`rstanarm` - Exercise 1

Bayesian Inference - Lab Sessions

Marika D'Agostini
marika.dagostini2@unibo.it

University of Bologna

November-December 2023

Exercise 1: Normal Linear Regression

- Dataset of size $n = 25$ vending machines
- Response variable: *recharge time*
- Explanatory variables: *product amount* and *distance covered by the operator*.
- The Normal Linear Regression Model is assumed

1) Write the theoretical form of the regression model.
2) Program the model using the default priors of `rstanarm`.
3) Program a second model specifying a flat prior for the model coefficients.
4) Program a third model specifying a $\mathcal{N}(0, 100)$ prior for the regression coefficients and the intercept, and a $Cauchy(0, 1)$ for σ.
5) Referring to the the model specified in step 3), monitor the convergence of the algorithm. In particular, discuss the interpretation of $\hat{R}$.
6) Generate from the posterior predictive distribution and implement the posterior predictive checks.
7) Carry out posterior inference and report the 90% credibility interval of the posterior distribution of observation number 10. In particular, monitor the statistic R^2_B (Bayesian version of R^2):

$$R^2_B = 1 - \frac{\sigma^2}{s^2(y)}.$$

1) Write the theoretical form of the model

$\rightarrow$ The Normal Linear Regression Model is assumed and it has the following **likelihood**:

$$y_i|\mu_i, \sigma^2 \sim \mathcal{N}\left(\mu_i, \sigma^2\right),$$

$$\mu_i|\boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}, \quad i = 1, \ldots, 25.$$

where

$\mathbf{Y}$ = recharge time
$\mathbf{X}_1$ = product amount
$\mathbf{X}_2$ = distance covered by the operator

2) Model with default priors

```
library(rstanarm)
library(rstan)

data1 <- read.csv("Data_Ex_1.csv")

mod_ex1 <- stan_glm(formula = time~amount+distance,
                    data = data1,
                    family = "gaussian")
```

With very few exceptions, the **default priors** in `rstanarm` are **not flat priors**. Rather, the **defaults are intended to be weakly informative**. That is, they are designed to provide moderate regularization and help stabilize computation.

For many (if not most) applications the defaults will perform well, but this is not guaranteed (no default priors make sense for every possible model specification).

The way `rstanarm` attempts to make priors weakly informative by default is to internally adjust the scales of the priors.

```
prior_summary(mod_ex1)
```

```
Priors for model 'mod_ex1'
------
Intercept (after predictors centered)
  Specified prior:
    ~ normal(location = 22, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 22, scale = 39)

Coefficients
  Specified prior:
    ~ normal(location = [0,0], scale = [2.5,2.5])
  Adjusted prior:
    ~ normal(location = [0,0], scale = [5.64,0.12])

Auxiliary (sigma)
  Specified prior:
    ~ exponential(rate = 1)
  Adjusted prior:
    ~ exponential(rate = 0.064)
------
See help('prior_summary.stanreg') for more details
```

3) Model with a flat prior

```
mod_ex1 <- stan_glm(formula = time~amount+distance,
                    data = data1,
                    family = "gaussian",
                    prior = NULL)
```

When "**non-informative**" or "uninformative" is used in the context of prior distributions, it typically refers to a **flat (uniform) distribution** or a nearly flat distribution. Sometimes it may also be used to refer to the parameterization-invariant Jeffreys priors.

Unless the data is very strong it is wise to avoid them.

`rstanarm` will use flat priors if NULL is specified rather than a distribution.

`prior_summary(mod_ex1)`

```
> prior_summary(mod_ex1)
Priors for model 'mod_ex1'
------
Intercept (after predictors centered)
  Specified prior:
    ~ normal(location = 22, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 22, scale = 39)

Coefficients
 ~ flat

Auxiliary (sigma)
  Specified prior:
    ~ exponential(rate = 1)
  Adjusted prior:
    ~ exponential(rate = 0.064)
------
See help('prior_summary.stanreg') for more details
```

4) Model with informative priors

```
mod_ex1 <- stan_glm(formula = time~amount+distance,
                    data = data1,
                    family = "gaussian",
                    prior = normal(0,100),
                    prior_intercept = normal(0,100),
                    prior_aux = cauchy(0,1))
```

Likelihood:

$$y_i|\mu_i, \sigma^2 \sim \mathcal{N}\left(\mu_i, \sigma^2\right),$$

$$\mu_i|\boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}, \quad i = 1, \ldots, 25.$$

Priors:

$$\beta_k \sim \mathcal{N}(0, 100) \qquad k = 1, 2$$

$$\beta_0 \sim \mathcal{N}(0, 100)$$

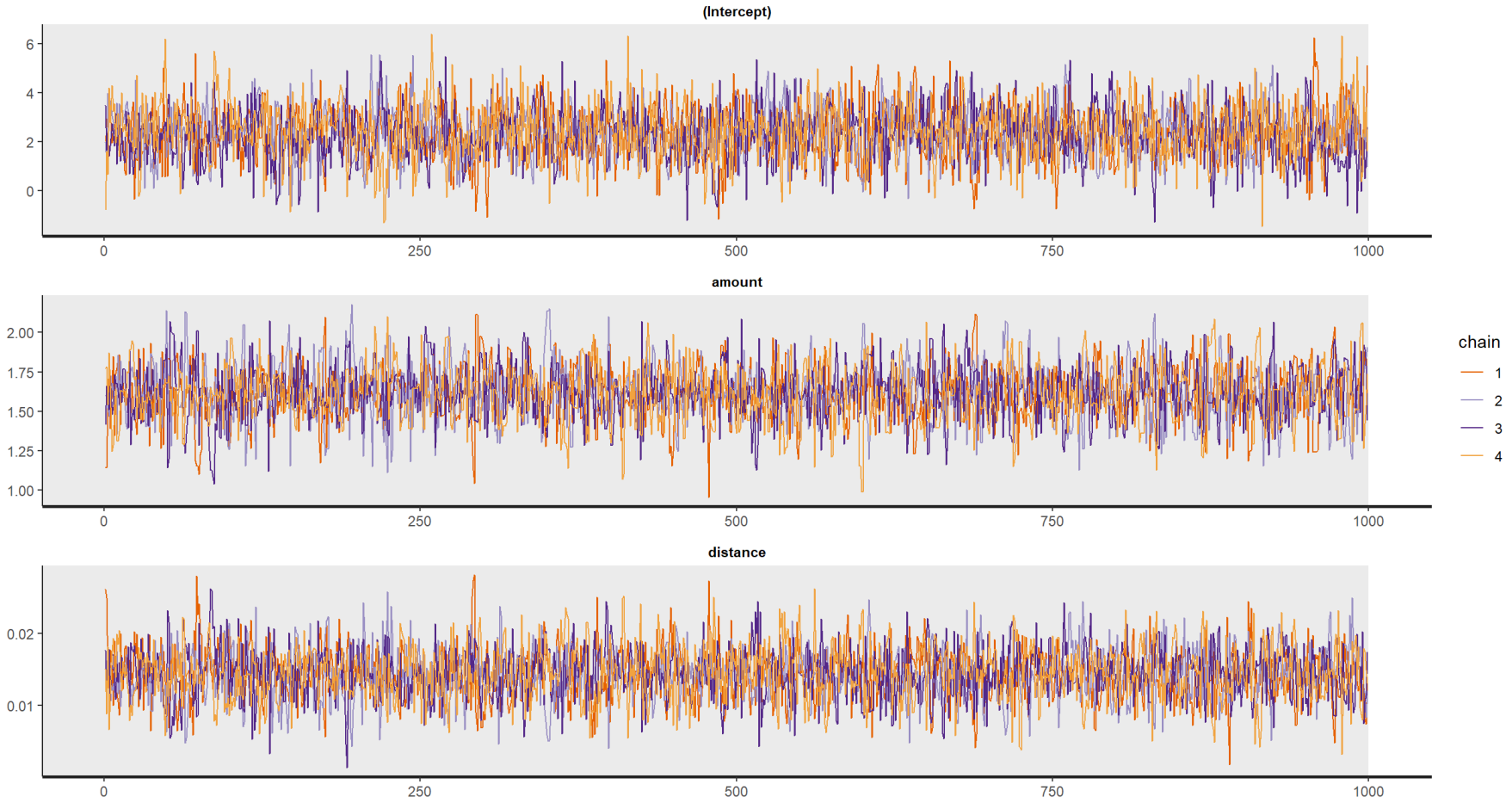
$$\sigma \sim Cauchy(0, 1)$$

```
prior_summary(mod_ex1)
```

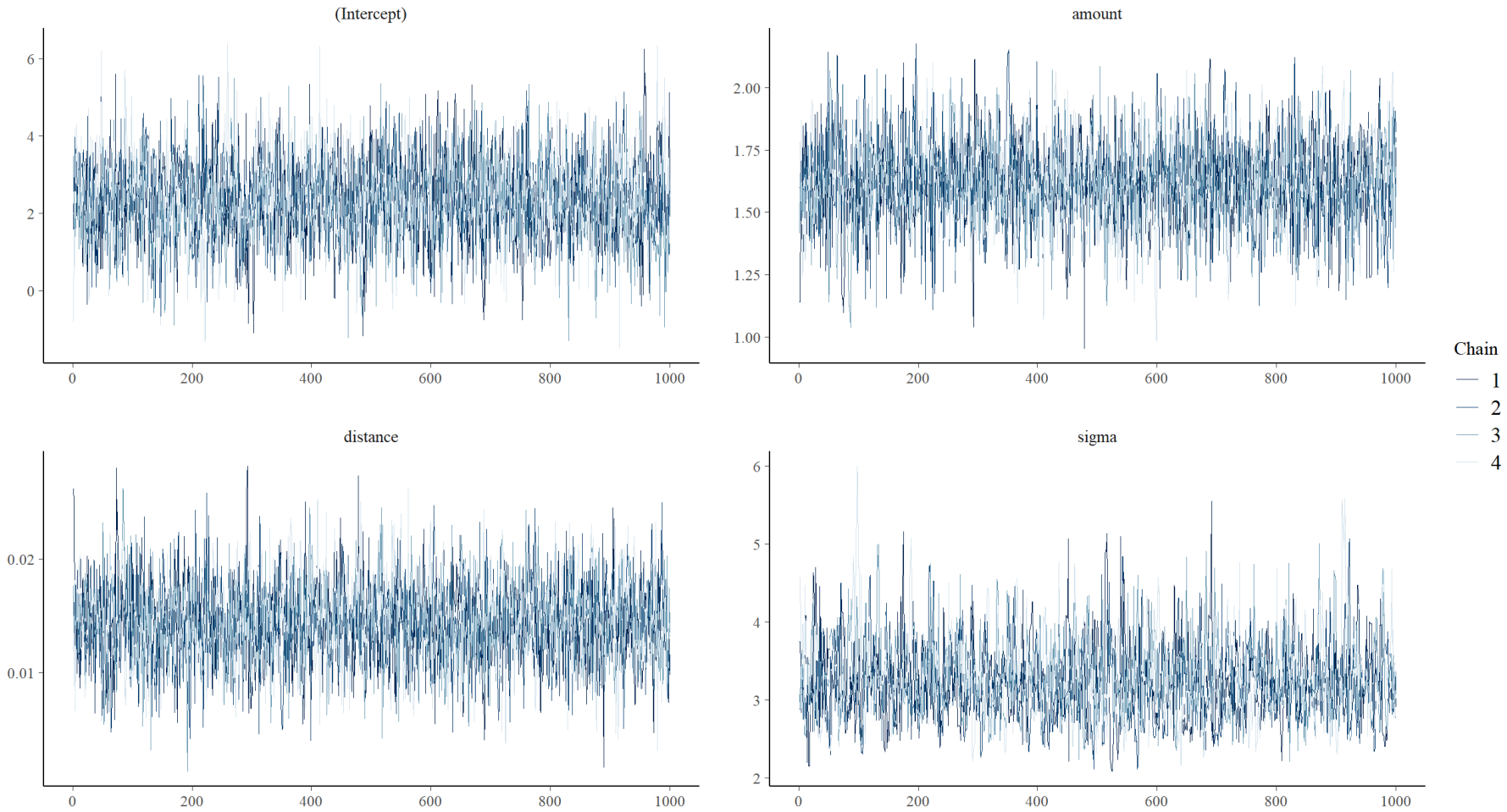
```
Priors for model 'mod_ex1'
------
Intercept (after predictors centered)
 ~ normal(location = 0, scale = 100)

Coefficients
 ~ normal(location = [0,0], scale = [100,100])

Auxiliary (sigma)
 ~ half-cauchy(location = 0, scale = 1)
------
See help('prior_summary.stanreg') for more details
```

5) Convergence of the algorithm

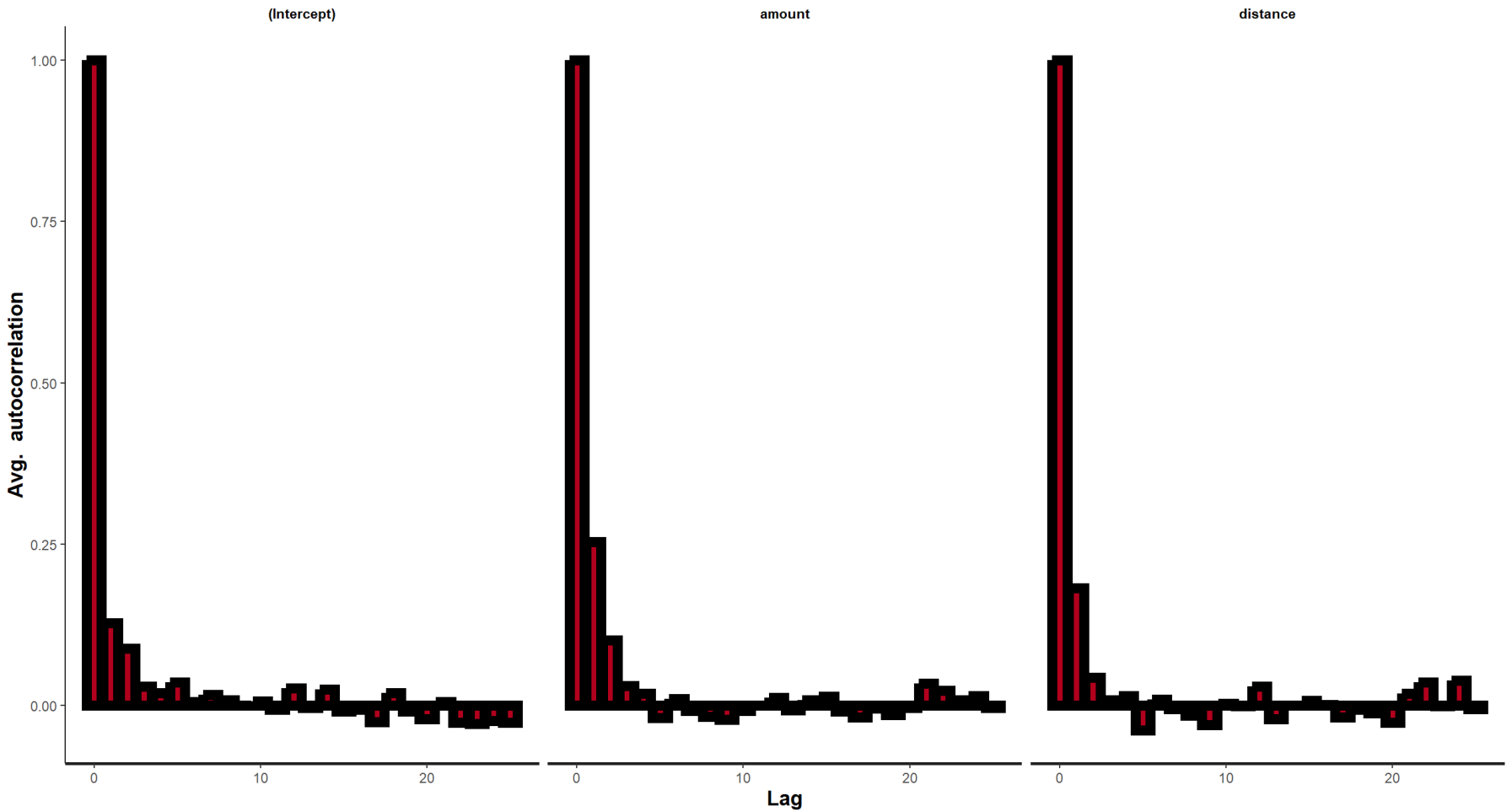
```
stan_trace(mod_ex1, nrow = 3, ncol = 1, inc_warmup = T)
```



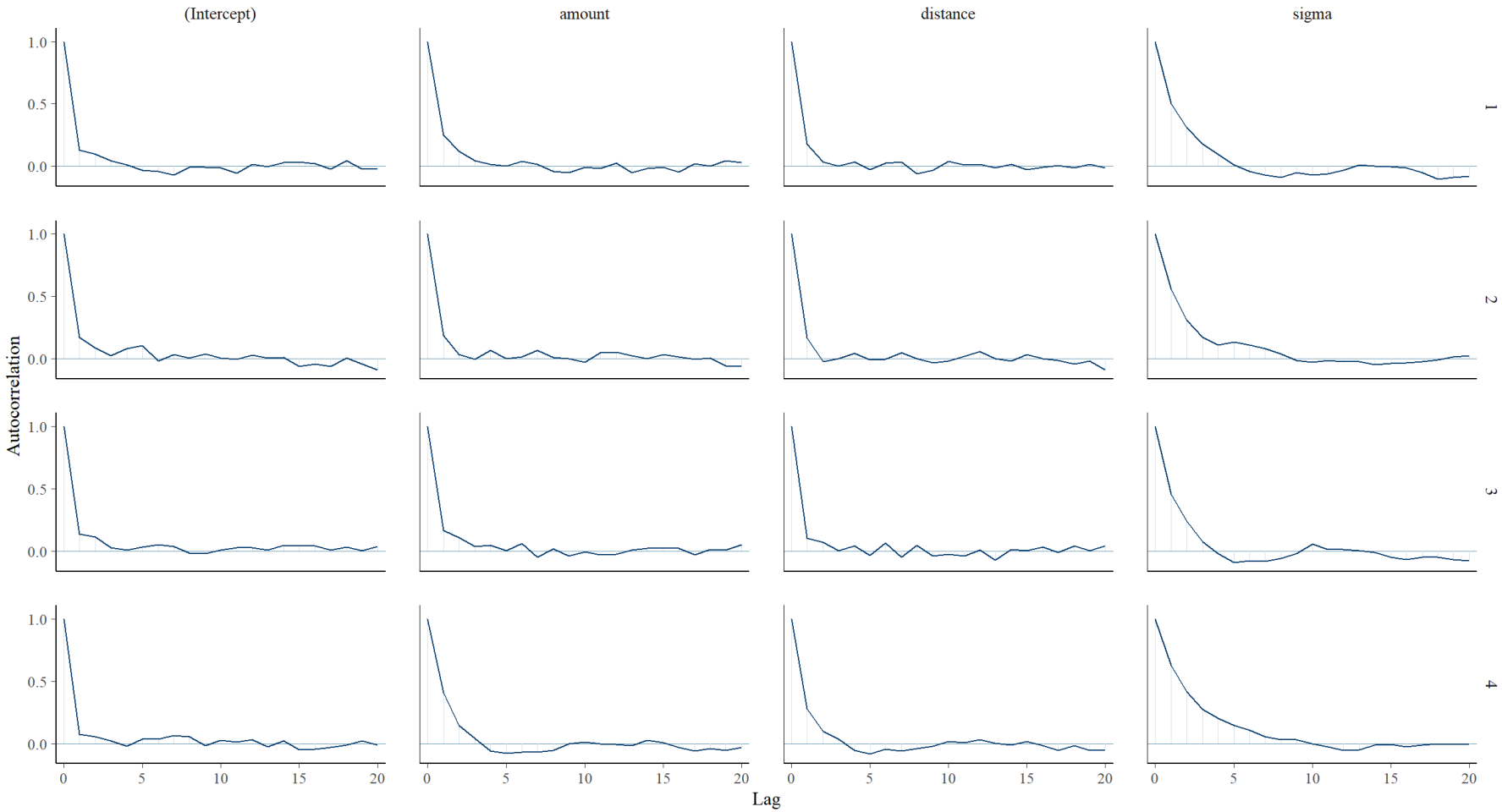
```
plot(mod_ex1, plotfun = "trace")
```



```
library(ggplot2)
stan_ac(mod_ex1)
```



```
plot(mod_ex1, plotfun = "ac")
```



```
summary(mod_ex1)
```

```
Fit Diagnostics:
           mean   sd   10%   50%   90%
mean_PPD 22.4    0.9 21.2  22.4  23.6
```

`mean_PPD` is the **sample average posterior predictive distribution of the outcome**.

A useful heuristic is to check if `mean_PPD` is plausible when compared to `mean(y)`. If it is plausible then this does not mean that the model is good in general (only that it can reproduce the sample mean), however if `mean_PPD` is implausible then it is a sign that something is wrong (severe model misspecification, problems with the data, computational issues, etc.).

In this exercise:

```
mean(data1$time)
> 22.384
```

```
summary(mod_ex1)
```

```
MCMC diagnostics
                mcse Rhat n_eff
(Intercept)     0.0  1.0  2323
amount          0.0  1.0  2244
distance        0.0  1.0  2696
sigma           0.0  1.0  1129
mean_PPD        0.0  1.0  1982
log-posterior 0.0  1.0  1178
```

NB: if you are asked to discuss the interpretation of a particular metric (e.g.: $\hat{R}$), it means that you should provide also a brief theoretical explanation of the metric itself.

```
plot(mod_ex1, plotfun = "rhat")
```

6) Posterior predictive checks

1) Generate from the posterior predictive distribution

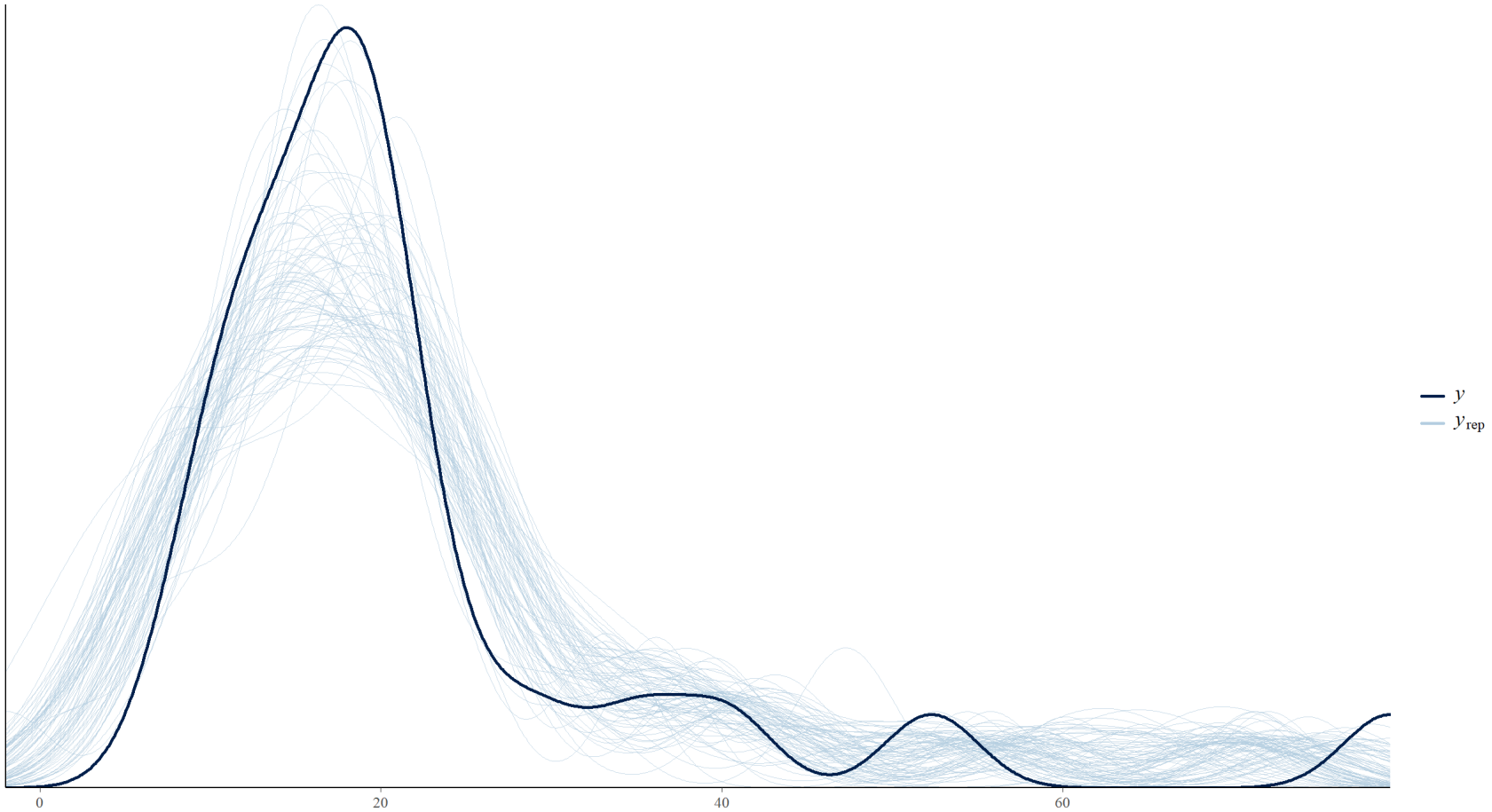
```
y_tilde <- posterior_predict(mod_ex1)
str(y_tilde)
```

```
> str(y_tilde)
 num [1:4000, 1:25] 27.4 19.6 25.9 25.4 27.4 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:25] "1" "2" "3" "4" ...
```

We generate 4000 new data sets (one for each row) of size 25 from the posterior predictive distribution.

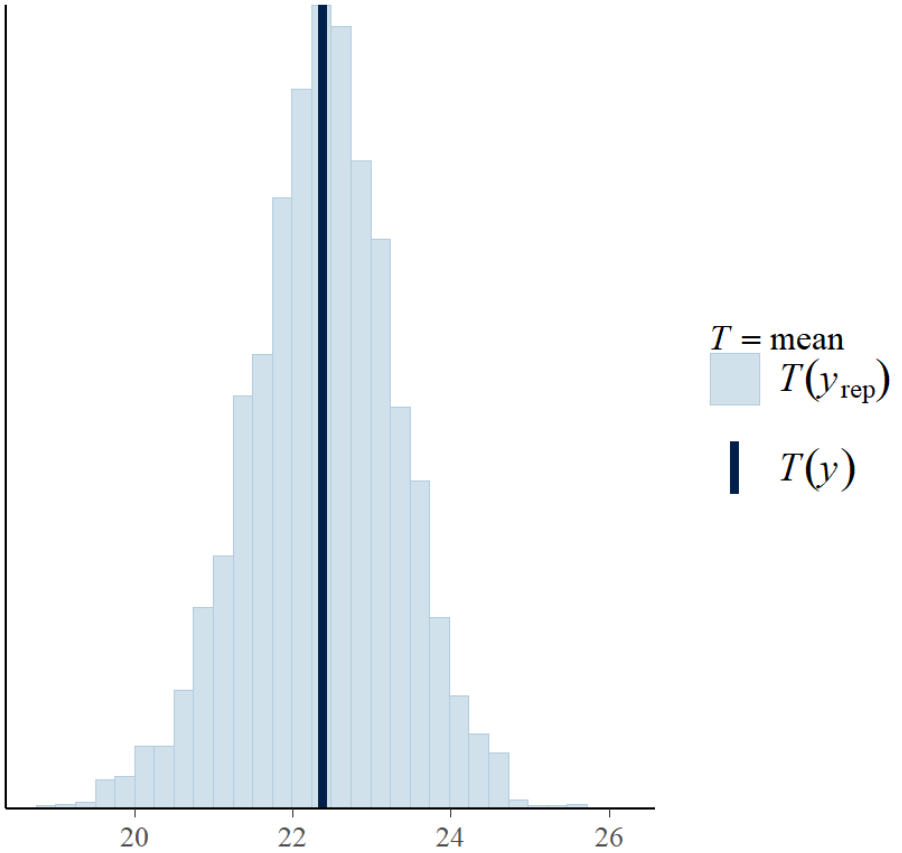
2) Densities comparison

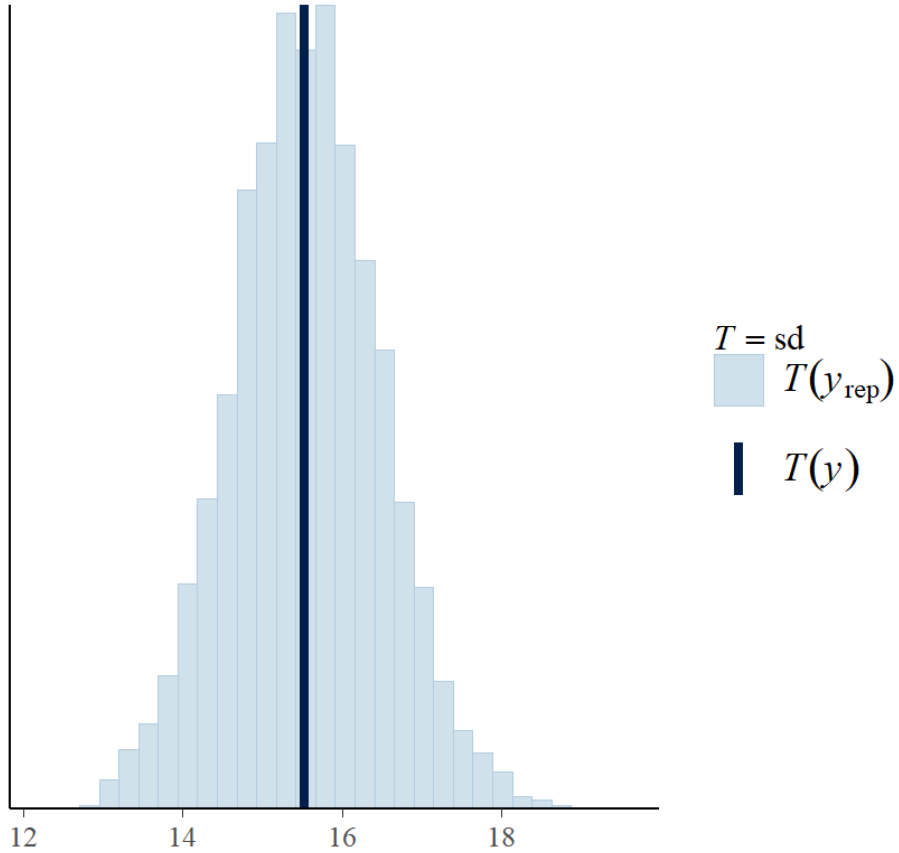
```
ppc_dens_overlay(y = data1$time,
                 yrep = y_tilde[1000:1080,])
```



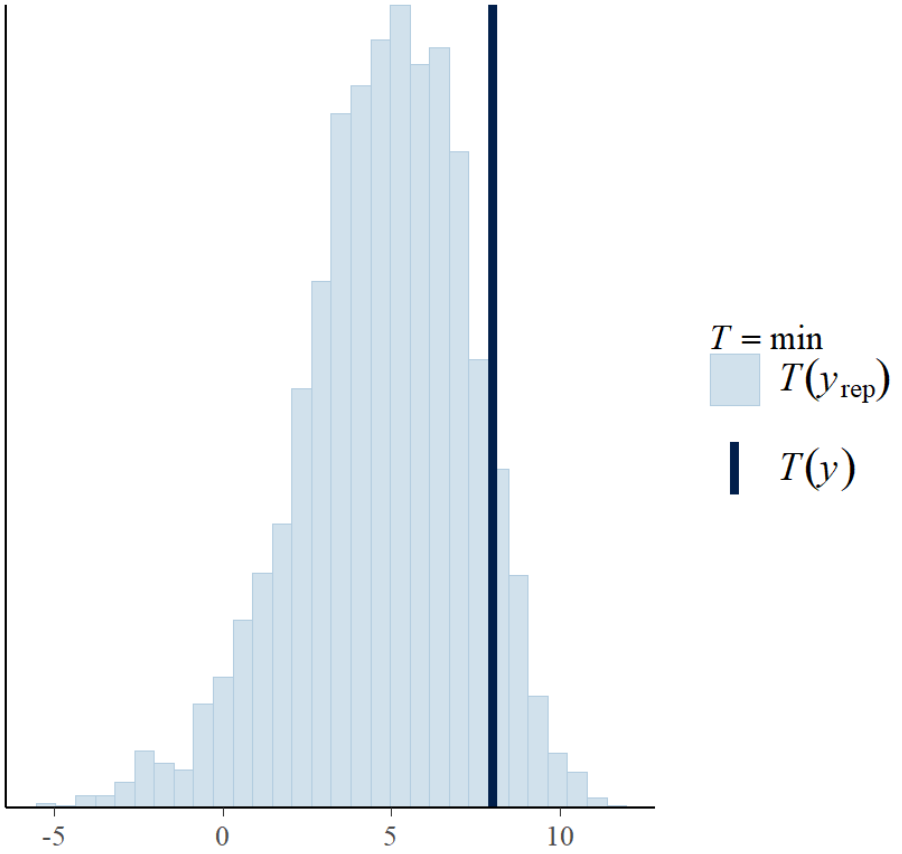
3) Posterior predictive checks

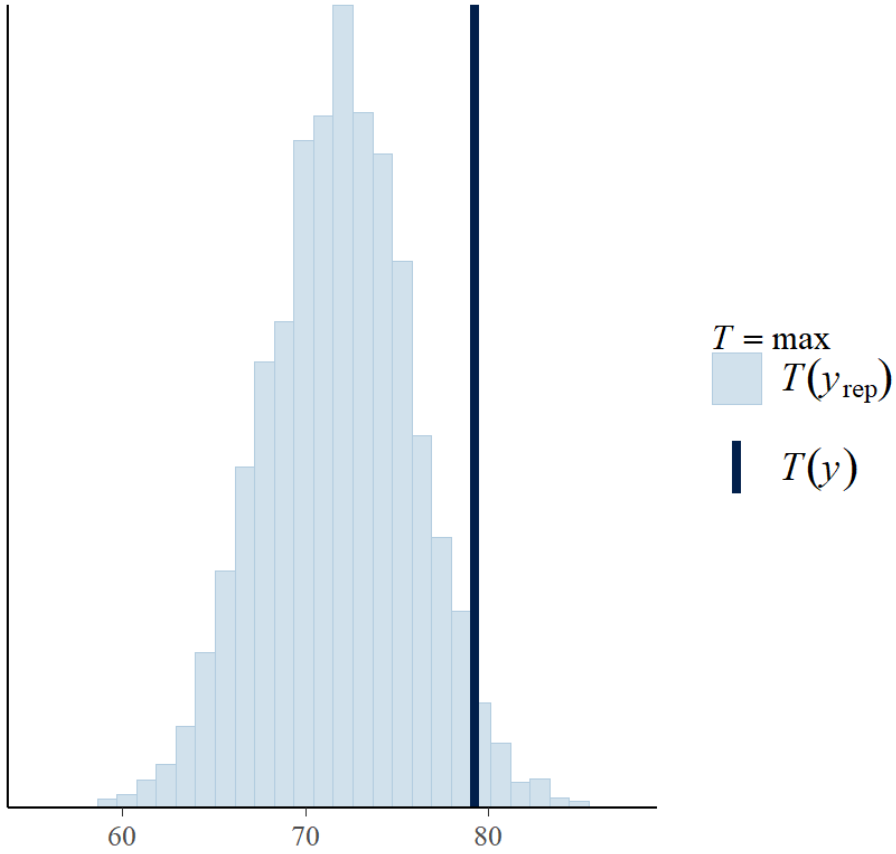
```
ppc_stat(y = data1$time, yrep = y_tilde, stat = "mean")
ppc_stat(y = data1$time, yrep = y_tilde, stat = "sd")
```





```
ppc_stat(y = data1$time, yrep = y_tilde, stat = "min")
ppc_stat(y = data1$time, yrep = y_tilde, stat = "max")
```



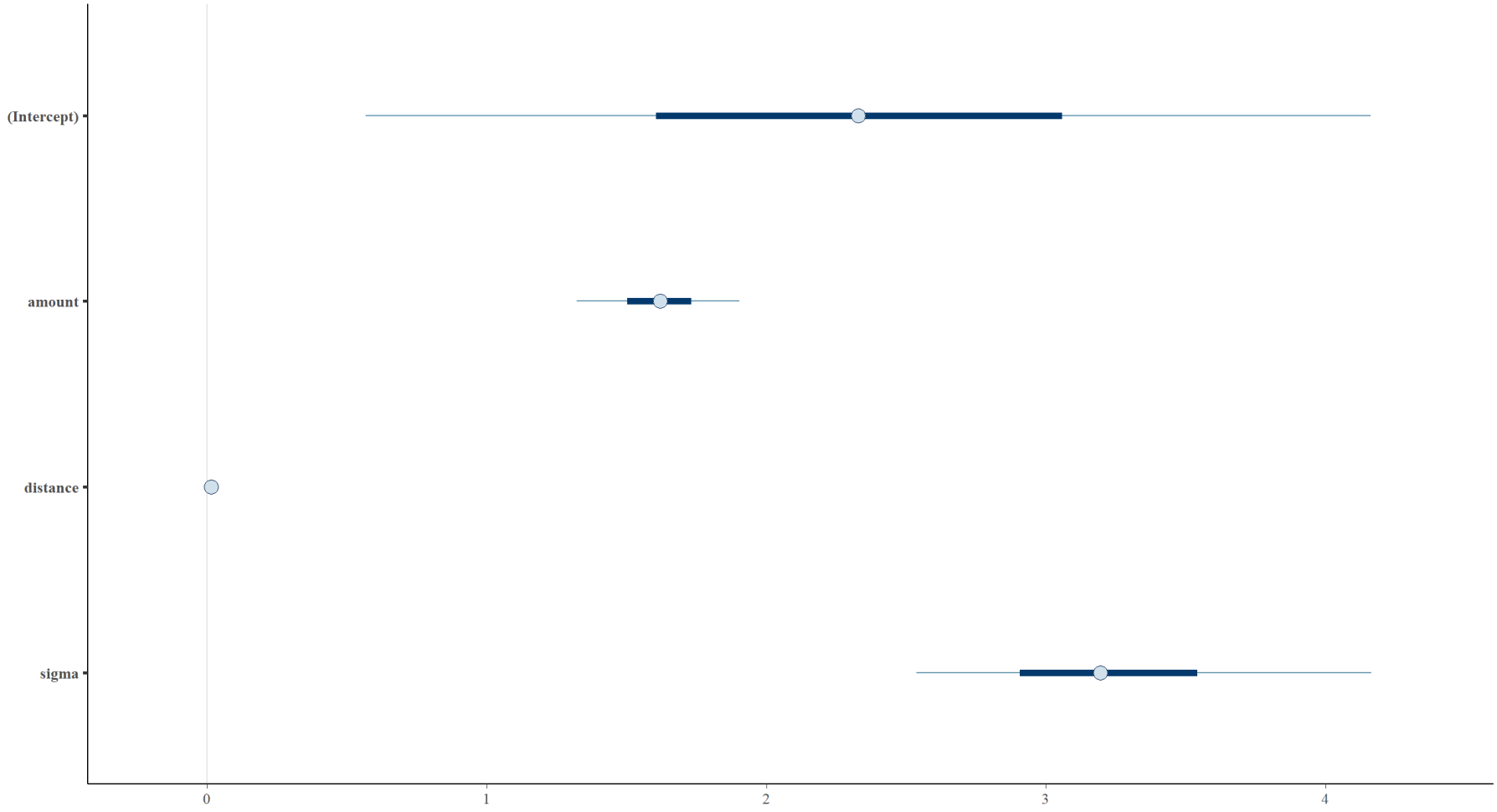


7) Posterior inference

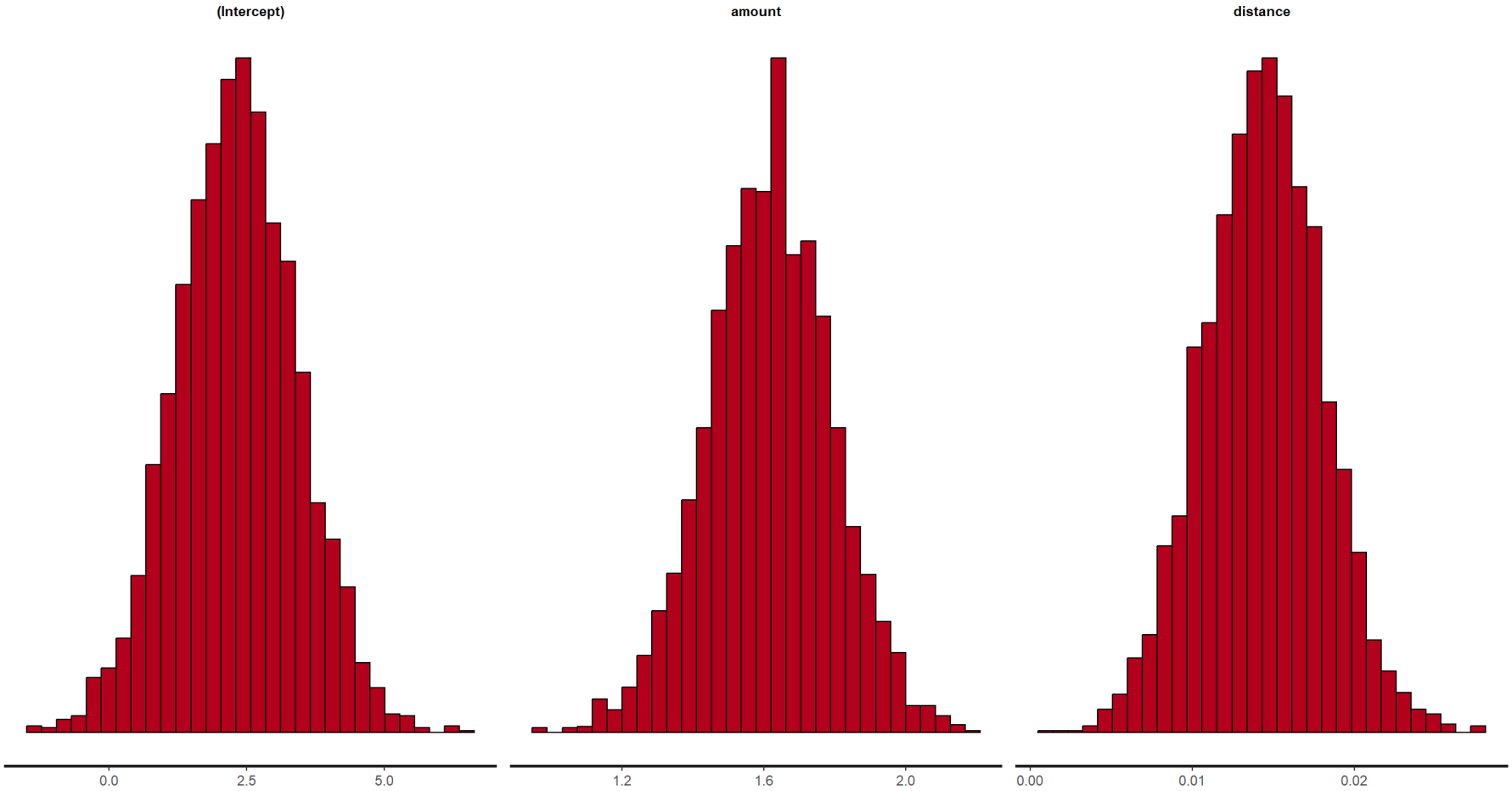
```
summary(mod_ex1,digits = 4)
```

```
> summary(mod_ex1,digits = 4)

Model Info:
 function:     stan_glm
 family:       gaussian [identity]
 formula:      time ~ amount + distance
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 25
 predictors:   3

Estimates:
              mean   sd     10%    50%    90%
(Intercept) 2.3322 1.0998 0.9270 2.3288 3.7384
amount      1.6165 0.1756 1.3882 1.6210 1.8360
distance    0.0144 0.0037 0.0097 0.0144 0.0190
sigma       3.2550 0.5033 2.6680 3.1943 3.9069
```

```
plot(mod_ex1)
```



```
stan_hist(mod_ex1)
```



Extract the posterior draws of the linear predictor

```
mu <- posterior_linpred(mod_ex1)
```

→ Report the 90% credibility interval of the posterior distribution of observation number 10.

```
mean(mu[,10]);sd(mu[,10])
quantile(mu[,10], probs = c(0.05,0.5,0.95))
```

```
> mean(mu[,10]);sd(mu[,10])
[1] 19.11945
[1] 1.536666
> quantile(mu[,10], probs = c(0.05,0.5,0.95))
      5%      50%      95%
16.58585 19.08139 21.61745
```

→ Monitor R^2_B - the Bayesian R^2

$$R^2_B = 1 - \frac{\sigma^2}{s^2(y)}.$$

```
# Extract the posterior sample of interest
sigma_post <- as.matrix(mod_ex1,pars = "sigma")

# Build the posterior distribution
n <- nrow(data1)
var_y <- var(data1$time)*(n-1)/n
R2bayes <- 1-sigma_post^2/var_y

# Posterior inference
mean(R2bayes); sd(R2bayes)
quantile(R2bayes, probs = c(0.025,0.5,0.975))
```

```
> mean(R2bayes);sd(R2bayes)
[1] 0.9531146
[1] 0.01512992
> quantile(R2bayes, probs = c(0.025,0.5,0.975))
     2.5%       50%     97.5%
0.9151463 0.9559027 0.9743654
```

```
hist(R2bayes, breaks=30)
```