

# Supplementary material - Orexin Cell transplant reduces behavioral arrest severity in narcoleptic mice.

*Ana Clementina Equihua-Benítez, Julián A. Equihua-Benítez, Khalil Guzmán-Vásquez, Oscar Prospero-García & René Drucker-Colín.*

## Introduction

The differences in groups was analyzed using Bayesian multilevel models. These analyses were carried out using the R language (Team, 2018) by means of the brms package (Bürkner, 2017) which is a high level interface to the Stan probabilistic programming language (Carpenter et al., 2017).

A total of six models were fitted for this purpose. Three (1-3) to analyze the treatment effects on BAs, and one model (4-6) to analyze the treatment effects on sleep architecture.

All models showed a good convergence and fit.

1. A model fit to predict BA duration, with response distribution set to log-normal, with the variable group as the population-level parameter of interest and a varying-subject intercept, following the terminology of (Bürkner and Vuorre, 2019) .
2. A model to predict the total number of BAs per sampled hour as the total sampled time varied slightly for each subject. Response distribution is set to Poisson, also with the variable group as the population-level parameter of interest. But as what is modeled are rates, we introduced a log(hours) offset.
3. A model to predict the percentage of total time spent in a BA state. As this response is limited to 0 and 1, it was set to be a Beta distribution. Again, the variable group is the population-level parameter of interest.
4. A model to predict the total time in seconds spent in a REM sleep. With response distribution set to log-normal, with the variable group as the population-level parameter of interest.
5. A model to predict the total time in seconds spent in a NREM sleep. With response distribution set to log-normal, with the variable group as the population-level parameter of interest.
6. A model to predict the total time in seconds spent awake. With response distribution set to log-normal, with the variable group as the population-level parameter of interest.

```
# Load needed packages.
```

```
library("tidyverse")
library("ggplot2")
library("brms")
library("brmstools")
library("bayesplot")
library("emmeans")
library("tidybayes")
library("here")
library("coda")
```

## Treatment effects on BAs

Models 1-3 are fit on the data generated by manually labelling mouse video recordings.

We read in the data and munge it to have it in the necessary format to fit each model.

## BA duration model

```
# Load misc. functions.
source(here("R", "misc_functions.R"))

# Load recorded behaviors data.
data <- read_csv(here("data", "videos", "recorded_behaviors.csv"))

# Select only BA events (units are seconds).
ba_seconds <- filter(data, behavior=="BA")

# Change variable types.
ba_seconds$group <- as.factor(ba_seconds$group)
ba_seconds$mouse <- as.factor(ba_seconds$mouse)

# Which priors can we specify for our model?
get_prior(seconds ~ group + (1|mouse), data=ba_seconds)

##           prior      class      coef group resp dpar nlpar bound
## 1                    b
## 2                    b groupsham
## 3                    b grouptxcb
## 4                    b grouptxox
## 5 student_t(3, 26, 18) Intercept
## 6 student_t(3, 0, 18)      sd
## 7                    sd          mouse
## 8                    sd Intercept mouse
## 9 student_t(3, 0, 18)      sigma
```

The default settings in the brms package for priors were used in all models. For the fitted models each group-level effect of the grouping factor has a standard deviation. By default ordinary regression coefficients have “improper flat” priors and the rest are restricted to be non-negative and have a half student-t prior (Gelman, 2006) with 3 degrees of freedom and a scale parameter that depends on the standard deviation of the response after applying the link function (Bürkner, 2017). These are equivalent to having no priors and weakly informative priors respectively. For example, for the BA duration model, 9 parameters need to be estimated.

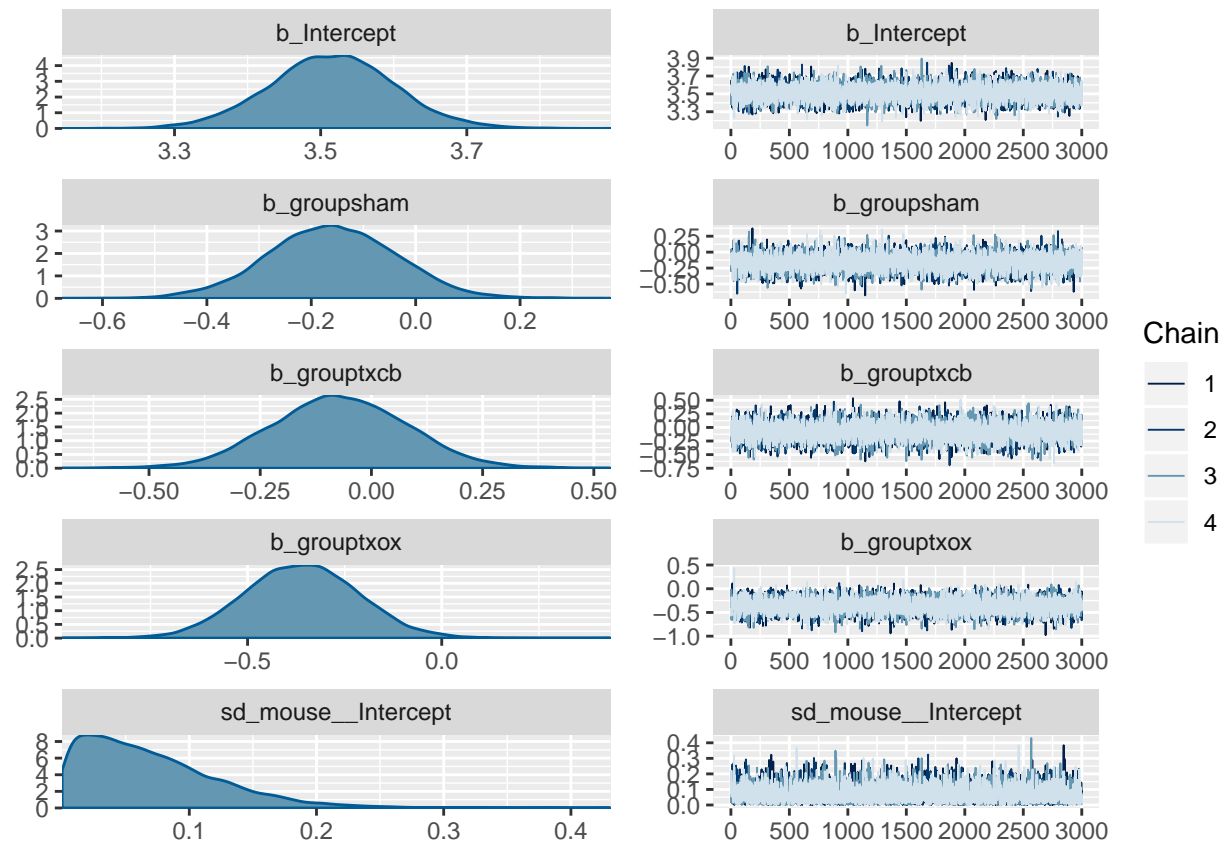
We proceed to fit the model.

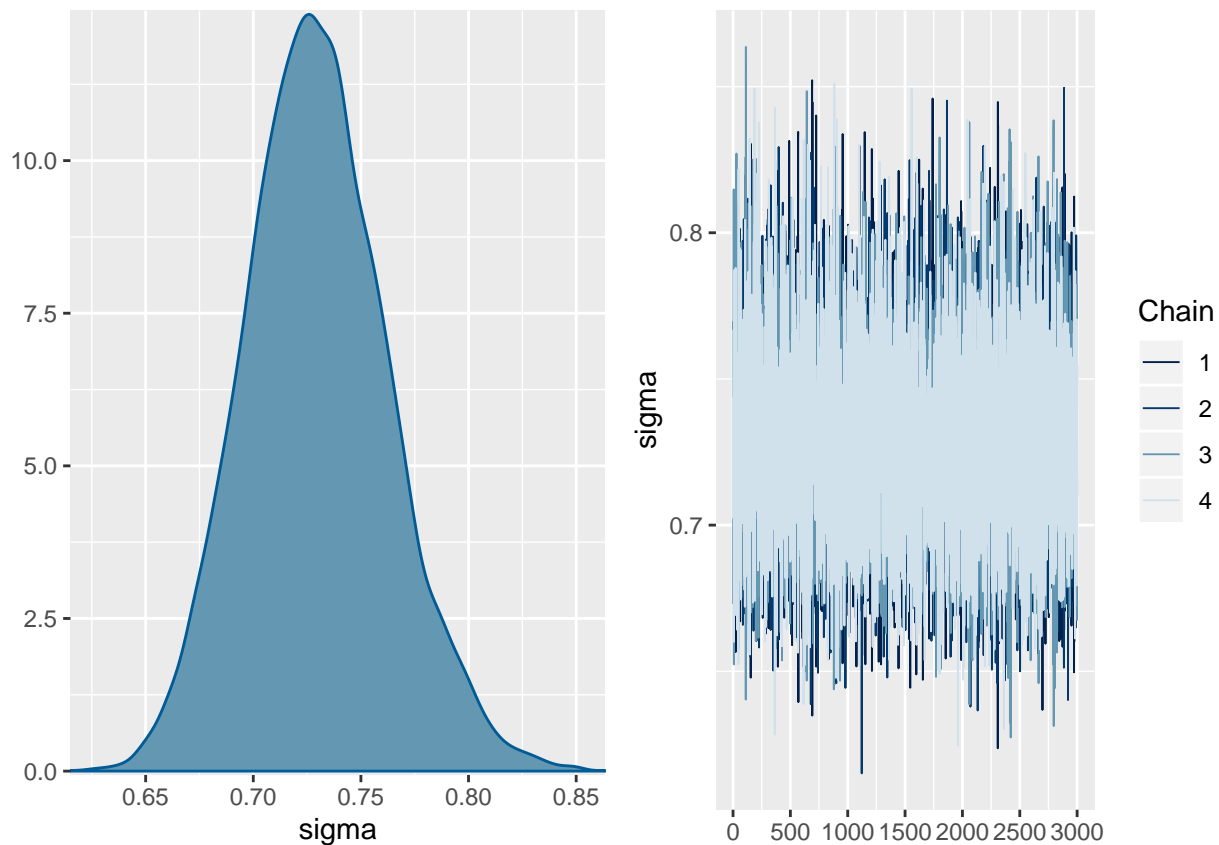
```
# Fit brms model for duration of each BA attack.
ba_seconds_fit <- brm(seconds ~ group + (1|mouse),
  data = ba_seconds,
  family = lognormal,
  warmup = 2000,
  iter = 5000,
  control = list(adapt_delta = 0.999))
```

No runtime warnings during estimation were thrown which is a first indication of proper convergence.

We perform a couple of more checks. First we see if the trace plot exhibits convergence.

```
plot(ba_seconds_fit)
```

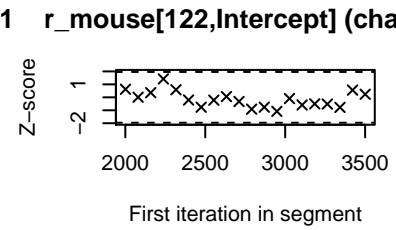
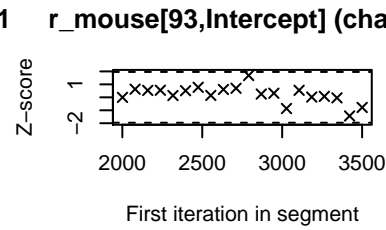
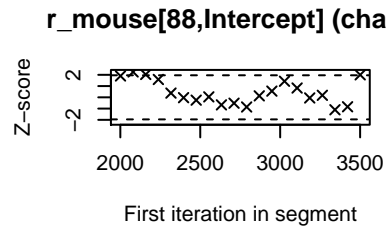
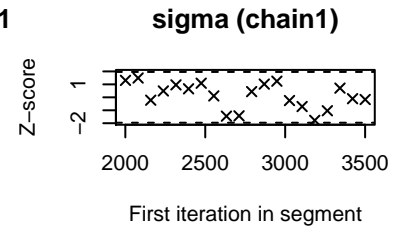
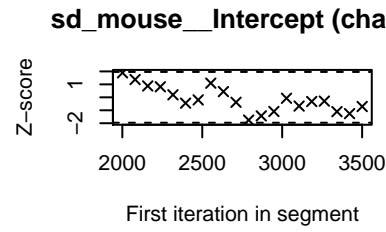
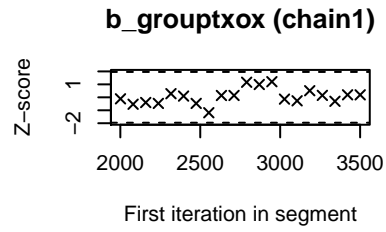
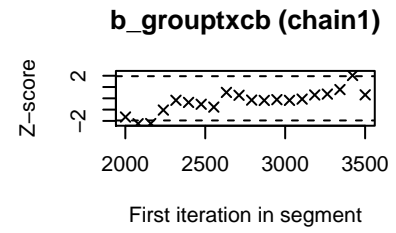
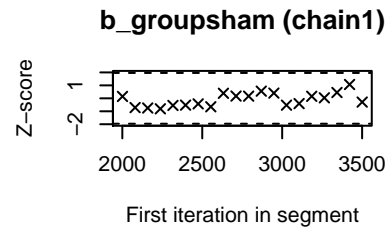
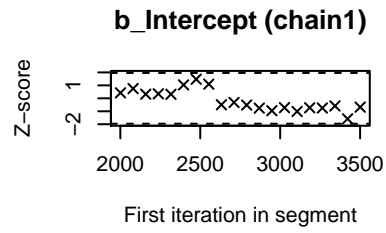


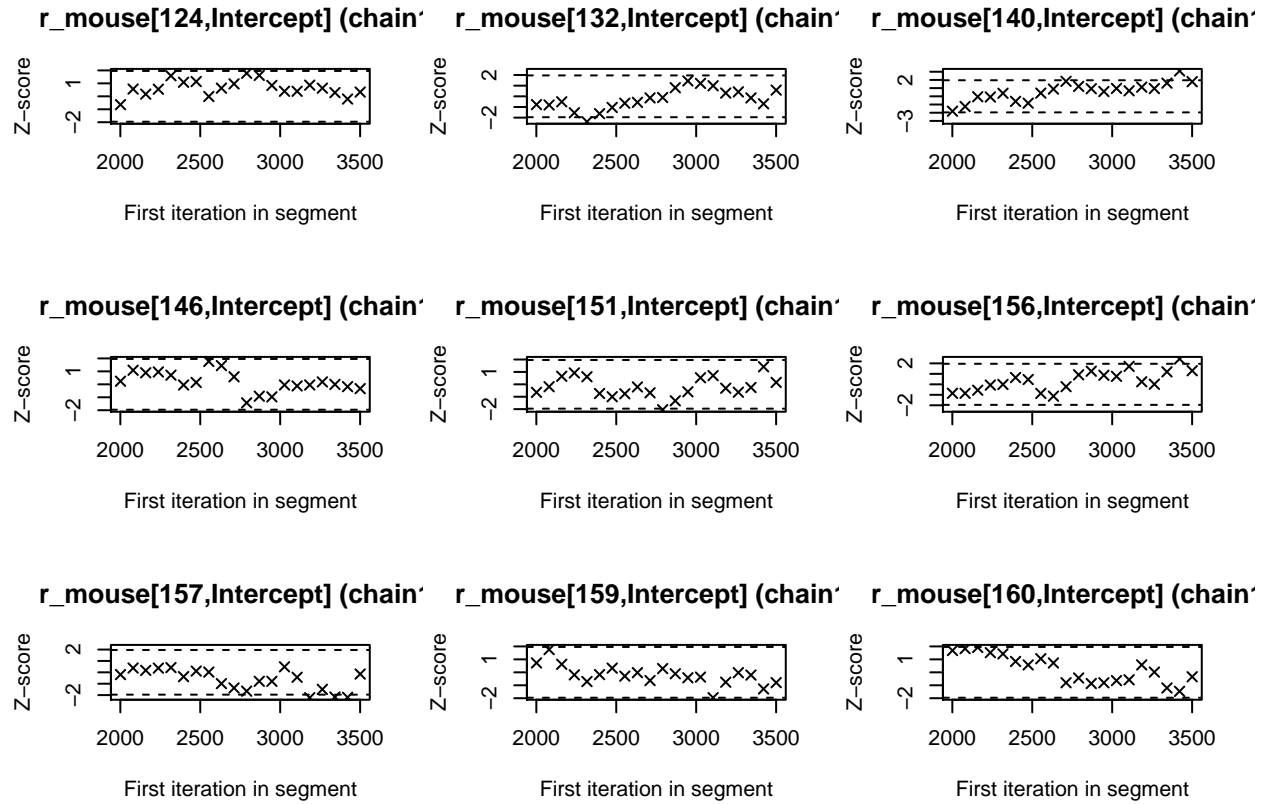


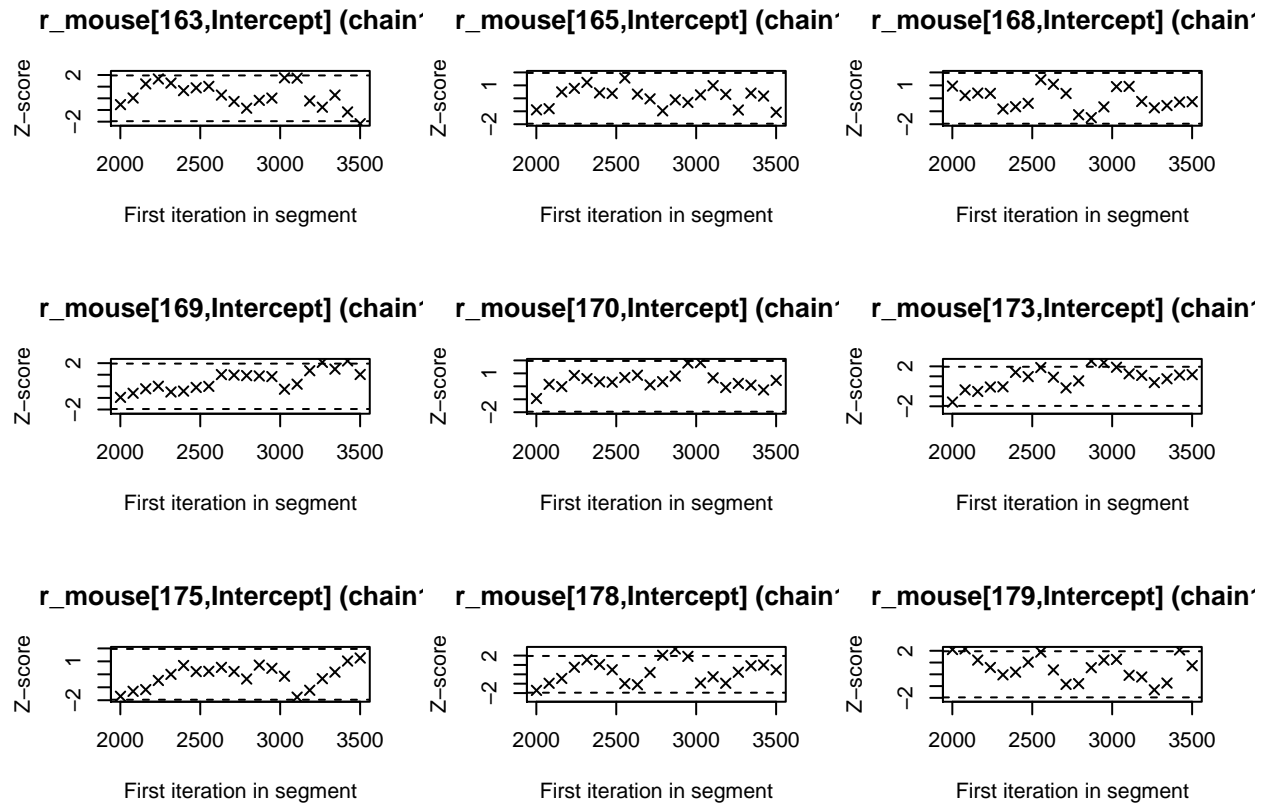
The classical “furry caterpillars” can be seen which is an indication of no divergences.

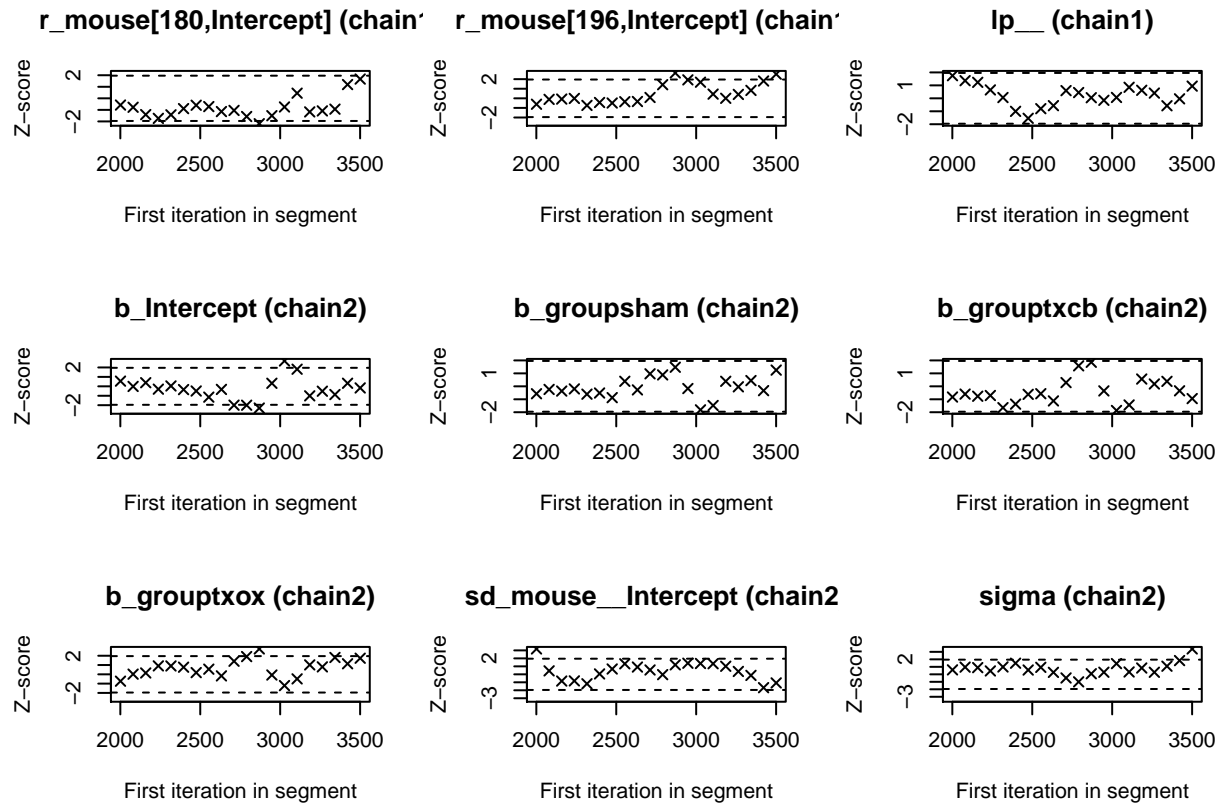
We then check the Geweke Diagnostic which shows the z-scores for a test of equality of means between the first and last parts of each chain, which should be in the interval  $[-1.96, 1.96]$ . Scores above 1.96 or below -1.96 mean that the two portions of the chain significantly differ and full chain convergence was not obtained.

```
modelposterior <- as.mcmc(ba_seconds_fit)
geweke.plot(modelposterior)
```

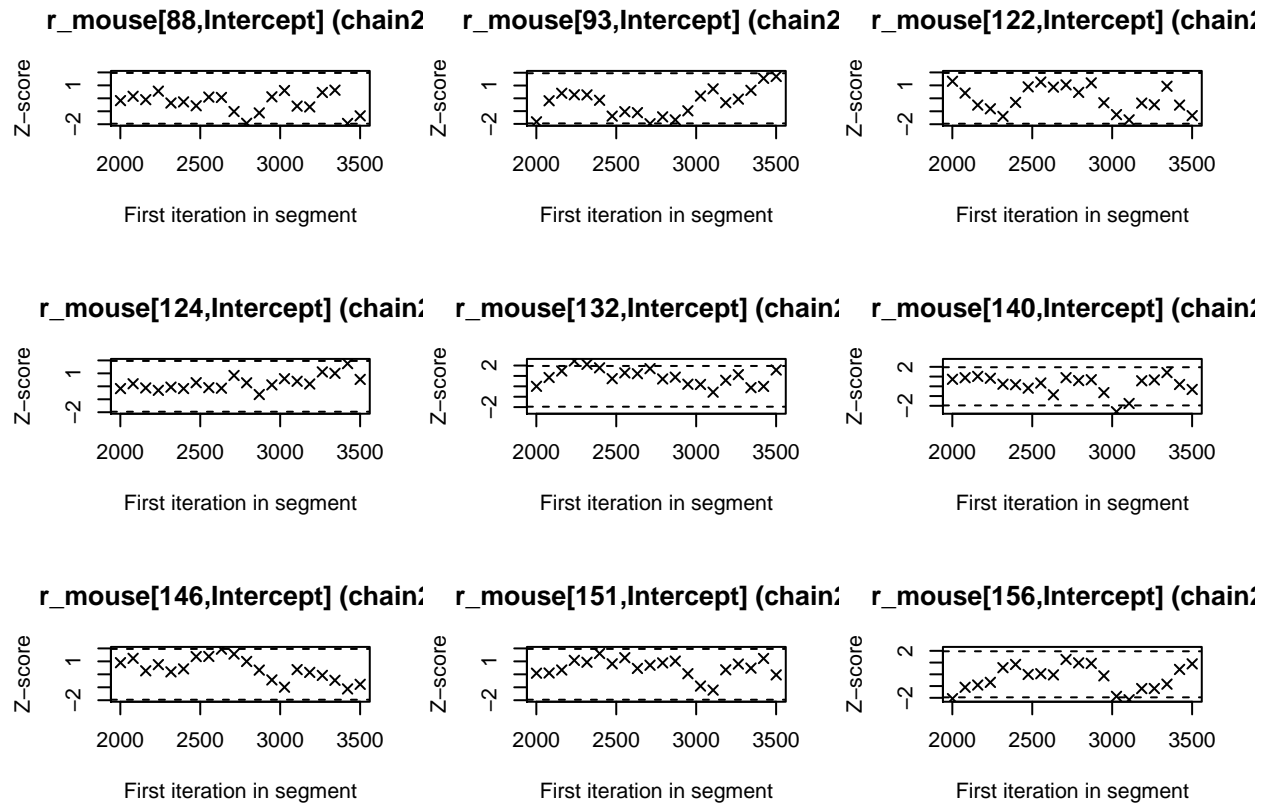




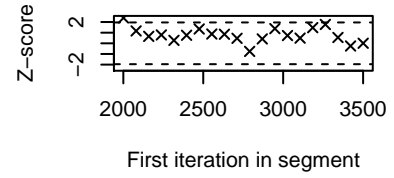
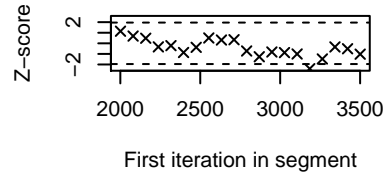
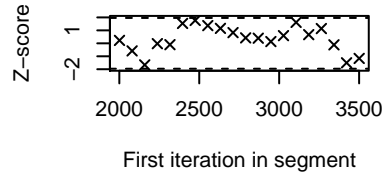




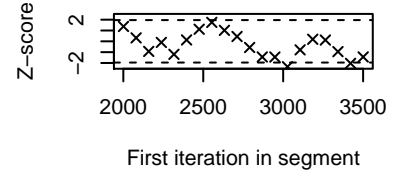
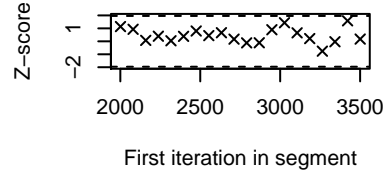
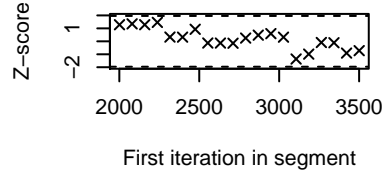




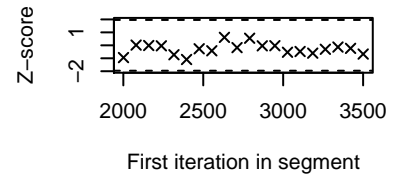
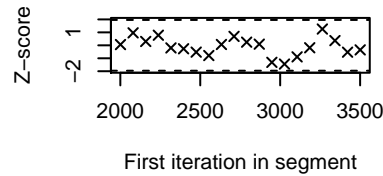
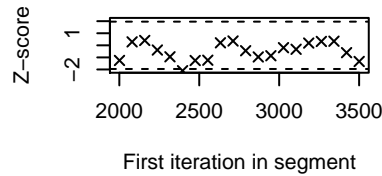
**r\_mouse[157,Intercept] (chain: r\_mouse[159,Intercept] (chain: r\_mouse[160,Intercept] (chain:**



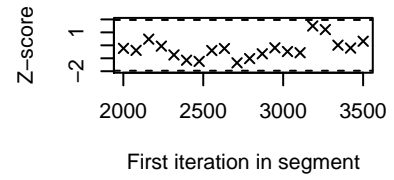
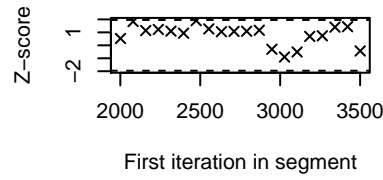
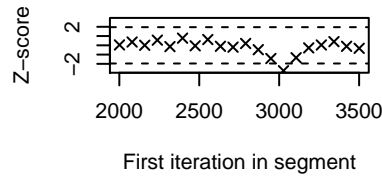
**r\_mouse[163,Intercept] (chain: r\_mouse[165,Intercept] (chain: r\_mouse[168,Intercept] (chain:**



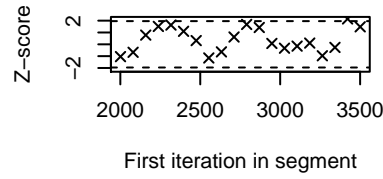
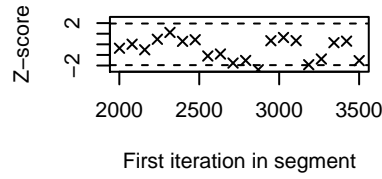
**r\_mouse[169,Intercept] (chain: r\_mouse[170,Intercept] (chain: r\_mouse[173,Intercept] (chain:**



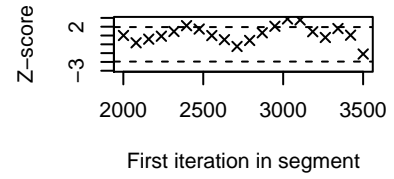
**r\_mouse[175,Intercept] (chain:** **r\_mouse[178,Intercept] (chain:** **r\_mouse[179,Intercept] (chain:**



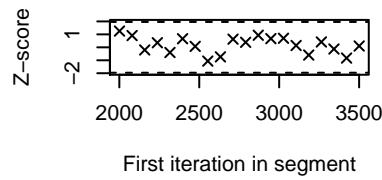
**r\_mouse[180,Intercept] (chain:** **r\_mouse[196,Intercept] (chain:**



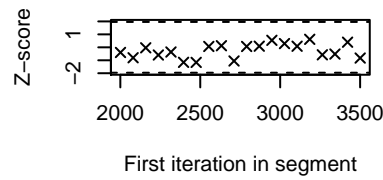
**lp\_\_ (chain2)**



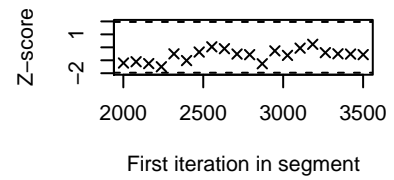
**b\_Intercept (chain3)**

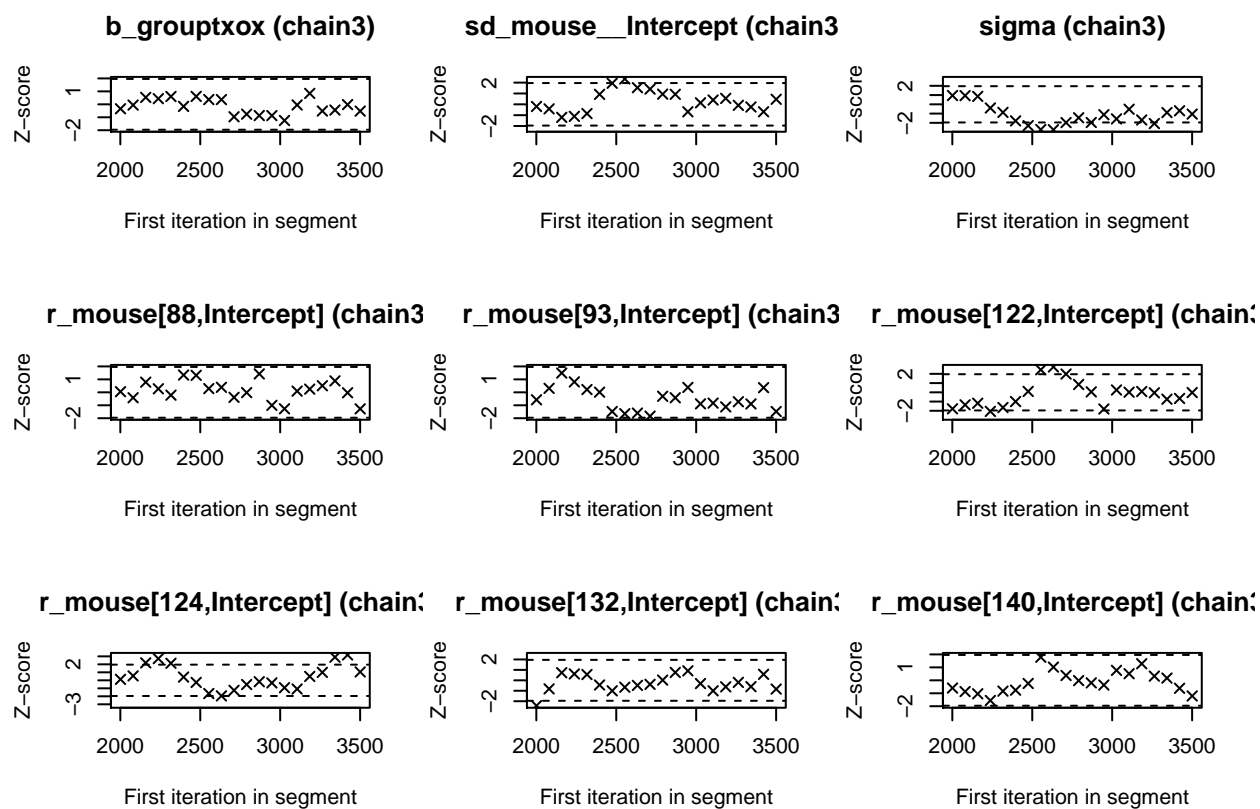


**b\_groupsham (chain3)**

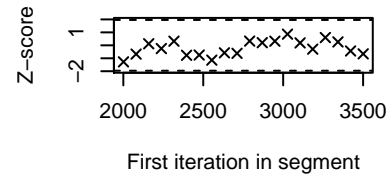
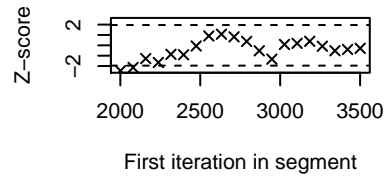
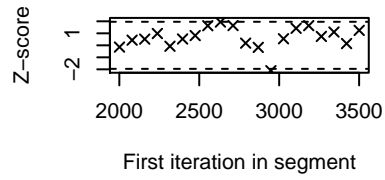


**b\_grouptxcb (chain3)**

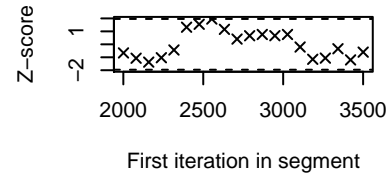
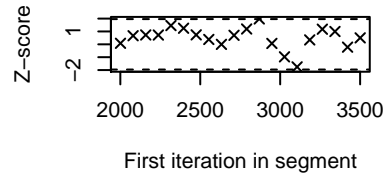
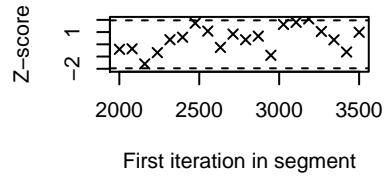




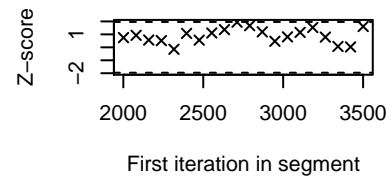
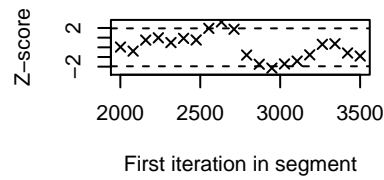
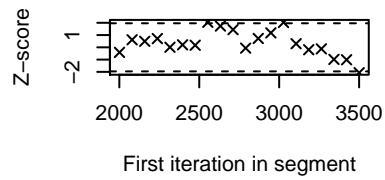
**r\_mouse[146,Intercept] (chain: r\_mouse[151,Intercept] (chain: r\_mouse[156,Intercept] (chain:**



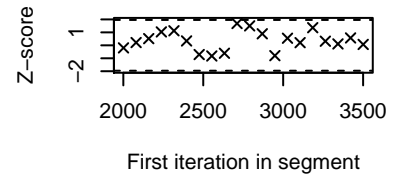
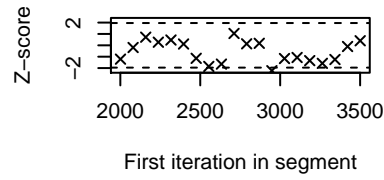
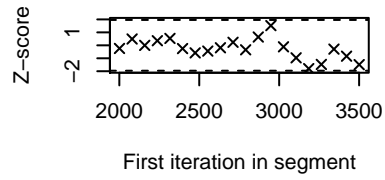
**r\_mouse[157,Intercept] (chain: r\_mouse[159,Intercept] (chain: r\_mouse[160,Intercept] (chain:**



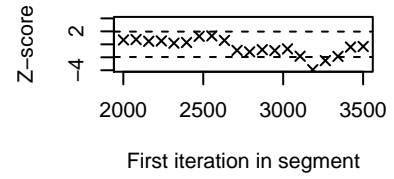
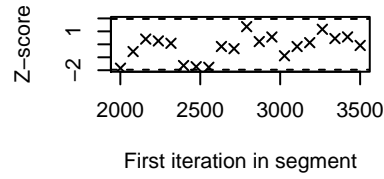
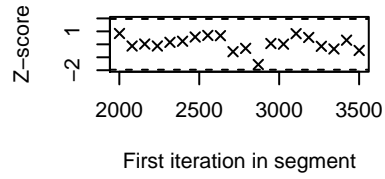
**r\_mouse[163,Intercept] (chain: r\_mouse[165,Intercept] (chain: r\_mouse[168,Intercept] (chain:**



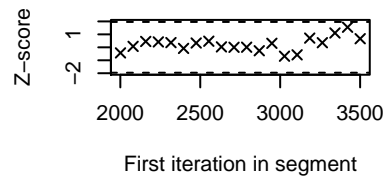
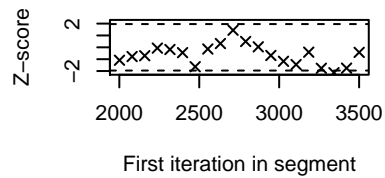
**r\_mouse[169,Intercept] (chain: r\_mouse[170,Intercept] (chain: r\_mouse[173,Intercept] (chain:**



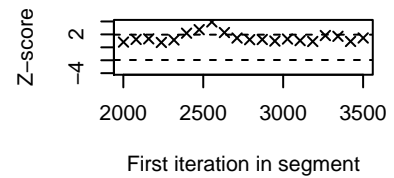
**r\_mouse[175,Intercept] (chain: r\_mouse[178,Intercept] (chain: r\_mouse[179,Intercept] (chain:**

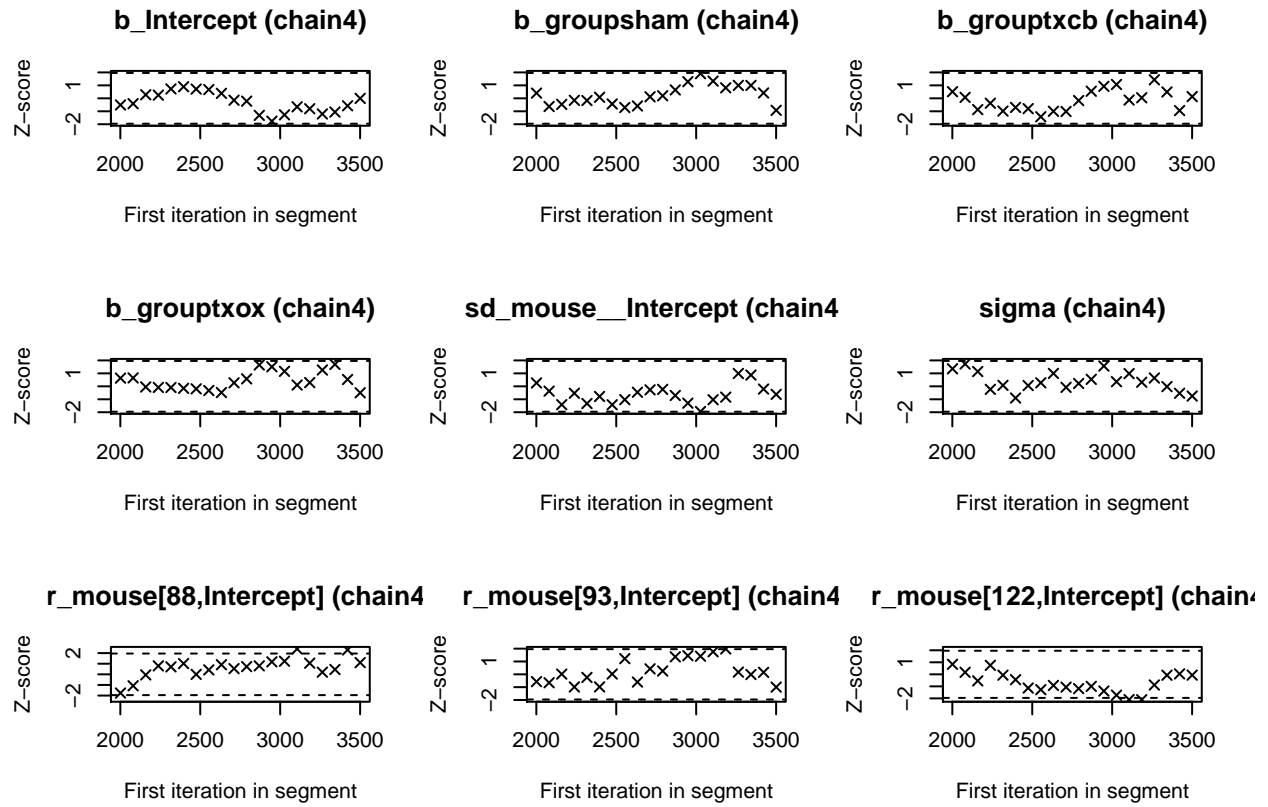


**r\_mouse[180,Intercept] (chain: r\_mouse[196,Intercept] (chain:**

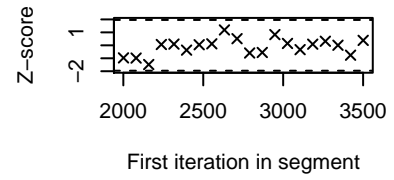
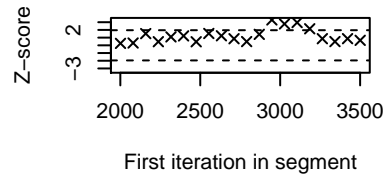
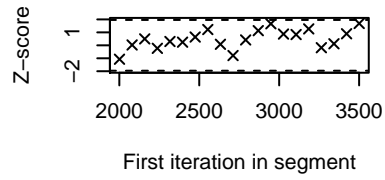


**lp\_\_ (chain3)**

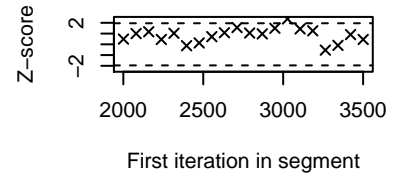
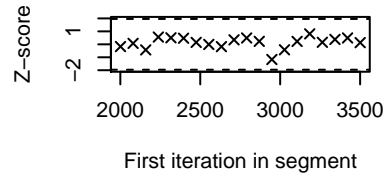
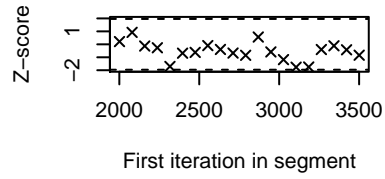




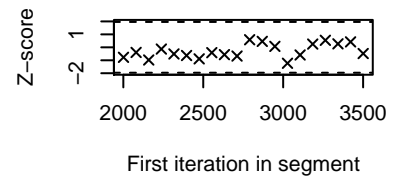
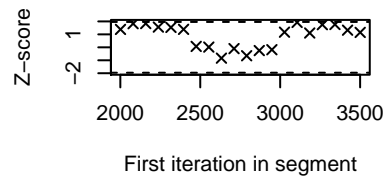
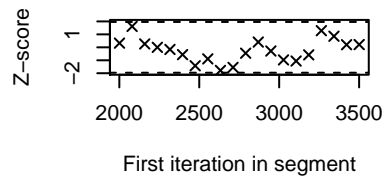
**r\_mouse[124,Intercept] (chain**   **r\_mouse[132,Intercept] (chain**   **r\_mouse[140,Intercept] (chain**



**r\_mouse[146,Intercept] (chain**   **r\_mouse[151,Intercept] (chain**   **r\_mouse[156,Intercept] (chain**

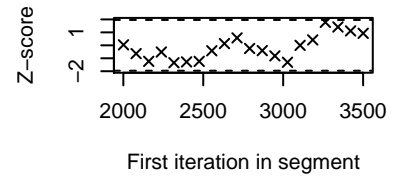
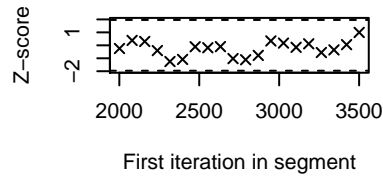
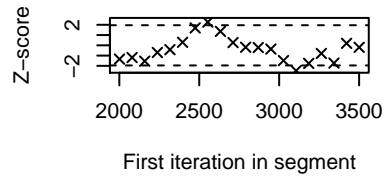


**r\_mouse[157,Intercept] (chain**   **r\_mouse[159,Intercept] (chain**   **r\_mouse[160,Intercept] (chain**

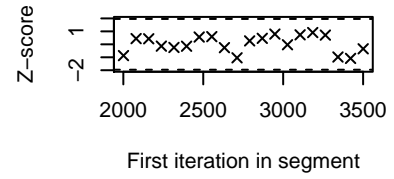
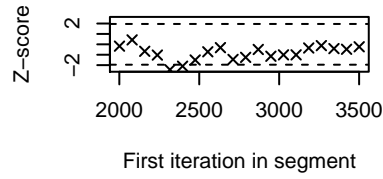
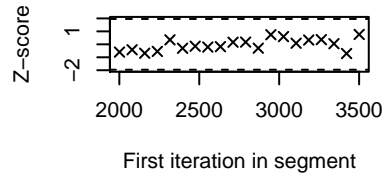




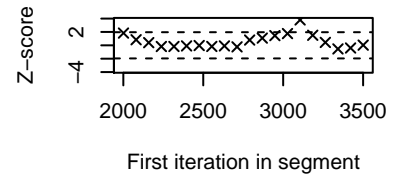
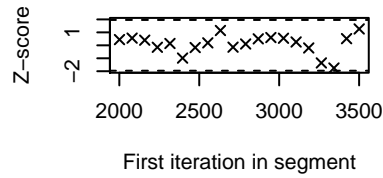
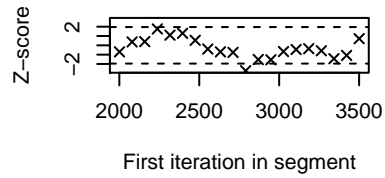
**r\_mouse[163,Intercept] (chain**   **r\_mouse[165,Intercept] (chain**   **r\_mouse[168,Intercept] (chain**

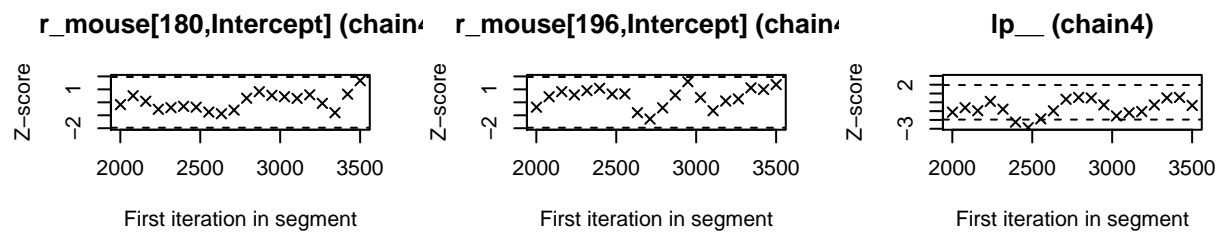


**r\_mouse[169,Intercept] (chain**   **r\_mouse[170,Intercept] (chain**   **r\_mouse[173,Intercept] (chain**



**r\_mouse[175,Intercept] (chain**   **r\_mouse[178,Intercept] (chain**   **r\_mouse[179,Intercept] (chain**

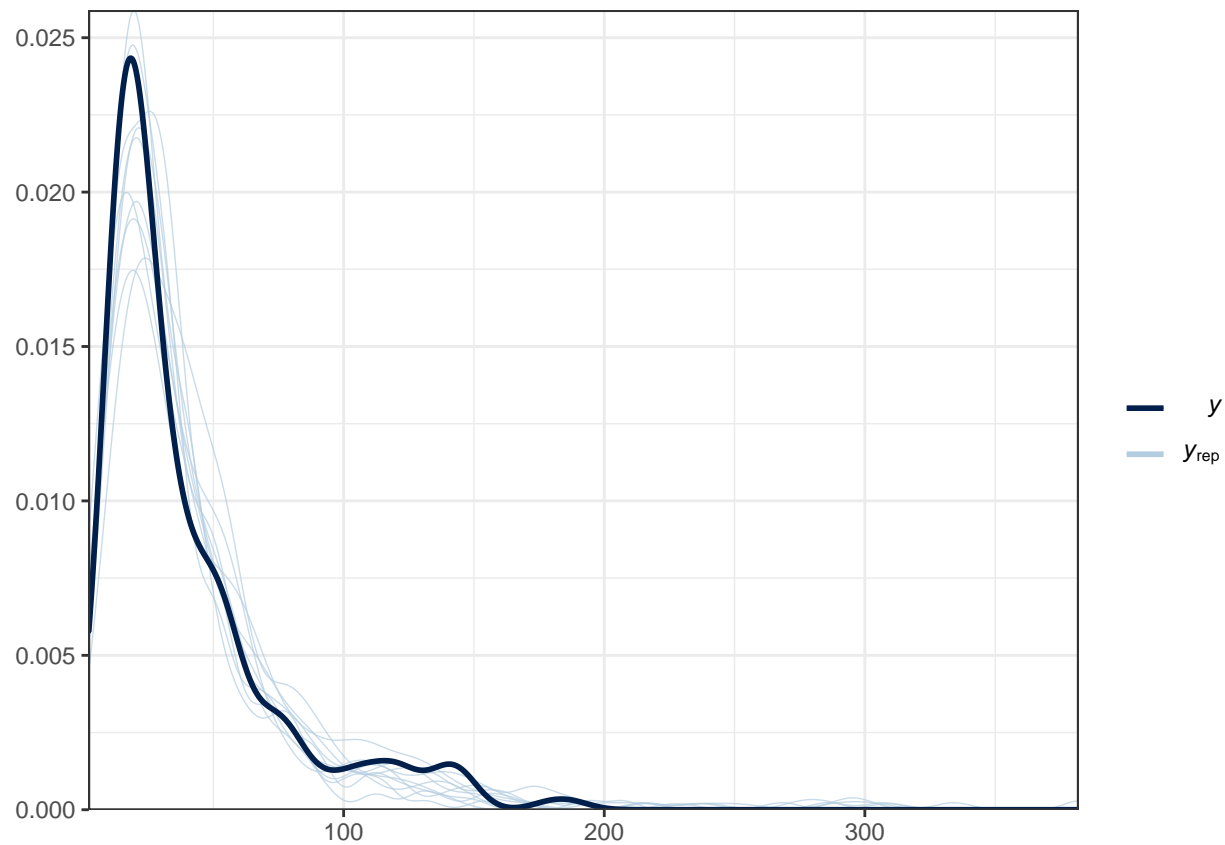




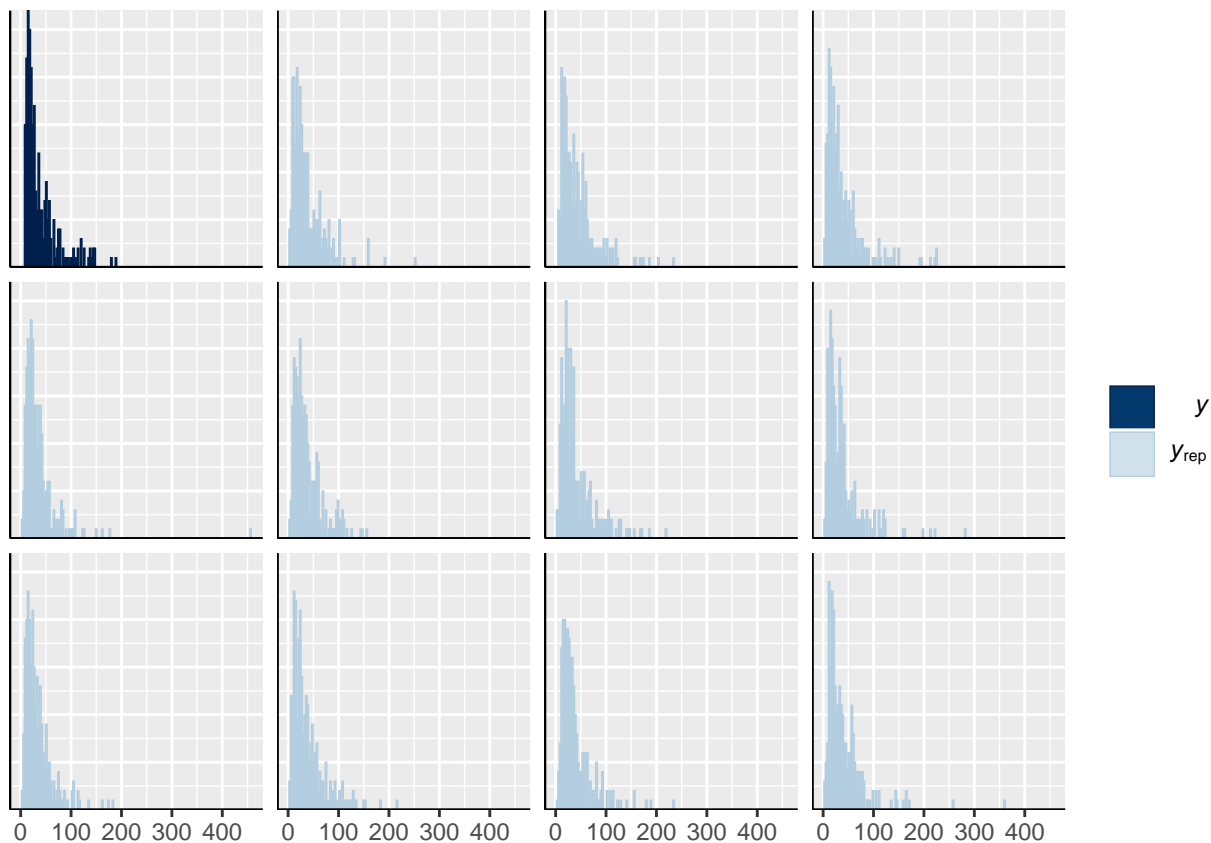
We see that the Geweke diagnostic is acceptably inside the  $[-1.96, 1.96]$  bands.

We will now assess model fit by examining how well the model can reproduce the data at hand by means of (visual) posterior predictive checks.

```
# Predictive posterior checks.
pp <- brms::pp_check(ba_seconds_fit, nsamples=10)
pp + theme_bw()
```



```
y_simulated <- posterior_predict(ba_seconds_fit, draws = 500)  
ppc_hist(ba_seconds$seconds, y_simulated[1:11,], binwidth = 3)
```



Fit seems adequate as density plots of observed and predicted values are similar.

Finally, we interpret our model results.

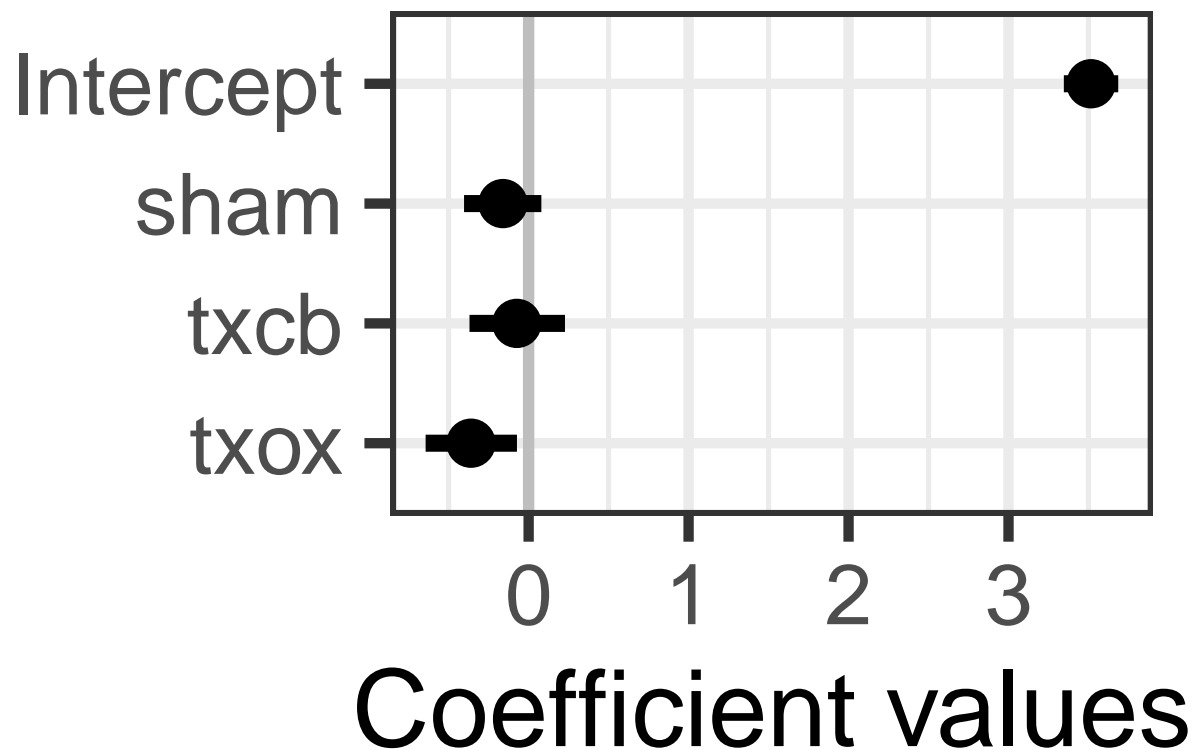
*# Model summary.*

```
summary(ba_seconds_fit)
```

```
## Family: lognormal
## Links: mu = identity; sigma = identity
## Formula: seconds ~ group + (1 | mouse)
## Data: ba_seconds (Number of observations: 258)
## Samples: 4 chains, each with iter = 5000; warmup = 2000; thin = 1;
##           total post-warmup samples = 12000
##
## Group-Level Effects:
## ~mouse (Number of levels: 23)
##           Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)    0.07     0.05    0.00    0.20     5468 1.00
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept         3.52     0.09     3.35     3.69      8082 1.00
## groupsham        -0.16     0.12    -0.40     0.08      8869 1.00
## grouptxcb        -0.07     0.15    -0.37     0.23      9616 1.00
## grouptxox        -0.36     0.15    -0.64    -0.07      9626 1.00
##
```

```
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sigma      0.73      0.03    0.67    0.80    15426 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

# Coefficient plot.
coeff_plot_data = interval_data(ba_seconds_fit)
plot_intervals(coeff_plot_data)
```



The first thing that must be noticed is that grouptxox estimate (effect) is negative as its 95%-CI does not touch 0. This indicates that, on average, the treatment credibly reduces the duration of BA episodes by some amount.

And finally we calculate effect sizes.

```
# Contrasts.
credible_intervals_dat = credible_intervals_data(ba_seconds,ba_seconds_fit)

# Plot contrasts.
plot_contrasts(credible_intervals_dat,xlab="Effect size")
```

