average_rent
  ) %>%
  unique() %>%
  arrange(building_idx) %>%
  select(-building_idx) %>%
  scale(scale=FALSE) %>%
  as.data.frame() %>%
  mutate( # scale by constants
    age_of_building = age_of_building / 10,
    total_sq_foot = total_sq_foot / 10000,
    average_tenant_age = average_tenant_age / 10,
    monthly_average_rent = monthly_average_rent / 1000
  ) %>%
  as.matrix()
str(building_data)
```

```
##  num [1:10, 1:5] -0.3 -0.3 -0.3 -0.3 0.7 -0.3 -0.3 0.7 -0.3 0.7 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "live_in_super" "age_of_building" "total_sq_foot" "average_tenant_age" ...
```

```
# Make data list for Stan
stan_dat_hier <-
  with(pest_data,
       list(complaints = complaints,
            traps = traps,
            N = length(traps),
            J = N_buildings,
            M = N_months,
            log_sq_foot = log(pest_data$total_sq_foot/1e4),
            building_data = building_data[,-3],
            mo_idx = as.integer(as.factor(date)),
            K = 4,
            building_idx = building_idx
       )
  )



## Models fit
```

```r
# simple poisson
comp_model_P <- stan_model('To_compare_models/simple_poisson_regression.stan')
fit_P_real_data <- sampling(comp_model_P, data = stan_dat_simple)
```

```
##
## SAMPLING FOR MODEL 'simple_poisson_regression' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000121 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.21 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.261239 seconds (Warm-up)
## Chain 1:                0.226545 seconds (Sampling)
## Chain 1:                0.487784 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'simple_poisson_regression' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.5e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.14514 seconds (Warm-up)
## Chain 2:                0.095421 seconds (Sampling)
## Chain 2:                0.240561 seconds (Total)
```

```
## Chain 2:
##
## SAMPLING FOR MODEL 'simple_poisson_regression' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.1 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.167045 seconds (Warm-up)
## Chain 3:                0.198774 seconds (Sampling)
## Chain 3:                0.365819 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'simple_poisson_regression' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.4e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.14 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.180254 seconds (Warm-up)
## Chain 4:                0.153306 seconds (Sampling)
## Chain 4:                0.33356 seconds (Total)
## Chain 4:
```

```
# multiple poisson
stan_dat_simple$log_sq_foot <- log(pest_data$total_sq_foot/1e4)
```

```r
stan_dat_simple$live_in_super <- pest_data$live_in_super
comp_model_P_mult <- stan_model('To_compare_models/multiple_poisson_regression.stan')
fit_model_P_mult_real <- sampling(comp_model_P_mult, data = stan_dat_simple)
```

```
##
## SAMPLING FOR MODEL 'multiple_poisson_regression' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.8e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.28 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.245616 seconds (Warm-up)
## Chain 1:                0.238833 seconds (Sampling)
## Chain 1:                0.484449 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'multiple_poisson_regression' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.24888 seconds (Warm-up)
## Chain 2:                0.220418 seconds (Sampling)
## Chain 2:                0.469298 seconds (Total)
## Chain 2:
```

```
##
## SAMPLING FOR MODEL 'multiple_poisson_regression' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.249662 seconds (Warm-up)
## Chain 3:                0.220059 seconds (Sampling)
## Chain 3:                0.469721 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'multiple_poisson_regression' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.249068 seconds (Warm-up)
## Chain 4:                0.231587 seconds (Sampling)
## Chain 4:                0.480655 seconds (Total)
## Chain 4:
```

```r
# negative binomial
comp_model_NB <- stan_model('To_compare_models/multiple_NB_regression.stan')
fitted_model_NB <- sampling(comp_model_NB, data = stan_dat_simple)
```

```
##
## SAMPLING FOR MODEL 'multiple_NB_regression' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 4.9e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.49 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.944765 seconds (Warm-up)
## Chain 1:                0.750283 seconds (Sampling)
## Chain 1:                1.69505 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'multiple_NB_regression' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 5.2e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.52 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.865498 seconds (Warm-up)
## Chain 2:                0.76354 seconds (Sampling)
## Chain 2:                1.62904 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'multiple_NB_regression' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 5.4e-05 seconds
```

```
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.54 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.837964 seconds (Warm-up)
## Chain 3:                0.75995 seconds (Sampling)
## Chain 3:                1.59791 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'multiple_NB_regression' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 5.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.51 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.863313 seconds (Warm-up)
## Chain 4:                0.708232 seconds (Sampling)
## Chain 4:                1.57154 seconds (Total)
## Chain 4:
```

```r
# hier NB regression
comp_model_NB_hier <- stan_model('To_compare_models/hier_NB_regression.stan')
fitted_model_NB_hier <-
  sampling(
    comp_model_NB_hier,
    data = stan_dat_hier,
    chains = 4,
```

```
    #cores = 4,
    iter = 4000
  )
```

```
##
## SAMPLING FOR MODEL 'hier_NB_regression' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 5.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.57 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 2.83045 seconds (Warm-up)
## Chain 1:                1.90252 seconds (Sampling)
## Chain 1:                4.73297 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hier_NB_regression' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3.6e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.36 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 3.26941 seconds (Warm-up)
## Chain 2:                2.61957 seconds (Sampling)
## Chain 2:                5.88898 seconds (Total)
## Chain 2:
```

```
##
## SAMPLING FOR MODEL 'hier_NB_regression' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 4e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.4 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 2.8541 seconds (Warm-up)
## Chain 3:                1.93276 seconds (Sampling)
## Chain 3:                4.78686 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hier_NB_regression' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 5.4e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.54 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 2.76047 seconds (Warm-up)
## Chain 4:                3.53031 seconds (Sampling)
## Chain 4:                6.29078 seconds (Total)
## Chain 4:

## Warning: There were 435 divergent transitions after warmup. Increasing adapt_delta above 0.8 may hel
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
# hier NCP NB regression
comp_model_NB_hier_ncp <- stan_model('To_compare_models/hier_NB_regression_ncp.stan')
fitted_model_NB_hier_ncp <- sampling(comp_model_NB_hier_ncp,
                                     data = stan_dat_hier,
                                     chains = 4,
                                     #cores = 4,
                                     iter=4000)
```

```
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000103 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.03 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 2.87328 seconds (Warm-up)
## Chain 1:                2.10388 seconds (Sampling)
## Chain 1:                4.97716 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.4 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 2.77813 seconds (Warm-up)
## Chain 2:                2.35575 seconds (Sampling)
## Chain 2:                5.13388 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 3.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.38 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 2.73638 seconds (Warm-up)
## Chain 3:                2.08735 seconds (Sampling)
## Chain 3:                4.82373 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.5 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
```

```
## Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 2.7735 seconds (Warm-up)
## Chain 4:                2.18198 seconds (Sampling)
## Chain 4:                4.95548 seconds (Total)
## Chain 4:
```

```
# hier NB slopes
comp_model_NB_hier_slopes <- stan_model('To_compare_models/hier_NB_regression_ncp_slopes_mod.stan')
fitted_model_NB_hier_slopes <-
  sampling(
    comp_model_NB_hier_slopes,
    data = stan_dat_hier,
    chains = 4, #cores = 4,
    control = list(adapt_delta = 0.95)
  )
```

```
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.8e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.78 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 3.67301 seconds (Warm-up)
## Chain 1:                2.37499 seconds (Sampling)
## Chain 1:                6.048 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 5.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.51 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 3.36474 seconds (Warm-up)
## Chain 2:                2.2473 seconds (Sampling)
## Chain 2:                5.61204 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 6.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.65 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.5319 seconds (Warm-up)
## Chain 3:                2.43537 seconds (Sampling)
## Chain 3:                5.96726 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 4e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.4 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
```

```
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 3.6972 seconds (Warm-up)
## Chain 4:                2.2965 seconds (Sampling)
## Chain 4:                5.99371 seconds (Total)
## Chain 4:
```

```
comp_model_NB_hier_slopes_mos <- stan_model('To_compare_models/hier_NB_regression_ncp_slopes_mod_mos.sta
```

```
## recompiling to avoid crashing R session
```

```
## Trying to compile a simple C file
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
##                 ^
##                 ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fa
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

```
fitted_model_NB_hier_slopes_mos <-
  sampling(
    comp_model_NB_hier_slopes_mos,
    data = stan_dat_hier,
    chains = 4, #cores = 4,
    control = list(adapt_delta = 0.95)
  )
```

```
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod_mos' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.4e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.74 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 3.95707 seconds (Warm-up)
## Chain 1:                2.51139 seconds (Sampling)
## Chain 1:                6.46846 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod_mos' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 6.5e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.65 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 3.58645 seconds (Warm-up)
## Chain 2:                2.41507 seconds (Sampling)
## Chain 2:                6.00153 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod_mos' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 7e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.7 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.85421 seconds (Warm-up)
## Chain 3:                2.47367 seconds (Sampling)
## Chain 3:                6.32789 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hier_NB_regression_ncp_slopes_mod_mos' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 7.2e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.72 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 4.22656 seconds (Warm-up)
## Chain 4:                2.47451 seconds (Sampling)
## Chain 4:                6.70107 seconds (Total)
## Chain 4:
```

```
## Warning: There were 3 divergent transitions after warmup. Increasing adapt_delta above 0.95 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Extract pointwise log-likelihood and
## compute loo and waic

# loo
log_lik_pois <- extract_log_lik(fit_P_real_data)
loo_pois <- loo(log_lik_pois)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
waic_pois <- waic(log_lik_pois)
```

```
## Warning:
## 4 (3.3%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_mult_pois <- extract_log_lik(fit_model_P_mult_real)
loo_mult_pois <- loo(log_lik_mult_pois)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
waic_mult_pois <- waic(log_lik_mult_pois)
```

```
## Warning:
## 5 (4.2%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_mult_NB <- extract_log_lik(fitted_model_NB)
loo_mult_NB <- loo(log_lik_mult_NB)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
waic_mult_NB <- waic(log_lik_mult_NB)
```

```
## Warning:
## 1 (0.8%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_hier_NB <- extract_log_lik(fitted_model_NB_hier)
loo_hier_NB <- loo(log_lik_hier_NB)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
waic_hier_NB <- waic(log_lik_hier_NB)
```

```
## Warning:
## 2 (1.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_hier_NB_ncp <- extract_log_lik(fitted_model_NB_hier_ncp)
loo_hier_NB_ncp <- loo(log_lik_hier_NB_ncp)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
waic_hier_NB_ncp <- waic(log_lik_hier_NB_ncp)
```

```
## Warning:
## 2 (1.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_slopes <- extract_log_lik(fitted_model_NB_hier_slopes)
loo_slopes <- loo(log_lik_slopes)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
## Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic') for deta
```

```
waic_slopes <- waic(log_lik_slopes)
```

```
## Warning:
## 2 (1.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
log_lik_slopes_mos <- extract_log_lik(fitted_model_NB_hier_slopes_mos)
loo_slopes_mos <- loo(log_lik_slopes_mos)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
waic_slopes_mos <- waic(log_lik_slopes_mos)
```

```
## Warning:
## 4 (3.3%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
compare(loo_pois, loo_mult_pois,
        loo_mult_NB, loo_hier_NB,
        loo_hier_NB_ncp, loo_slopes,
        loo_slopes_mos)
```

```
## Warning: 'compare' is deprecated.
## Use 'loo_compare' instead.
## See help("Deprecated")
```

```
##                elpd_diff se_diff elpd_loo p_loo  looic
## loo_mult_NB        0.0       0.0  -279.5      3.6  559.0
## loo_hier_NB       -2.9       1.4  -282.3      8.3  564.7
## loo_hier_NB_ncp   -3.0       1.2  -282.5      8.2  564.9
## loo_slopes_mos    -4.6       2.5  -284.1     14.3  568.2
## loo_slopes        -4.7       1.9  -284.2     11.0  568.4
## loo_mult_pois    -52.1      15.2  -331.5      8.7  663.1
## loo_pois         -72.2      17.9  -351.6      6.1  703.3
```

```
looic1<-loo_pois$estimates[3,1]
looic2<-loo_mult_pois$estimates[3,1]
looic3<-loo_mult_NB$estimates[3,1]
looic4<-loo_hier_NB$estimates[3,1]
looic5<-loo_hier_NB_ncp$estimates[3,1]
looic6<-loo_slopes$estimates[3,1]
looic7<-loo_slopes_mos$estimates[3,1]

looics <-c(looic1, looic2, looic3, looic4, looic5,
           looic6, looic7)

waic1<-waic_pois$estimates[3,1]
waic2<-waic_mult_pois$estimates[3,1]
waic3<-waic_mult_NB$estimates[3,1]
waic4<-waic_hier_NB$estimates[3,1]
waic5<-waic_hier_NB_ncp$estimates[3,1]
waic6<-waic_slopes$estimates[3,1]
waic7<-waic_slopes_mos$estimates[3,1]

waics <-c(waic1, waic2, waic3, waic4, waic5,
           waic6, waic7)
```

```
par(xaxt="n", mfrow=c(1,2))
plot(looics, type="b", xlab="", ylab="LOOIC")
par(xaxt="s")
axis(1, c(1:7), c("Pois", "Pois mult", "NB",
                  "Hier NB", "Hier NB ncp", "NB slopes", "NB slopes month"), las=2)
par(xaxt="n")
plot(waics, type="b", xlab="", ylab="WAIC")
par(xaxt="s")
axis(1, c(1:7), c("Pois", "Pois mult", "NB",
                  "Hier NB", "Hier NB ncp", "NB slopes", "NB slopes month"), las=2)
```