**85182 - Bayesian Inference**
University of Bologna
November - December 2023

# Pratical Exam Simulation - Solutions

**December 6, 2023**

This example comes from Chapter 8.3 of Gelman and Hill (2007).

Suppose that we want to make inferences about the efficacy of a certain pest management system at reducing the number of roaches in urban apartments. Here is how Gelman and Hill describe the experiment (pg. 161):

> [...] the treatment and control were applied to 160 and 104 apartments, respectively, and the outcome measurement $y_i$ in each apartment $i$ was the number of roaches caught in a set of traps. [...]

In addition to an intercept, the regression predictors for the model are the pre-treatment number of roaches `roach1`, the treatment indicator `treatment`, and a variable `senior` indicating whether the apartment is in a building restricted to elderly residents.

Load the data and rescale the variable `roach1` using the following piece of code

```
library(rstanarm)
data(roaches)
# Rescale
roaches$roach1 <- roaches$roach1 / 100
```

## Exercise questions:

1. Assume a simple Poisson regression model without random effects considering `roach1` and `senior` as independent variables (model a). Write the theoretical form of the model assuming weakly-informative prior distributions for the parameters and program the model in `rstanarm`.

$$
\begin{aligned}
y_i|\mu_i &\sim Poisson(\mu_i) \\
\log(\mu_i)|\boldsymbol{\beta} &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} \\
\beta_0 &\sim \mathcal{N}(0, 2.5), \\
\beta_1 &\sim \mathcal{N}(0, 3.32), \\
\beta_2 &\sim \mathcal{N}(0, 5.42)
\end{aligned}
$$

where

- Y = post-treatment number of roaches
- $X_1$ = pre-treatment number of roaches
- $X_2$ = elderly residents indicator

```
mod_a <- stan_glm(formula = y~roach1+senior,
                  data = roaches,
                  family = "poisson")

prior_summary(mod_a)
```

```
> prior_summary(mod_a)
Priors for model 'mod_a'
------
Intercept (after predictors centered)
~ normal(location = 0, scale = 2.5)

Coefficients
Specified prior:
~ normal(location = [0,0,0], scale = [2.5,2.5])
Adjusted prior:
~ normal(location = [0,0], scale = [3.32,5.42])
------
See help('prior_summary.stanreg') for more details
```

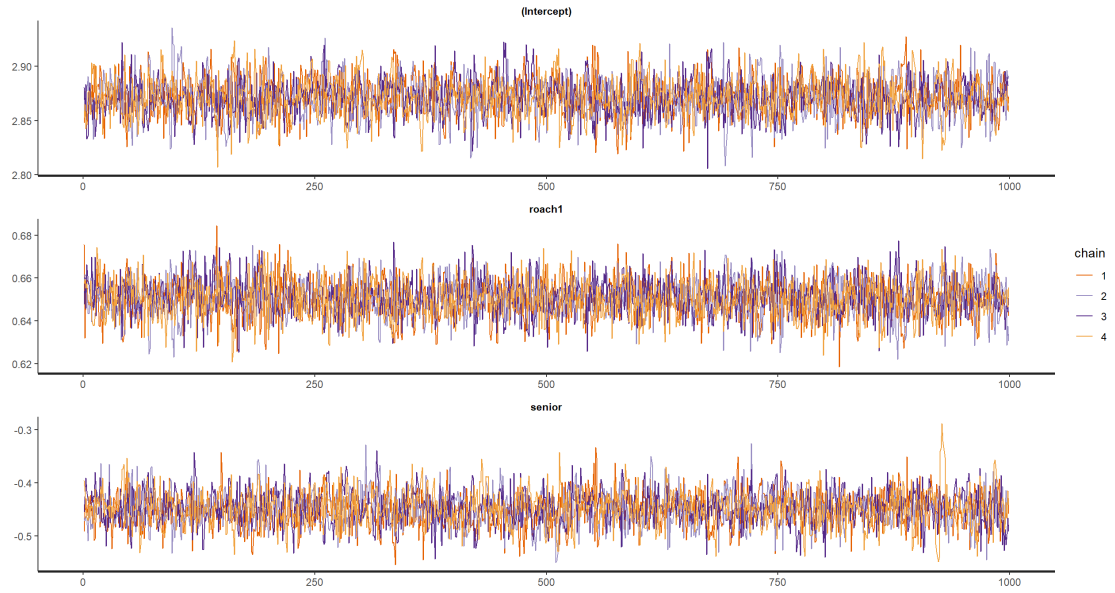
```
summary(mod_a)
```

```
MCMC diagnostics
              mcse  Rhat  n_eff
(Intercept)   0.0   1.0   2312
roach1        0.0   1.0   3208
senior        0.0   1.0   2500
mean_PPD      0.0   1.0   3338
```

No need to increase the number of iterations.

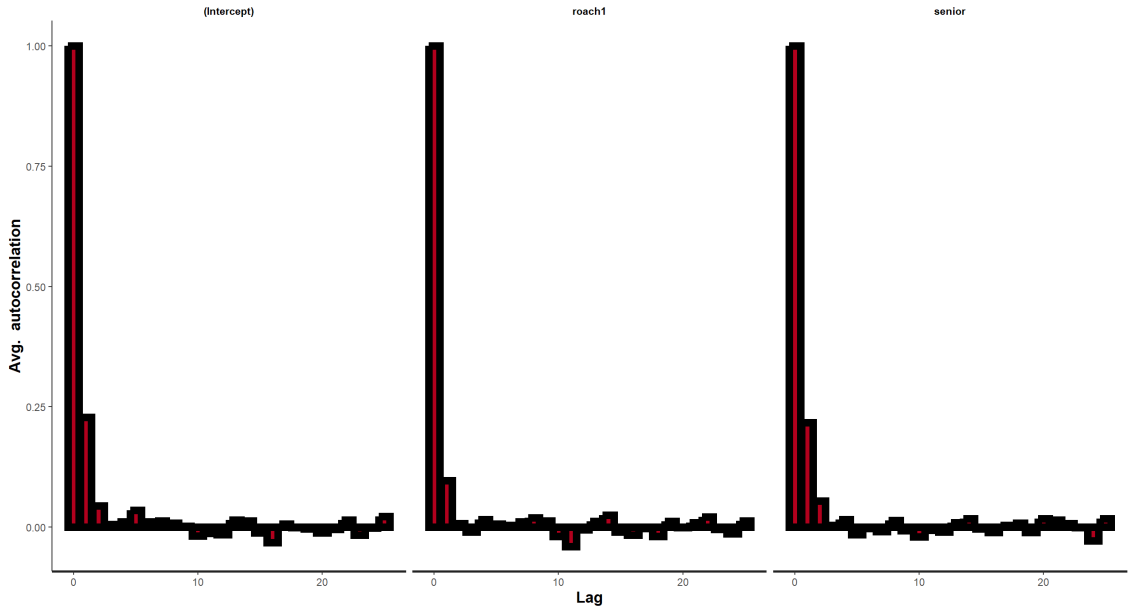
2. Monitor the convergence of the algorithm (referred to model a).

```
stan_trace(mod_a, nrow=4,ncol=1)
```




The chains are acceptable and it seems that the 4 chains reached the same target distribution.

```
stan_ac(mod_a)
```




The decay is quite fast and, even if it is not a proper Markov chain,it is acceptable to carry out the posterior analysis

```
summary(mod_a)

MCMC diagnostics
            mcse Rhat n_eff
(Intercept)  0.0  1.0  2312
roach1       0.0  1.0  3208
```

```
senior          0.0   1.0   2500
mean_PPD        0.0   1.0   3338
```

Values of `Rhat` near to 1 point out the convergence. `n_eff` is a first indicator of autocorrelation of the chains. If the value is too far from the sample size (in this case 4000), then the draws might be too correlated among them.

3. Assume now that we are interested in update `model a` in order to estimate also group-specific effects with respect to the type of treatment (model b). Write the theoretical form of the model assuming a $\mathcal{N}(0,10)$ prior distribution with automatic scale adjustments for the parameters and program the model in `rstanarm`.

**Likelihood:**

$$y_{ij}|\mu_{ij} \sim Poisson(\mu_{ij})$$
$$\log(\mu_{ij})|\boldsymbol{\beta}, \lambda_j = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \lambda_j$$

**Priors:**

$$\lambda_j|\sigma_\lambda^2 \sim \mathcal{N}(0, \sigma_\lambda^2),\ j = 0, 1;$$
$$\beta_0 \sim \mathcal{N}(0, 10),$$
$$\beta_1 \sim \mathcal{N}(0, 13.29),$$
$$\beta_2 \sim \mathcal{N}(0, 21.67)$$

**Hyperprior:**

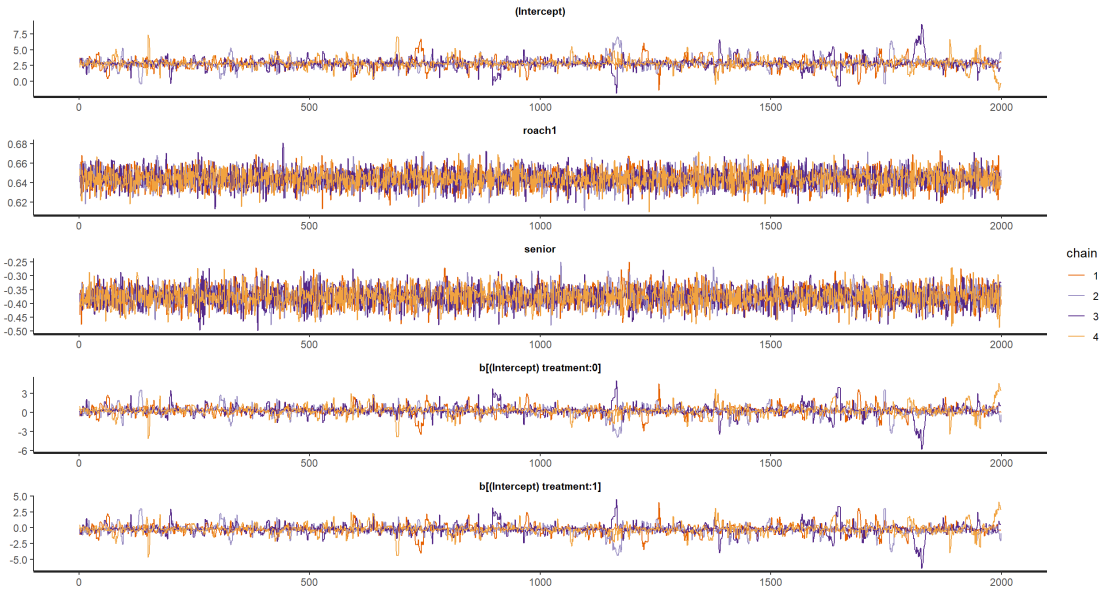
$$\sigma_\lambda \sim \pi(\sigma_\lambda).$$

```
mod_b <- stan_glmer(formula = y~roach1+senior+(1|treatment),
                    data = roaches,
                    family = "poisson",
                    prior = normal(0,10, autoscale=T),
                    prior_intercept = normal(0,10, autoscale=T))

------------------------------------------------------------
Warning messages:
1: There were 6 divergent transitions after warmup.
2: Examine the pairs() plot to diagnose sampling problems
------------------------------------------------------------

summary(mod_b)

MCMC diagnostics
                                          mcse  Rhat  n_eff
(Intercept)                               0.0   1.0    867
roach1                                    0.0   1.0   4669
senior                                    0.0   1.0   4037
b[(Intercept) treatment:0]                0.0   1.0    864
```

```
b[(Intercept) treatment:1]                  0.0   1.0    865
Sigma[treatment:(Intercept),(Intercept)] 0.0   1.0   1226
mean_PPD                                    0.0   1.0   4059
log-posterior                               0.1   1.0   1229

mod_b <- update(mod_b, iter=4000, adapt_delta=.99)

summary(mod_b)

MCMC diagnostics
                                            mcse  Rhat  n_eff
(Intercept)                                 0.0   1.0    985
roach1                                      0.0   1.0   9160
senior                                      0.0   1.0   6795
b[(Intercept) treatment:0]                  0.0   1.0    985
b[(Intercept) treatment:1]                  0.0   1.0    985
Sigma[treatment:(Intercept),(Intercept)] 0.1   1.0   1551
mean_PPD                                    0.0   1.0   7014
```

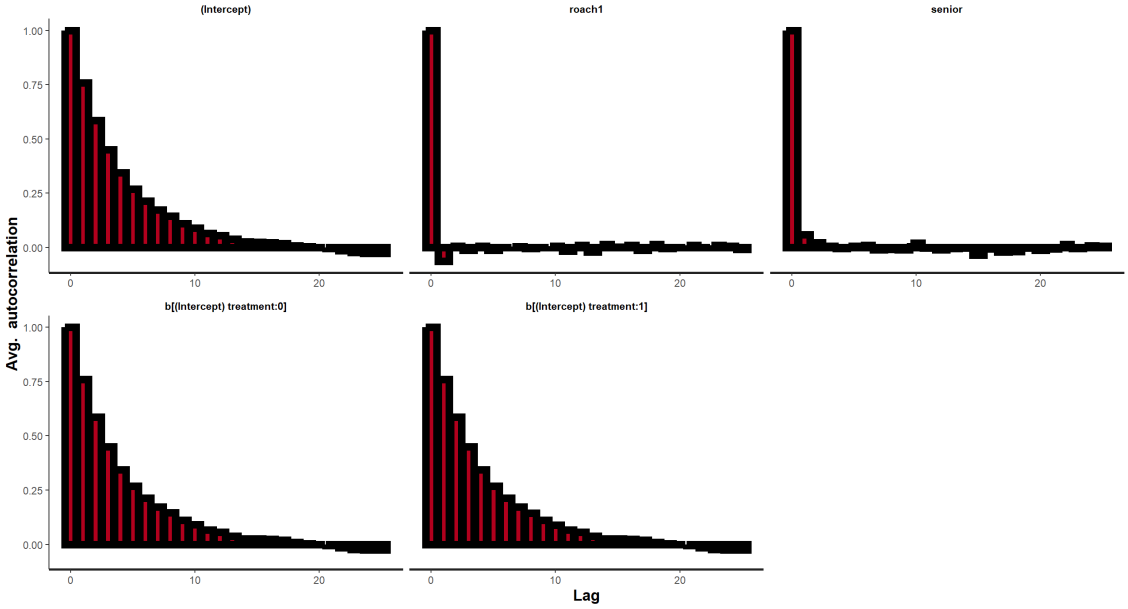
4. Monitor the convergence of the algorithm (referred to model b) and in particular discuss the interpretation of `n_eff`.

```
stan_trace(mod_b, nrow=5,ncol=1)
```




There may be some problems in the chains of the treatment random effects.

```
stan_ac(mod_b)
```




Some evident problems of autocorrelation.

```
summary(mod_b)

MCMC diagnostics
                                                mcse  Rhat  n_eff
(Intercept)                                      0.0   1.0    985
roach1                                           0.0   1.0   9160
senior                                           0.0   1.0   6795
b[(Intercept) treatment:0]                       0.0   1.0    985
b[(Intercept) treatment:1]                       0.0   1.0    985
Sigma[treatment:(Intercept),(Intercept)]         0.1   1.0   1551
mean_PPD                                         0.0   1.0   7014
```

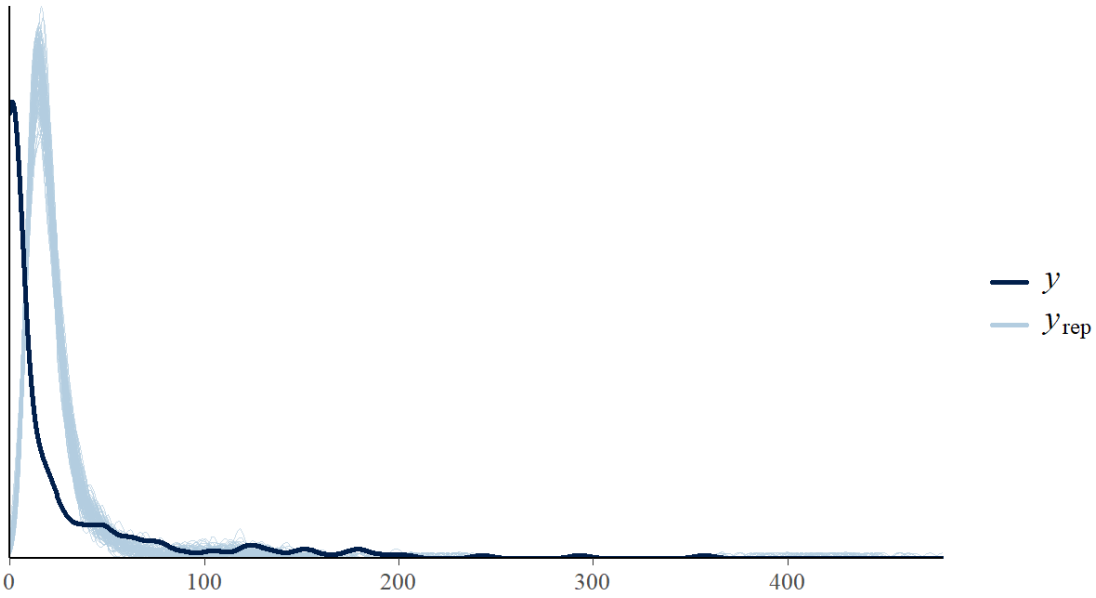
5. Compare the two models in terms of goodness of fit. Which one provides a better fit of the original data?

```
> waic(mod_a)

Computed from 4000 by 262 log-likelihood matrix

Estimate      SE
elpd_waic   -6456.3  810.5
p_waic        248.1   69.8
waic        12912.6 1621.1

> waic(mod_b)

Computed from 8000 by 262 log-likelihood matrix

Estimate      SE
elpd_waic   -6315.1  756.8
```

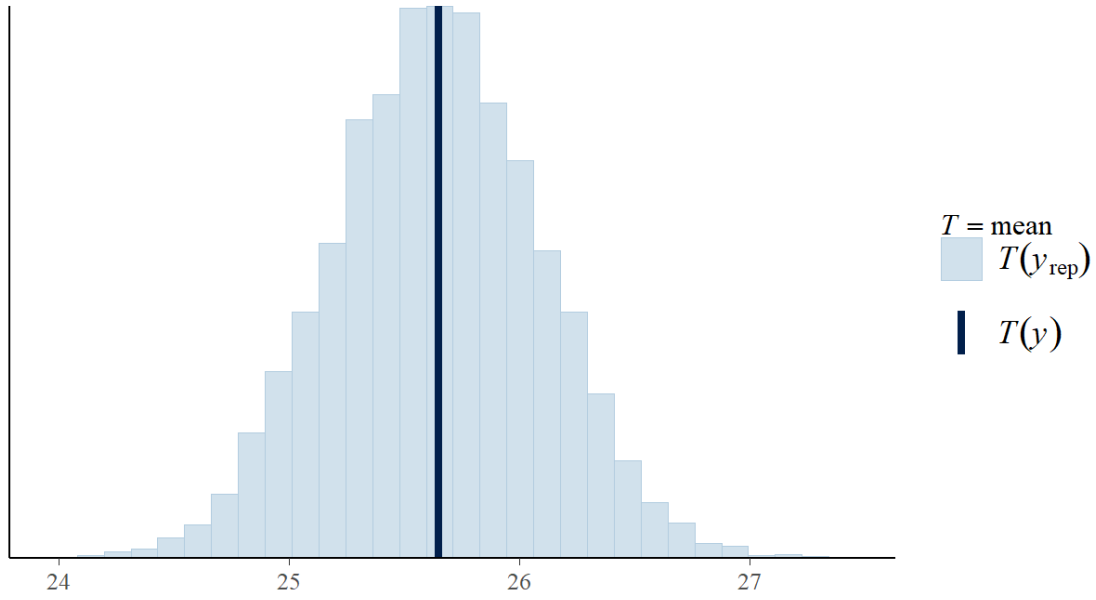
```
p_waic        348.6   114.0
waic        12630.1  1513.6
```

6. Suppose that we are interested in verifying that the chosen model is able to reproduce the mean number of post-treatment roaches. Check this assumption. How do the posterior predictive checks work? Provide a brief explanation.

```
y_tildeb<-posterior_predict(mod_b)
ppc_dens_overlay(y = roaches$y, yrep = y_tildeb[1100:1200,])
```




```
ppc_stat(y = roaches$y, yrep = y_tildeb, stat = "mean")
```




7. Report the 90% credible interval of the predicted parameters.

```
mu <- posterior_epred(mod_b)
quantile(mu, probs = c(0.05,0.5,0.95))

> quantile(mu, probs = c(0.05,0.5,0.95))
5%        50%        95%
9.514181 16.471064 58.663156
```

8. Provide an overall comment of the performed analysis. How could you improve, if necessary, the fitted models in order to better fit the initial dataset?

   We choose `mod_b` according to the value of WAIC. Is it really a "good" model? Should we try to improve it? Was `model_a` preferable somehow?