Chapter 9: Simple Normal Regression

Hendra Bunyamin

2023-12-22

```
library(bayesrules)
library(tidyverse)
library(rstan)
library(rstanarm)
library(bayesplot)
library(tidybayes)
library(janitor)
library(broom.mixed)
```

Building the regression model

Dalam subbab ini, kita akan membangun framework dari model regresi **Normal Bayesian**.

Putting it all together

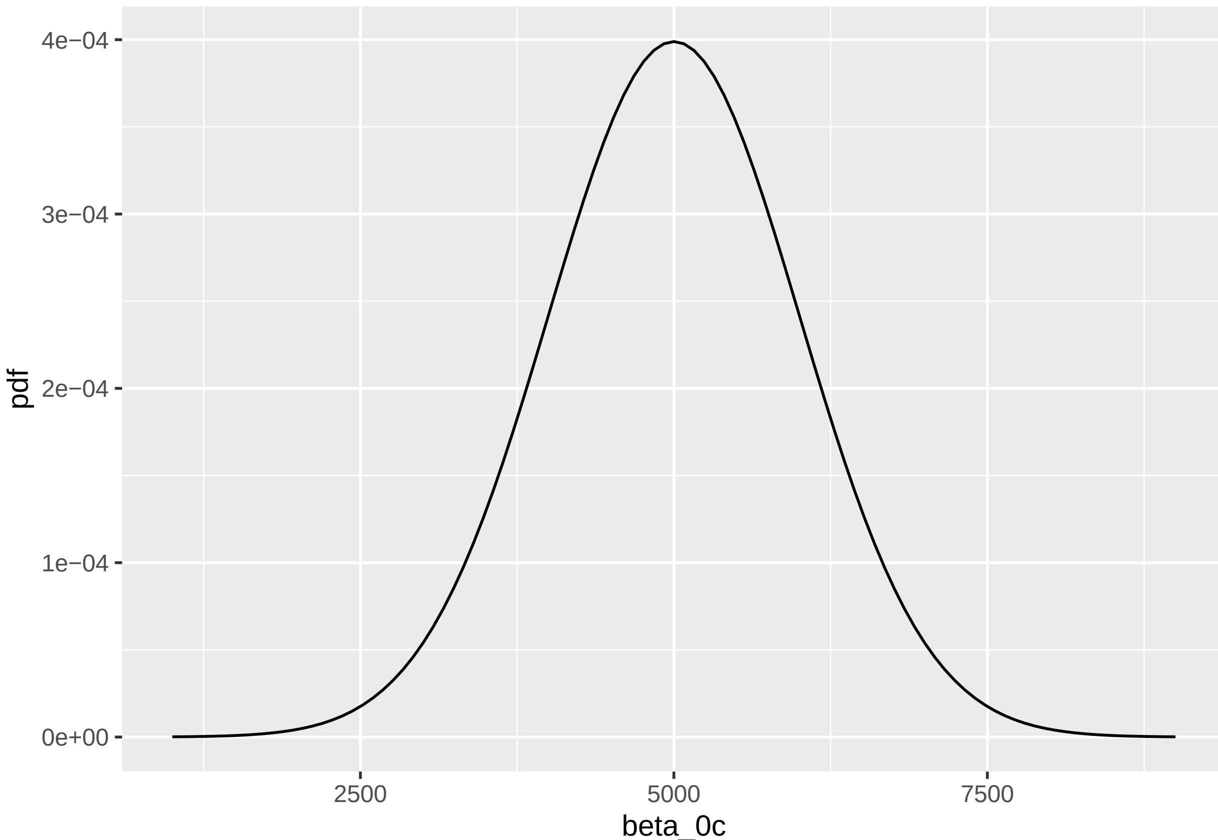
$$\begin{aligned}\text{data:}\ & Y_i \mid \beta_0, \beta_1, \sigma \sim N(\mu_i, \sigma^2) \text{ dengan } \mu_i = \beta_0 + \beta_1 X_i \\ \text{priors:}\ & \beta_0 \sim N(m_0, s_0^2) \\ & \beta_1 \sim N(m_1, s_1^2) \\ & \sigma \sim \text{Exp}(l).\end{aligned}$$

Model building dilakukan dengan one step at a time, yaitu:

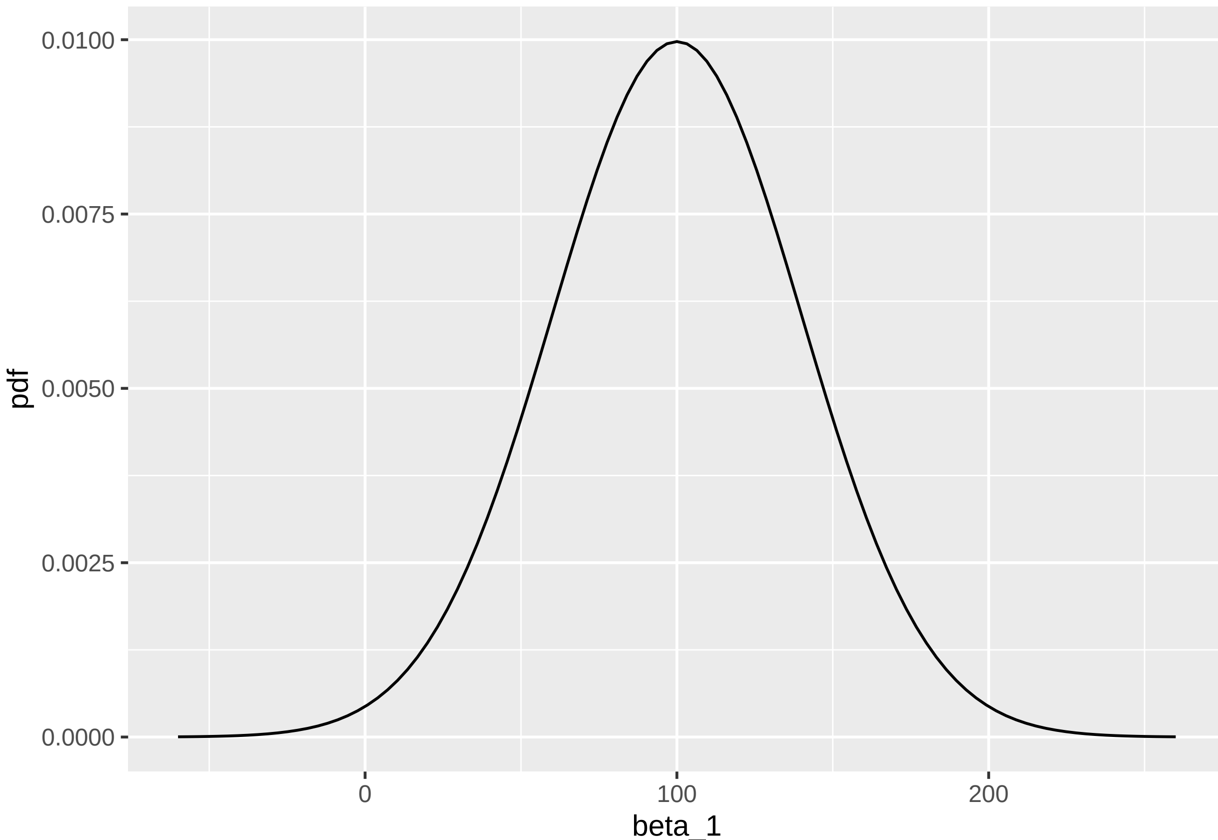
- Perhatikan apakah Y diskrit atau kontinu.
- Tuliskan bahwa the mean of Y sebagai fungsi dari prediktor X (contoh: $\mu = \beta_0 + \beta_1 X$).

Tuning prior models for regression parameters

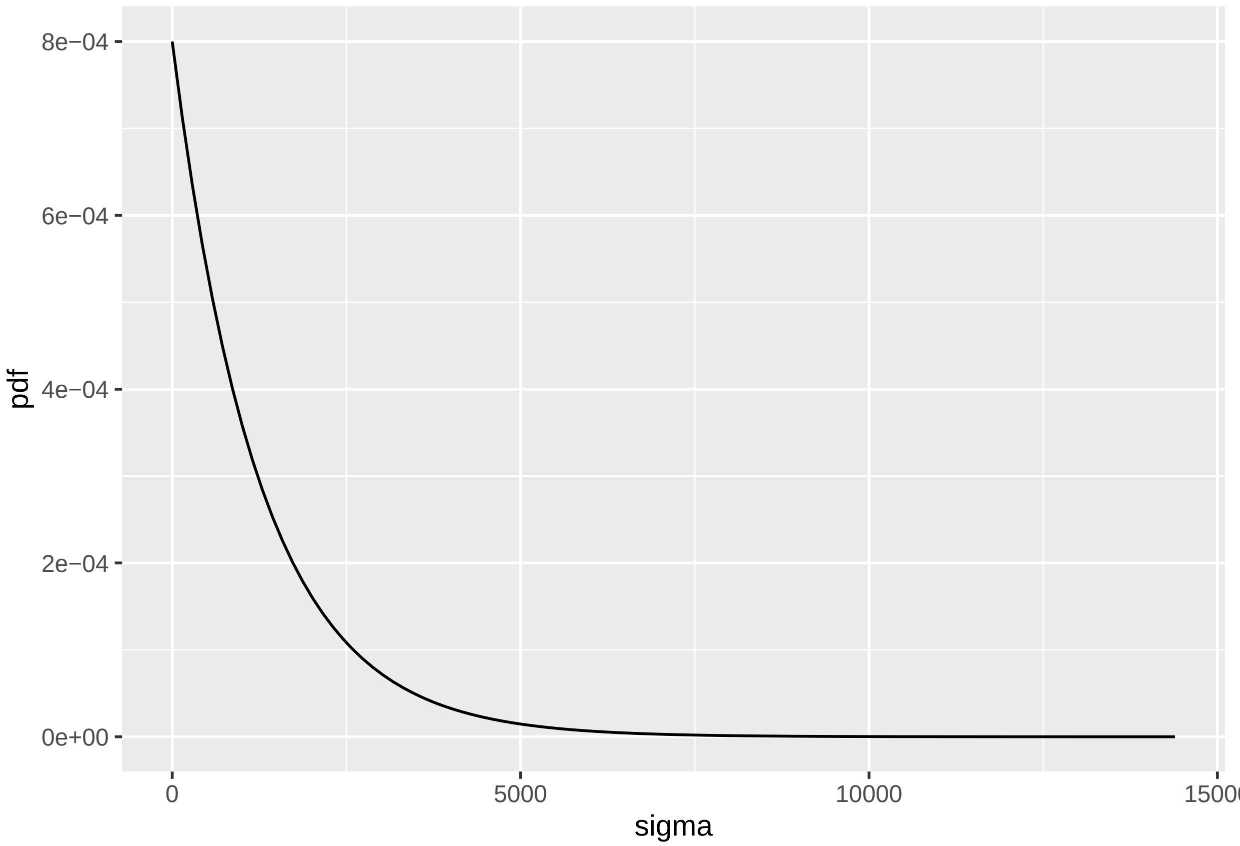
```
plot_normal(mean=5000, sd=1000) + labs( x="beta_0c", y = "pdf")
```




```r
plot_normal(mean=100, sd=40) + labs( x="beta_1", y = "pdf")
```




```r
plot_gamma(shape=1, rate=0.0008) + labs( x="sigma", y = "pdf")
```

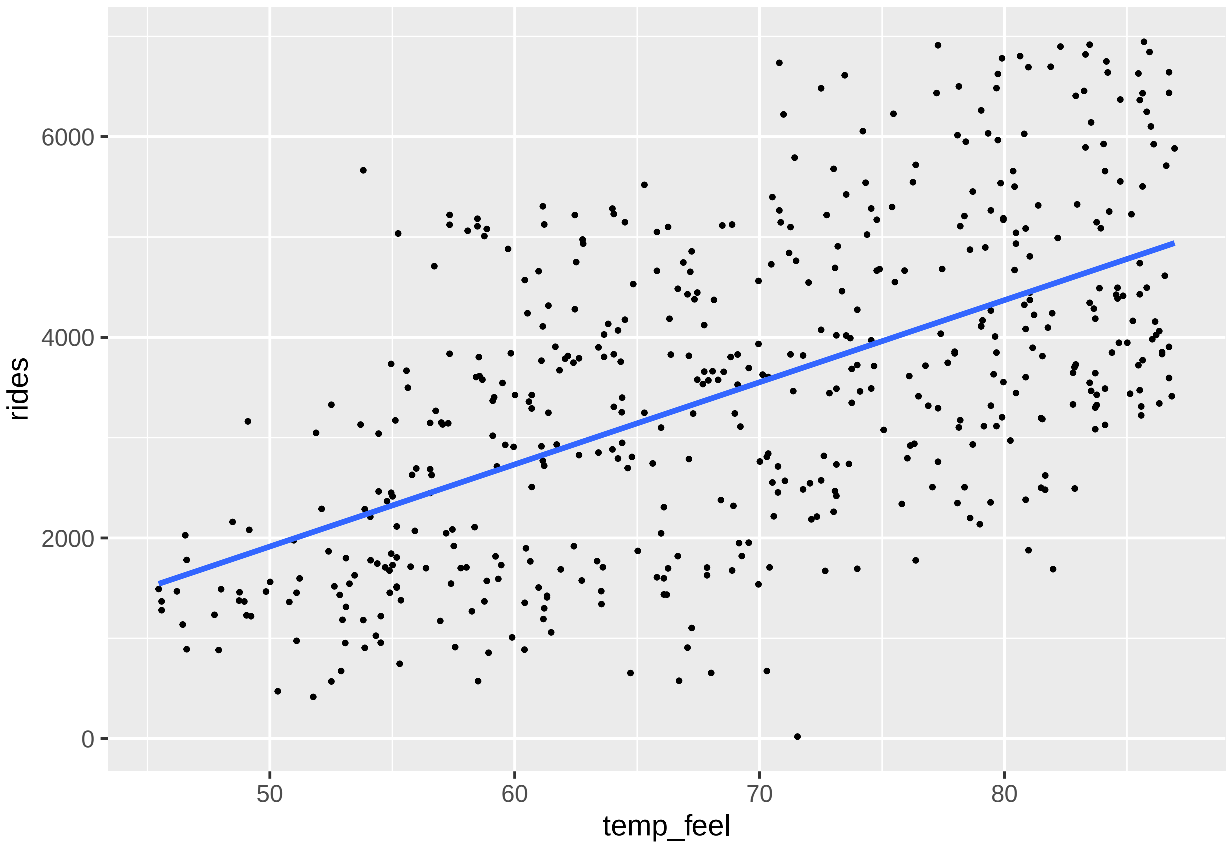


Kita akan memodelkan *ridership* (Y) dengan *temperature* (X) sebagai berikut:

$$
\begin{aligned}
\text{data:}\ & Y_i \mid \beta_0, \beta_1, \sigma \sim N(\mu_i, \sigma^2) \text{ dengan } \mu_i = \beta_0 + \beta_1 X_i \\
\text{priors:}\ & \beta_{0c} \sim N(5000, 1000^2) \\
& \beta_1 \sim N(100, 40^2) \\
& \sigma \sim \text{Exp}(0.0008).
\end{aligned}
$$

Posterior simulation

```r
# Load and plot data
data("bikes")
ggplot(bikes, aes( x = temp_feel, y = rides )) +
  geom_point(size=0.5) +
  geom_smooth(method = "lm", se=FALSE)
```



Simulation via rstanarm

Kita dapat menggunakan fungsi `stan_glm()` yang merupakan keluarga dari **generalized linear regression models (glm)**:

```
bike_model <- stan_glm( rides ~ temp_feel, data = bikes, family = gaussian, prior_intercept = normal(50
                        prior=normal(100,40),
                        prior_aux = exponential(0.0008),
                        chains=4, iter=5000*2, seed=84735)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 1.5e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
```

```
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.227 seconds (Warm-up)
## Chain 1:                0.316 seconds (Sampling)
## Chain 1:                0.543 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.08 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.251 seconds (Warm-up)
## Chain 2:                0.293 seconds (Sampling)
## Chain 2:                0.544 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 7e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.205 seconds (Warm-up)
```

```
## Chain 3:                0.291 seconds (Sampling)
## Chain 3:                0.496 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 7e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.173 seconds (Warm-up)
## Chain 4:                0.292 seconds (Sampling)
## Chain 4:                0.465 seconds (Total)
## Chain 4:
```

Selanjutnya, kita hitung nilai rasio effective sample size dan R-hat sbb:

```r
# Effective sample size ratio and Rhat
neff_ratio(bike_model)
```

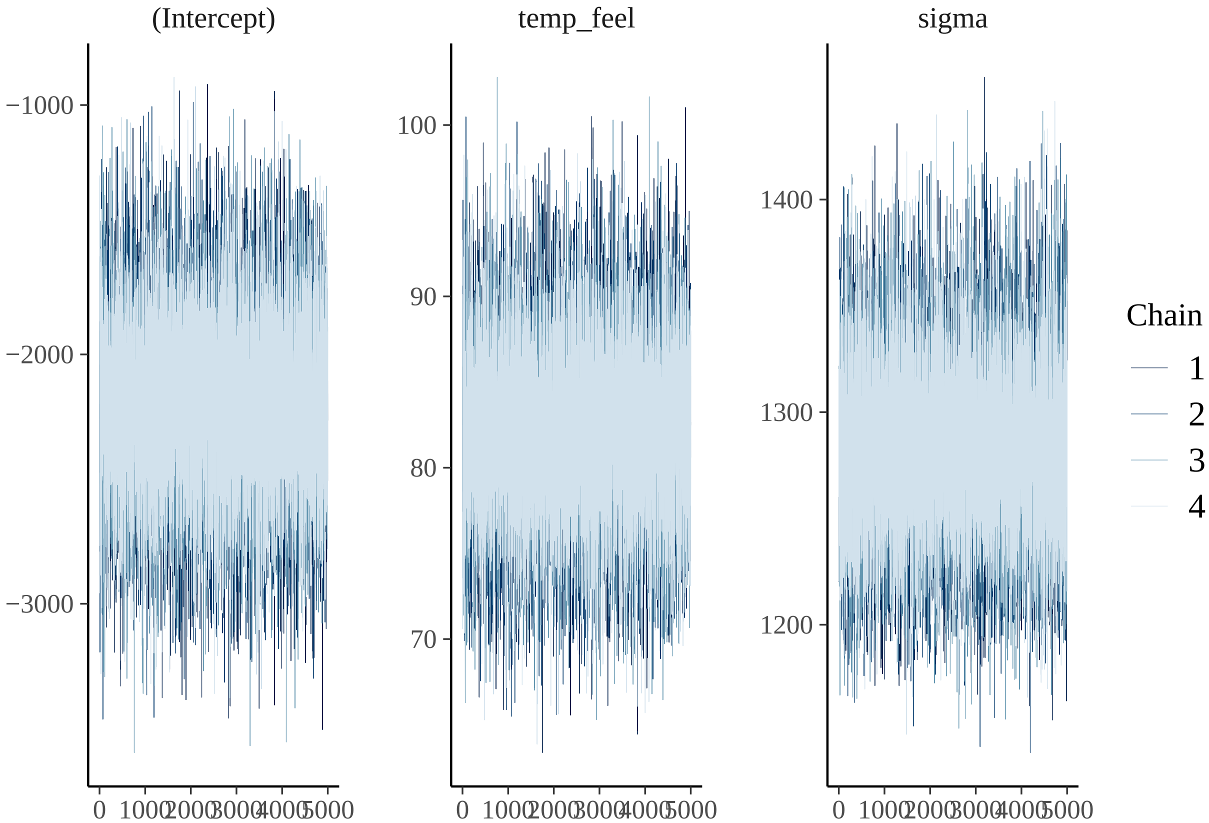
```
## (Intercept)   temp_feel       sigma
##     0.99220     0.99105     0.98165
```

```r
rhat(bike_model)
```

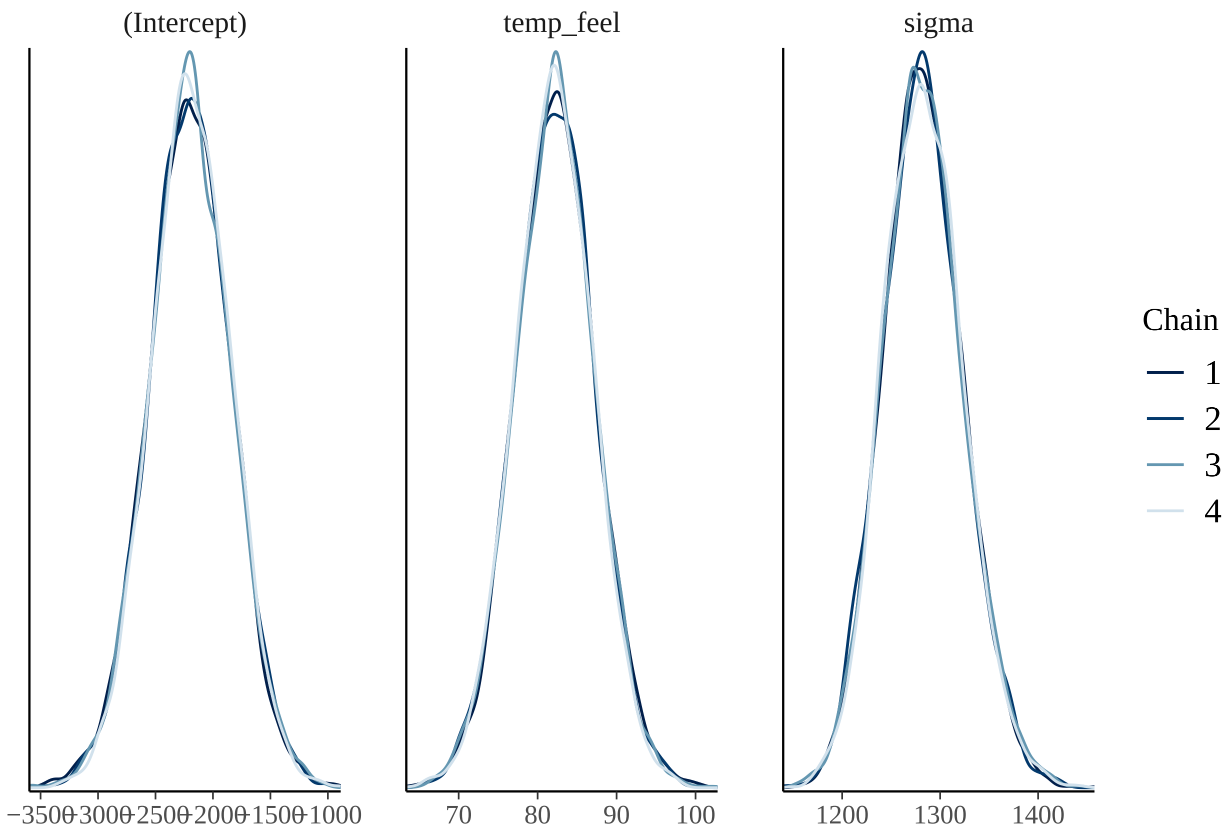
```
## (Intercept)   temp_feel       sigma
##   0.9999840   0.9999928   0.9998961
```

Kita cek juga trace dan density plots.

```r
# Trace plots of parallel chains
mcmc_trace(bike_model, size=0.1)
```




```r
# Density plots of parallel chains
mcmc_dens_overlay(bike_model)
```


(Intercept)
temp_feel
sigma
−3500
−3000
−2500
−2000
−1500
−1000
70
80
90
100
1200
1300
1400
Chain
1
2
3
4