Estimation of the effects of physiological attributes on Cardiovascular Disease: Bayesian Logistic Regression.

Md Tahsin Anam

Table of Contents

Abstract: The aim of this project is to estimate effects of important physiological attributes on cardiovascular disease. There are 13 important attributes considered for the analysis. The Bayesian Logistic Regression was used to estimate the model parameters. To estimate the model parameters normal prior was used. The results suggested varying effects of the attributes on cardiovascular disease. Also, the effects of the attributes on different severity of cardiovascular disease are different.

Methodology:

Data:

The data is collected from UCI Machine Learning repository. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. The attributes are as followed,

Id	Attribute Name	Data Type	Description
1	Age	Continuous	Age in Year
2	Sex	Discrete	1 = male; 0 = female
3	Cp	Discrete	Chest pain type
4	trestbps	Continuous	Resting blood pressure (in mm Hg on admission to the hospital)
5	chol	Continuous	Serum cholestoral in mg/dl
6	fbs	Discrete	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
7	restecg	Discrete	Resting electrocardiographic results
8	thalach	Continuous	Maximum heart rate achieved
9	exang	Discrete	Exercise induced angina (1 = yes; 0 = no)
10	oldpeak	Continuous	ST depression induced by exercise relative to rest
11	slope	Discrete	The slope of the peak exercise ST segment
12	ca	Discrete	Number of major vessels (0-3) colored by flourosopy
13	thal	Discrete	3 = normal; 6 = fixed defect; 7 = reversable defect
14	classs	Discrete	Diagnosis of heart disease (angiographic disease status)

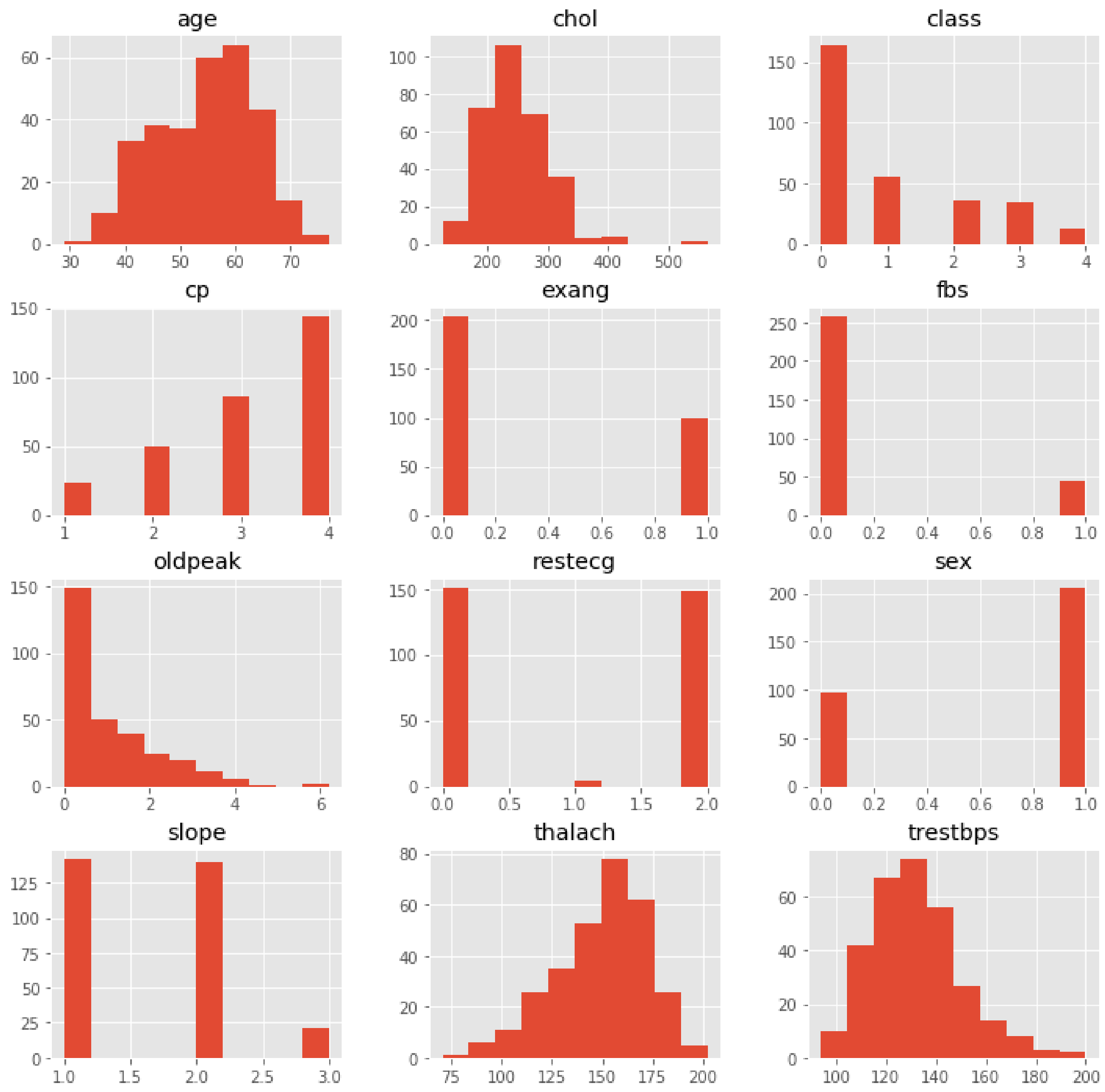


Figure 1. Histogram of the Attributes for estimating effects of Cardiovascular disease.

We have performed 5 different logistic regression models.

1. No Cardiovascular Disease vs. All level of Cardiovascular Disease
2. No Cardiovascular Disease vs. Cardiovascular Disease, severity = 1
3. No Cardiovascular Disease vs. Cardiovascular Disease, severity = 2

4. No Cardiovascular Disease vs. Cardiovascular Disease, severity = 3
5. No Cardiovascular Disease vs. Cardiovascular Disease, severity = 4

Analysis:

For a logistic model the likelihood for observation y is,

$$\binom{n}{y}\pi^{y}(1-\pi)^{(n-y)}$$

Here,

n: total number of observations

and,

$\pi = g^{-1}(\eta)$ is the probability of success.

$\eta = \alpha + X^{T}\beta$ is the linear predictor.

Where, X is the design matrix consisted of the predictor(attributes)

And, α and β are vector of coefficients.

We used probit link function.

$$g(x) = \phi^{-1}(x)$$

Which gives the likelihood,

$$\binom{n}{y}\phi^{-1}(x)^{y}(1-\phi^{-1}(x))^{(n-y)}$$

Here, ϕ is the CDF of the standard normal distribution.

Posterior Distribution:

The posterior distribution for the coefficient α and β is proportional to the product of priors and Likelihood function.

$$f(\alpha, \beta | y, X) \propto \prod_{k=1}^{K} f(\beta_k) \prod_{i=1}^{n} g^{-1}(\eta_i)^{y_i} (1 - g^{-1}(\eta_i))^{(n_i - y_i)}$$

For Prior distribution, normal prior with mean 0 and variance 1 was used.

The posterior distribution was drawn using MCMC.

The convergence of the estimation was availed by first half of the draw as warm up(burn). The convergence of the distribution checked from the results.

A positive coefficient means that an increase in the predictor leads to an increase in the predicted probability. A negative coefficient means that an increase in the predictor leads to a decrease in the predicted probability.

In the summary of results the following outcomes are presented

1. Effect Estimates: The mean of the posterior distribution with 95% Credible Interval.
2. Standard Deviation: The standard deviation of the posterior distribution.

Software:

For the analysis R (The R Project for Statistical Computing) was used. For Bayesian Logistic Regression ‘rstanarm’ was used, for plotting ‘bayesplot’ and ‘ggplot2’ packages were used.

Results:

Comparison of all Cardiovascular Disease

		Cardiovascular Disease		
	Coefficient	2.50%	97.50%	SD
(INTERCEPT)	-6.11	-12.29	-0.3	3.045818
AGE	-0.02	-0.07	0.03	0.025436
SEX1	1.63	0.62	2.77	0.522271
CP2	1.19	-0.4	2.68	0.777781
CP3	0.12	-1.19	1.43	0.647701
CP4	2.29	0.97	3.6	0.652577
TRESTBPS	0.03	0.01	0.05	0.011745
CHOL	0	0	0.01	0.004019
FBS1	-0.52	-1.64	0.59	0.571681
RESTECG1	0.67	-2.4	3.98	1.71668
RESTECG2	0.51	-0.29	1.32	0.394571
THALACH	-0.02	-0.05	0	0.012124
EXANG1	0.68	-0.22	1.57	0.452423
OLDPEAK	0.48	0.02	0.94	0.230107
SLOPE2	1.21	0.26	2.16	0.483126
SLOPE3	0.5	-1.22	2.23	0.898075
CA1.0	2.22	1.3	3.2	0.507283
CA2.0	3.05	1.59	4.52	0.732342
CA3.0	2.14	0.53	3.95	0.864367
THAL6.0	0.04	-1.61	1.64	0.841815
THAL7.0	1.53	0.67	2.41	0.427471

Comparison of Cardiovascular Disease severity = 1

	Cardiovascular Disease Severity = 1			
	Coefficient	2.50%	97.50%	SD
(INTERCEPT)	-7.8	-14.62	-1.29	3.464183
AGE	0	-0.06	0.06	0.028962
SEX1	1.69	0.5	2.89	0.601837
CP2	0.62	-0.98	2.23	0.829041
CP3	-0.08	-1.58	1.42	0.778531
CP4	1.78	0.37	3.29	0.762851
TRESTBPS	0.03	0	0.05	0.013327
CHOL	0	-0.01	0.01	0.004762
FBS1	-1.46	-3.17	0.04	0.799099
RESTECG1	-0.42	-5.03	3.68	2.159266
RESTECG2	0.54	-0.29	1.44	0.450256
THALACH	-0.01	-0.04	0.02	0.013986
EXANG1	0.51	-0.53	1.52	0.527887
OLDPEAK	0.18	-0.38	0.71	0.283101
SLOPE2	1.09	0.1	2.11	0.52233
SLOPE3	0.25	-2.16	2.18	1.065974
CA1.0	1.72	0.63	2.83	0.566408
CA2.0	2.03	0.29	3.74	0.889364
CA3.0	1.12	-1.14	3.38	1.171248
THAL6.0	-1.01	-3.01	0.84	1.014831
THAL7.0	1.16	0.2	2.17	0.519108

Comparison of Cardiovascular Disease severity = 2

	Cardiovascular Disease Severity = 2			
	Coefficient	2.50%	97.50%	SD
(INTERCEPT)	-7.4	-19.85	3.65	6.321124
AGE	-0.01	-0.12	0.11	0.059658
SEX1	1.55	-0.31	3.74	0.999443
CP2	0.39	-3.11	3.43	1.71771
CP3	-0.33	-2.8	2.04	1.320454
CP4	3.17	1.02	5.51	1.123834
TRESTBPS	0.02	-0.03	0.07	0.024132
CHOL	0.01	-0.01	0.03	0.008733
FBS1	0.62	-1.25	2.53	0.946439
RESTECG1	0.77	-3.17	4.92	2.15454
RESTECG2	-0.38	-1.96	1.25	0.804302
THALACH	-0.04	-0.09	0	0.021534
EXANG1	1.23	-0.41	2.92	0.86095
OLDPEAK	1.1	0.2	2.06	0.467549
SLOPE2	1.4	-0.3	3.3	0.932067
SLOPE3	0.57	-2.4	3.5	1.553148
CA1.0	4.37	2.42	6.49	1.040949
CA2.0	3.06	0.57	5.54	1.243559
CA3.0	2.84	0.44	5.45	1.231769
THAL6.0	1.71	-0.54	3.92	1.120969
THAL7.0	1.88	0.27	3.65	0.871163

Comparison of Cardiovascular Disease severity = 3

	Cardiovascular Disease Severity = 3			
	Coefficient	2.50%	97.50%	SD
(INTERCEPT)	-2.1	-16.3	11.37	7.181702
AGE	-0.11	-0.25	0.03	0.069026
SEX1	0.98	-1.44	3.39	1.257145
CP2	2.45	-0.76	5.72	1.612775
CP3	0.82	-1.82	3.37	1.2834
CP4	3.2	0.61	5.67	1.321983
TRESTBPS	0.02	-0.03	0.07	0.027551
CHOL	0	-0.02	0.02	0.009713
FBS1	1.92	-0.37	4.32	1.190263
RESTECG1	0.87	-3.17	5.09	2.160681
RESTECG2	2.25	0.47	4.27	0.981702
THALACH	-0.05	-0.11	0	0.028869
EXANG1	0.98	-1.19	3.09	1.093076
OLDPEAK	1.15	0.3	2.1	0.457852
SLOPE2	1.95	0.11	3.89	0.943403
SLOPE3	0.46	-2.6	3.59	1.573713
CA1.0	2.18	-0.34	4.59	1.211327
CA2.0	5.84	3.37	8.32	1.254721
CA3.0	1.8	-0.89	4.61	1.437936
THAL6.0	-0.71	-4.8	3.11	2.119764
THAL7.0	3.93	1.88	6.33	1.119319

Comparison of Cardiovascular Disease severity = 4

	Cardiovascular Disease Severity = 4			
	Coefficient	2.50%	97.50%	SD
(INTERCEPT)	-31.7	-57.31	-10.55	12.02378
AGE	0.14	-0.03	0.33	0.08851
SEX1	0.95	-2.32	4.19	1.633845
CP2	-0.12	-4.57	4.14	2.318994
CP3	-0.77	-4.17	2.41	1.684778
CP4	3.8	0.95	7	1.551706
TRESTBPS	0.02	-0.05	0.1	0.03817
CHOL	-0.01	-0.03	0.02	0.011938
FBS1	0.05	-3.66	3.7	1.828402
RESTECG1	0.9	-3.45	5.22	2.261564
RESTECG2	1.8	-0.58	4.25	1.25033
THALACH	0.05	-0.03	0.13	0.041567
EXANG1	2.02	-0.45	4.65	1.287033
OLDPEAK	2.26	1.05	3.75	0.693494
SLOPE2	2.06	-0.73	5.04	1.41662
SLOPE3	-0.52	-4.54	3.2	1.909505
CA1.0	2.22	-0.77	5.27	1.493041
CA2.0	1.69	-1.72	5.15	1.735799
CA3.0	4.22	1.26	7.13	1.50193
THAL6.0	0.77	-2.67	4.04	1.691288
THAL7.0	3.11	0.37	5.85	1.423633

Discussion:

The effects of each of the attributes on heart disease is summarized bellow

Age: The effect of Age on Heart Disease is marginal. But, when we considered the severity of heart disease, the effect of age has noticeable effect on heart disease. For heart disease severity = 4, the effect of Age is 0.14 with sd = 0.09.

Sex: From the result, it is apparent that Male are more probable to get heart disease. The z-score male is 1.63 more than that of female.

Chest pain type (CP): If we consider the Chest Pain type, we observe that the z-value increases as the CP move from 1 to 4. It increases to 2.29 for the chest pain type 4.

Resting blood pressure (trestbps) : The blood pressure is a very important marker for heart disease. We observe the Resting Blood Pressure coefficient is 0.03 with sd = 0.01, which makes it a positive indication for heart disease.

Cholesterol (Chol): The effect of cholesterol has provided a very marginal effect on the detection of heart disease with mean of the coefficient is close to zero.

Fasting blood sugar (FBS): The Fasting Blood Sugar has very wide range of effect with coefficient -0.52 with sd = 0.57 . The high sd makes it a less reliable attribute.

Resting electrocardiographic results (RESTECG) : The Resting electrocardiographic results are positive indicator of heart disease. The coefficient of RESTECG 1 is 0.67 and RESTECG2 is 0.51 .

Exercise induced angina(EXANG): The presence of Exercise induced angina is a positive indicator of heart disease with the coefficient is 0.68 and sd = 1.57 .

ST depression induced by exercise relative to rest (Oldpeak): The oldpeak has a positive effect on the prediction of heart disease with the coefficient = 0.48 and sd = 0.23 .

The slope of the peak exercise ST segment (Slope) : The slope is a positive indication on the heart disease with coefficient 1.21 for slope 2 and 0.5 for slope 3.

Number of major vessels (0-3) colored by flourosopy (CA) : The attribute ca is also a positive indication for heart disease with coefficients 2.22 for ca = 1, 3.05 for ca = 2 and 2.14 for ca = 3.

Thal : The attribute Thal is a positive indicator of heart disease with fixed defect has positive coefficient 0.04 and reversible coefficient 1.53 .

Model Criticism:

1. MCMC Convergence of the estimation of the parameters reached.
2. The interaction among the attributes should be checked for better estimation of the effects of the attributes.
3. Different power of the attributes should be checked for further research.

Summary:

The important attributes for the indication of heart disease are

Sex, Chest Pain(Cp), Resting ECG(RestECG), Maximum heart rate achieved(thalach), ST depression induced by exercise relative to rest(oldpeak) , The slope of the peak exercise ST segment(Slope), Number of major vessels (0-3) colored by flourosopy (CA) and defect(thal).

Bayesian Inference Project

Tahsin Anam

December 17, 2018

Install necessary packages

```
library(rstanarm)

## Loading required package: Rcpp

## rstanarm (Version 2.18.2, packaged: 2018-11-08 22:19:38 UTC)

## - Do not expect the default priors to remain the same in future rstanarm v
ersions.

## Thus, R scripts should specify priors explicitly, even if they are just th
e defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend cal
ling

## options(mc.cores = parallel::detectCores())

## - Plotting theme set to bayesplot::theme_default().

library(ggplot2)
library(bayesplot)

## This is bayesplot version 1.6.0

## - 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
```

Loading and Exploring the data

```
# Import data
url = "http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease
/processed.cleveland.data"
data <- read.table(url, header = TRUE, sep = ',')

# Change the column names
# changing the names of the attributes
names = c('age',
          'sex',
          'cp',
          'trestbps',
          'chol',
          'fbs',
          'restecg',
          'thalach',
          'exang',
          'oldpeak',
          'slope',
          'ca',
          'thal',
          'class')

colnames(data) <- names

data[c('class', 'cp', 'exang', 'fbs', 'restecg', 'sex', 'slope', 'ca', 'thal'
)] <- lapply(data[c('class', 'cp', 'exang', 'fbs', 'restecg', 'sex', 'slope',
'ca', 'thal')], factor)

# removing rows which have missing values
data1 <- data[data[,'ca'] != '?',]
data2 <- data1[data1[,'thal'] != '?',]
```

Analyze for No Cardiovascular disease and Cardiovascular disease

```
data3 <- data2

l <- function(x){
  if (x >= 1){
    return(1)
  }
  else
    return(0)
}

data3$class <- apply(data3['class'], 1, l)


normal_prior <- normal(location = 0, scale = NULL, autoscale = TRUE)
post1 <- stan_glm(class ~ ., data = data3,
                  family = binomial(link = "logit"),
                  prior = normal_prior, prior_intercept = normal_prior)

##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.024 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 240 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: 2.451 seconds (Warm-up)
## Chain 1:                2.077 seconds (Sampling)
## Chain 1:                4.528 seconds (Total)
## Chain 1: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2: 
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.431 seconds (Warm-up)
## Chain 2:                2.082 seconds (Sampling)
## Chain 2:                4.513 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.411 seconds (Warm-up)
## Chain 3:                2.101 seconds (Sampling)
## Chain 3:                4.512 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.419 seconds (Warm-up)
## Chain 4:                2.107 seconds (Sampling)
## Chain 4:                4.526 seconds (Total)
## Chain 4: 
```

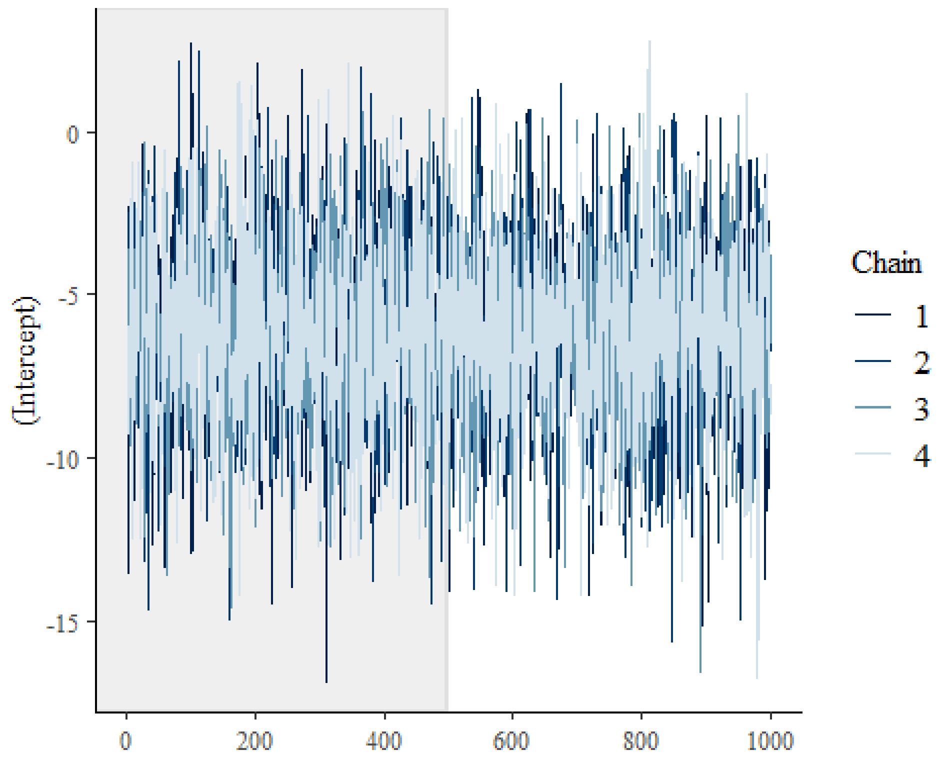
```
round(coef(post1), 2)
```

```
## (Intercept)         age        sex1         cp2         cp3         cp4 
##       -6.17       -0.02        1.65        1.19        0.11        2.25 
##    trestbps        chol        fbs1    restecg1    restecg2     thalach 
##        0.03        0.00       -0.49        0.63        0.52       -0.02
##      exang1     oldpeak      slope2      slope3       ca1.0       ca2.0
##        0.69        0.48        1.21        0.52        2.21        3.06
##       ca3.0     thal6.0     thal7.0
##        2.10        0.02        1.52
```

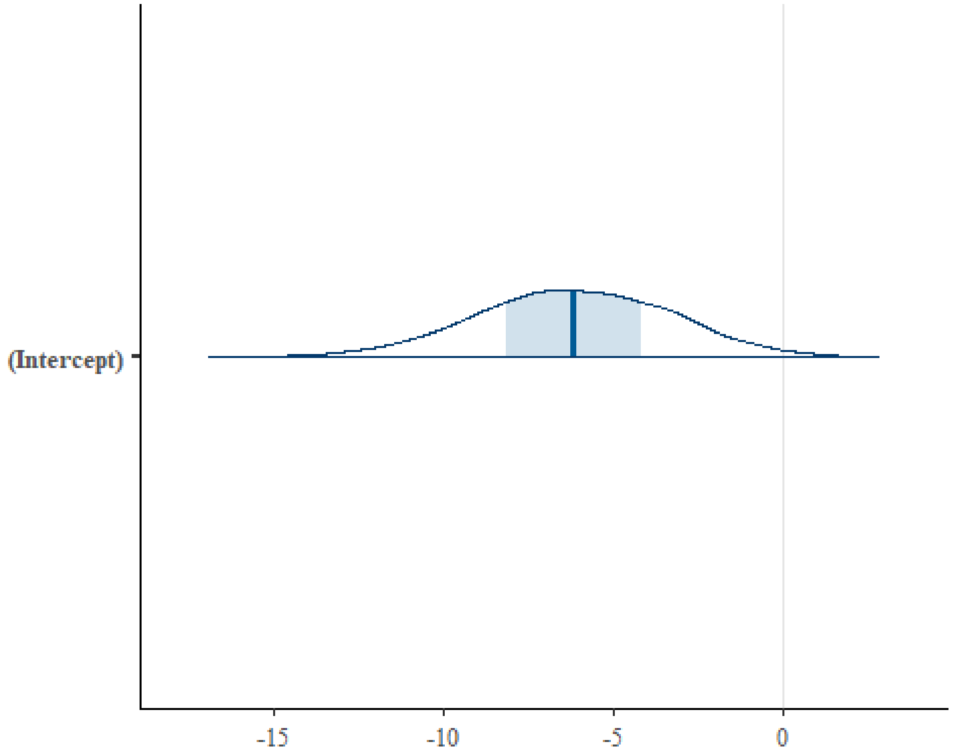
```r
round(posterior_interval(post1, prob = 0.95), 2)
```

```
##               2.5% 97.5%
## (Intercept) -12.09 -0.64
## age          -0.07  0.03
## sex1          0.65  2.76
## cp2          -0.24  2.70
## cp3          -1.23  1.43
## cp4           1.00  3.60
## trestbps      0.00  0.05
## chol          0.00  0.01
## fbs1         -1.70  0.67
## restecg1     -2.40  3.97
## restecg2     -0.28  1.32
## thalach      -0.05  0.00
## exang1       -0.20  1.61
## oldpeak       0.01  0.96
## slope2        0.29  2.16
## slope3       -1.32  2.29
## ca1.0         1.22  3.23
## ca2.0         1.72  4.58
## ca3.0         0.52  3.90
## thal6.0      -1.58  1.66
## thal7.0       0.67  2.41
```

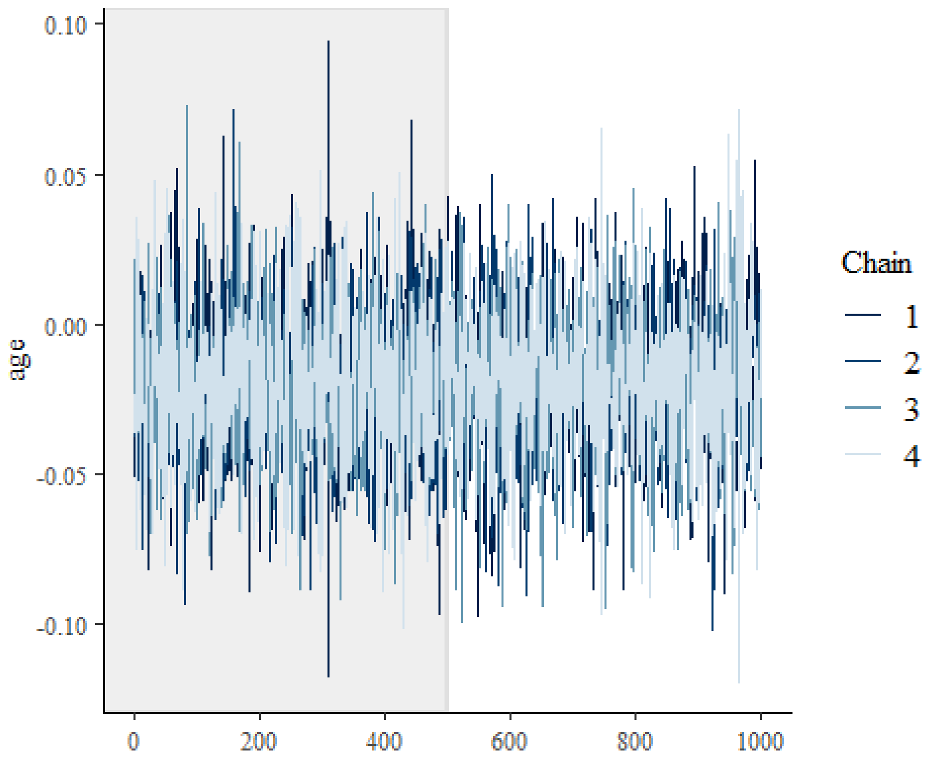
```r
mcmc_trace(as.array(post1),  pars = c('(Intercept)'), n_warmup = 500)
```



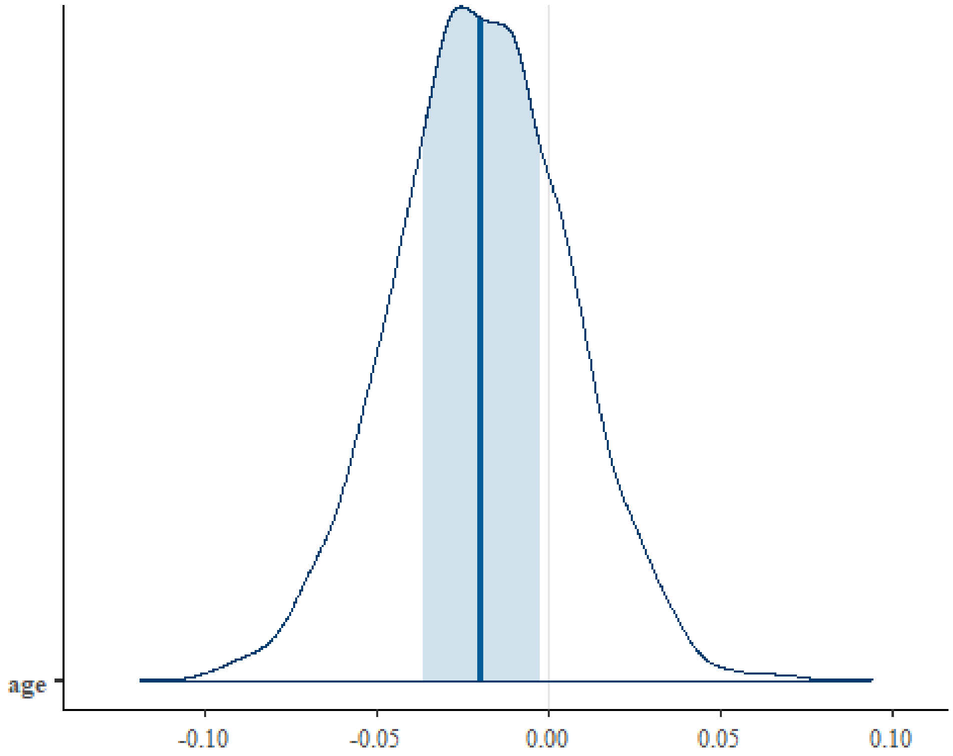
```
mcmc_areas(as.array(post1), pars = c('(Intercept)'))
```



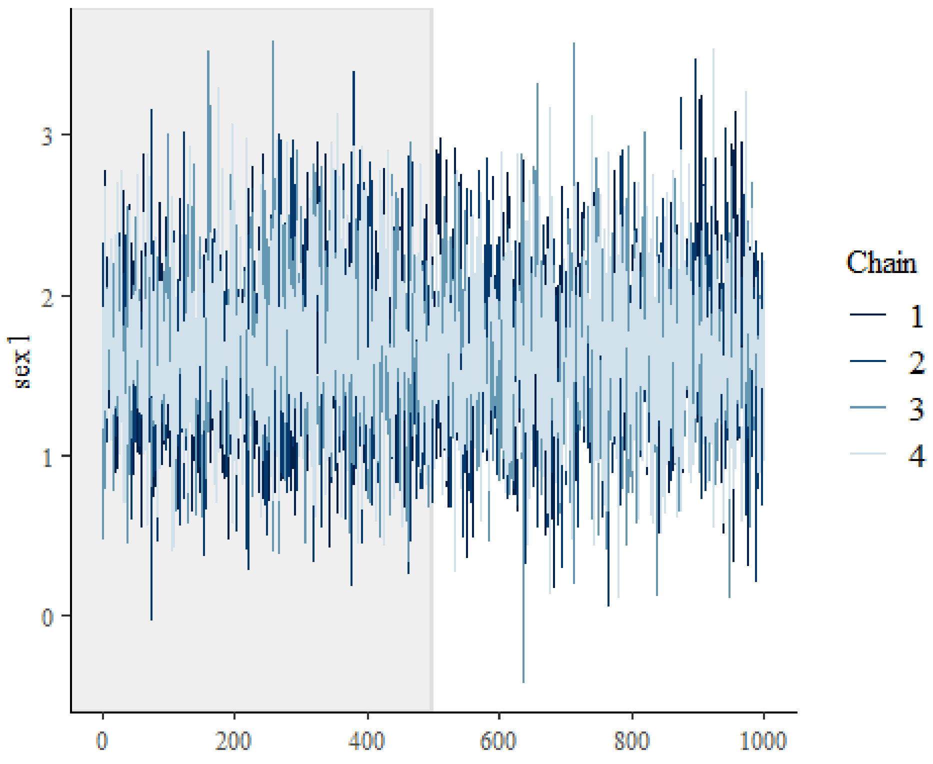
```
mcmc_trace(as.array(post1),  pars = c('age'), n_warmup = 500)
```



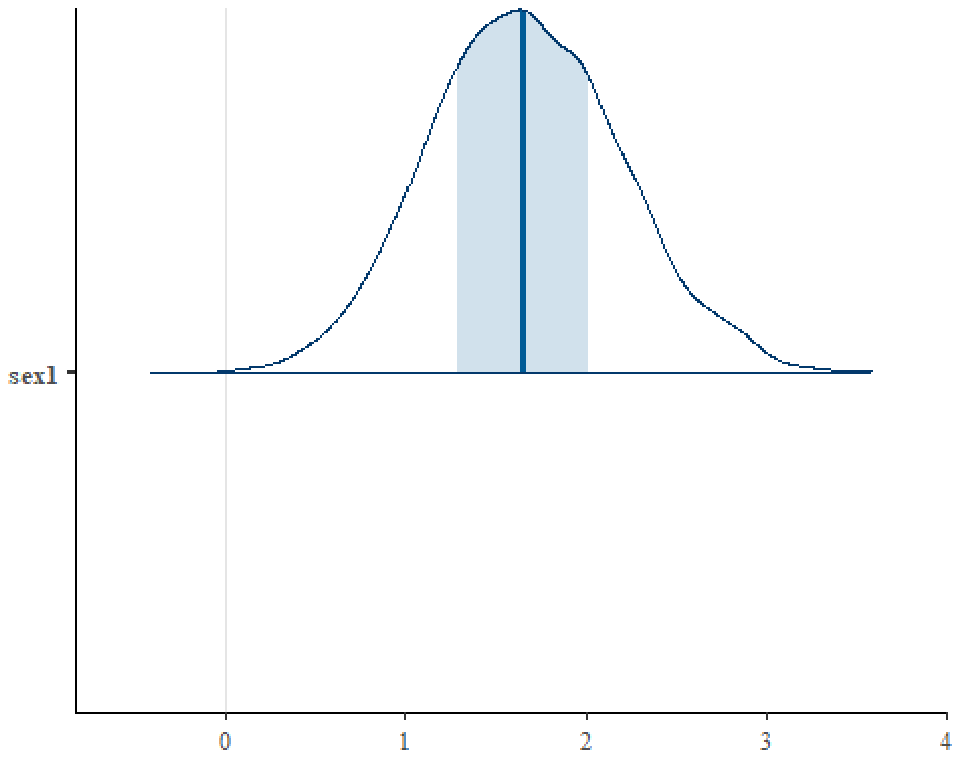
```
mcmc_areas(as.array(post1), pars = c('age'))
```



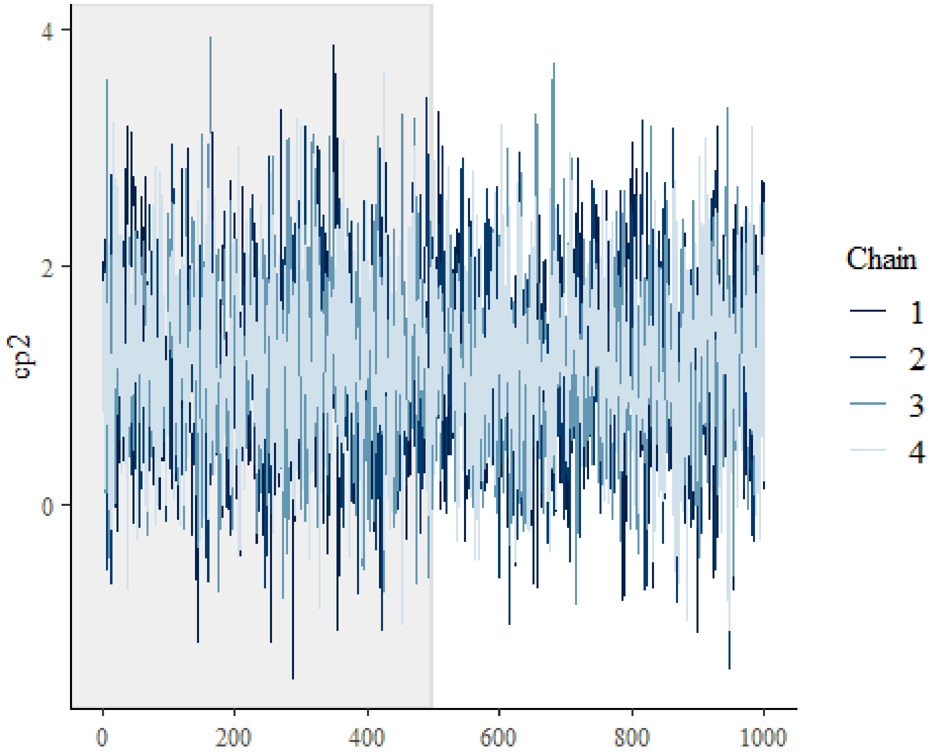
```
mcmc_trace(as.array(post1),  pars = c('sex1'), n_warmup = 500)
```



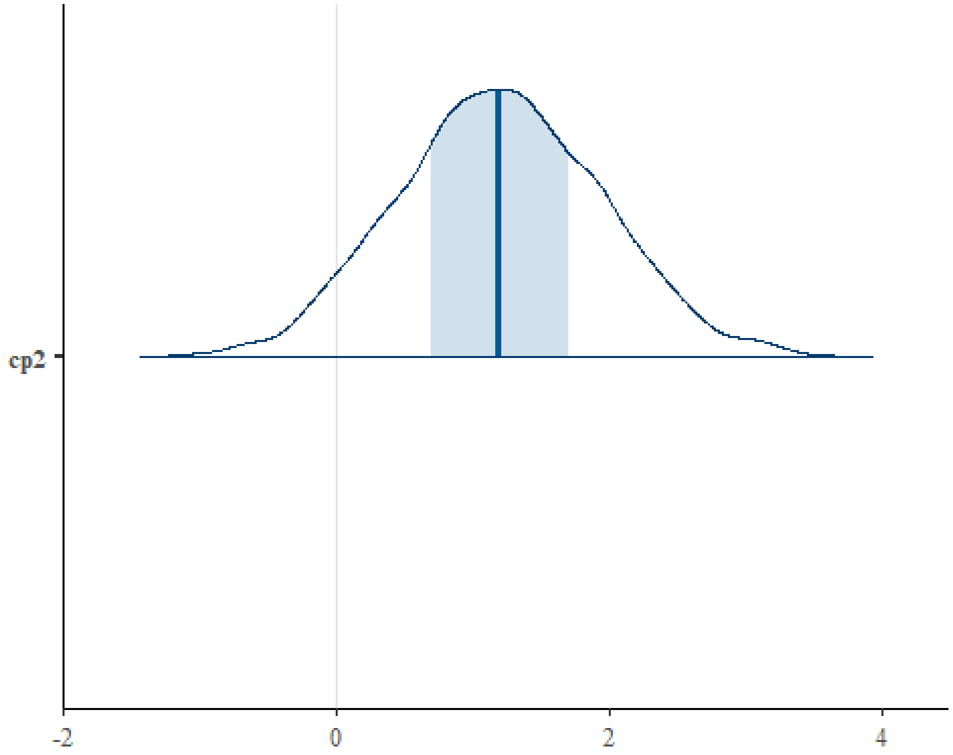
```
mcmc_areas(as.array(post1), pars = c('sex1'))
```



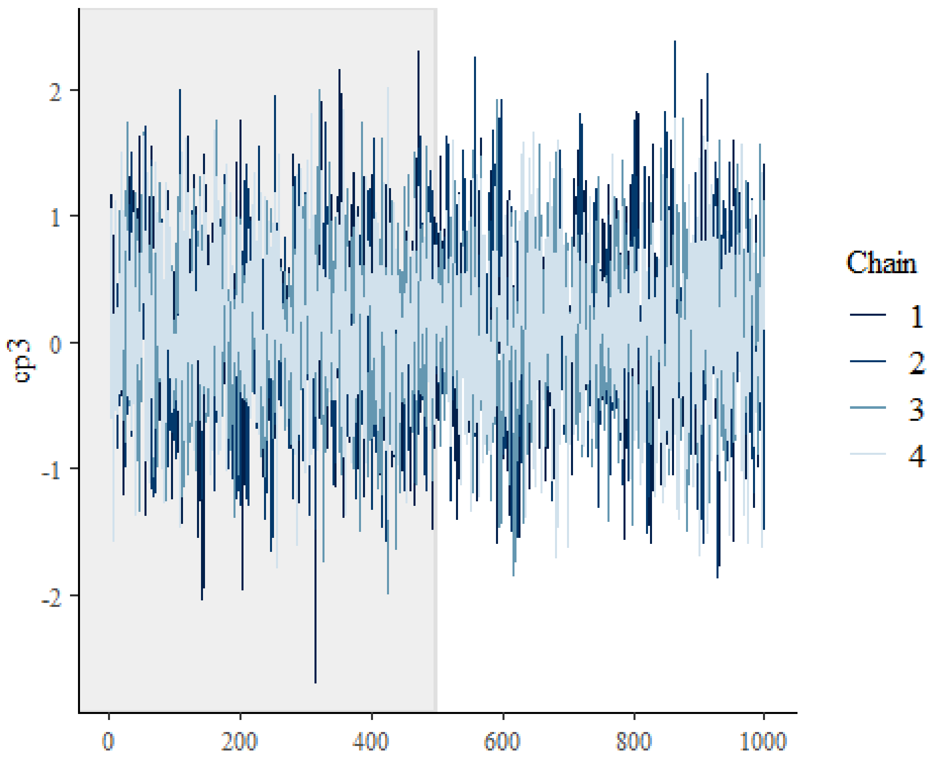
```
mcmc_trace(as.array(post1),  pars = c('cp2'), n_warmup = 500)
```



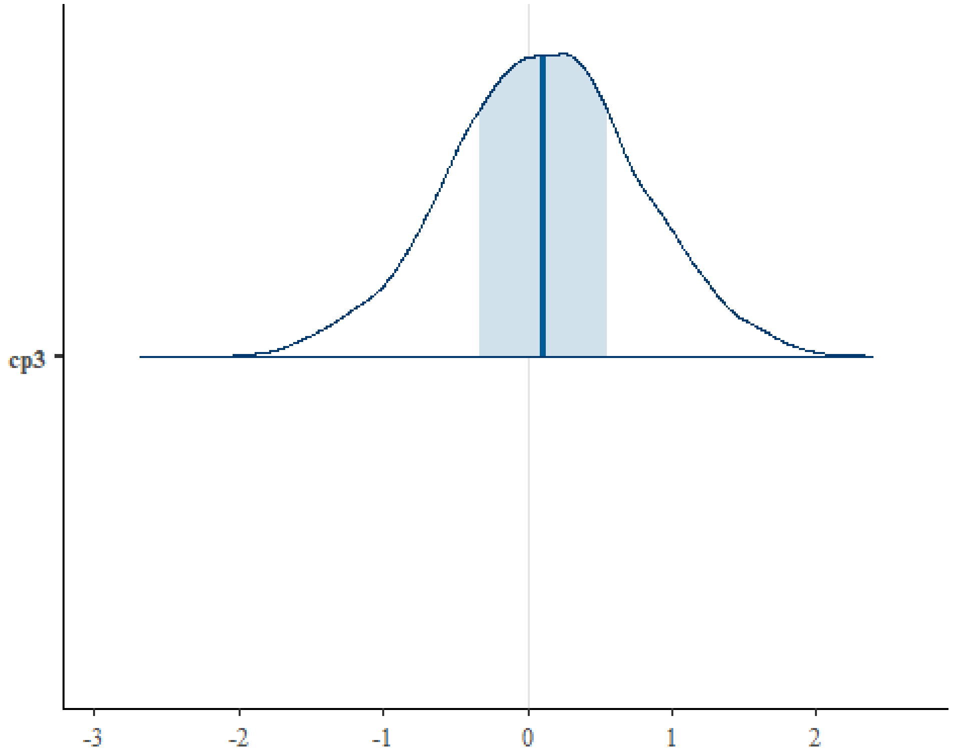
```
mcmc_areas(as.array(post1), pars = c('cp2'))
```



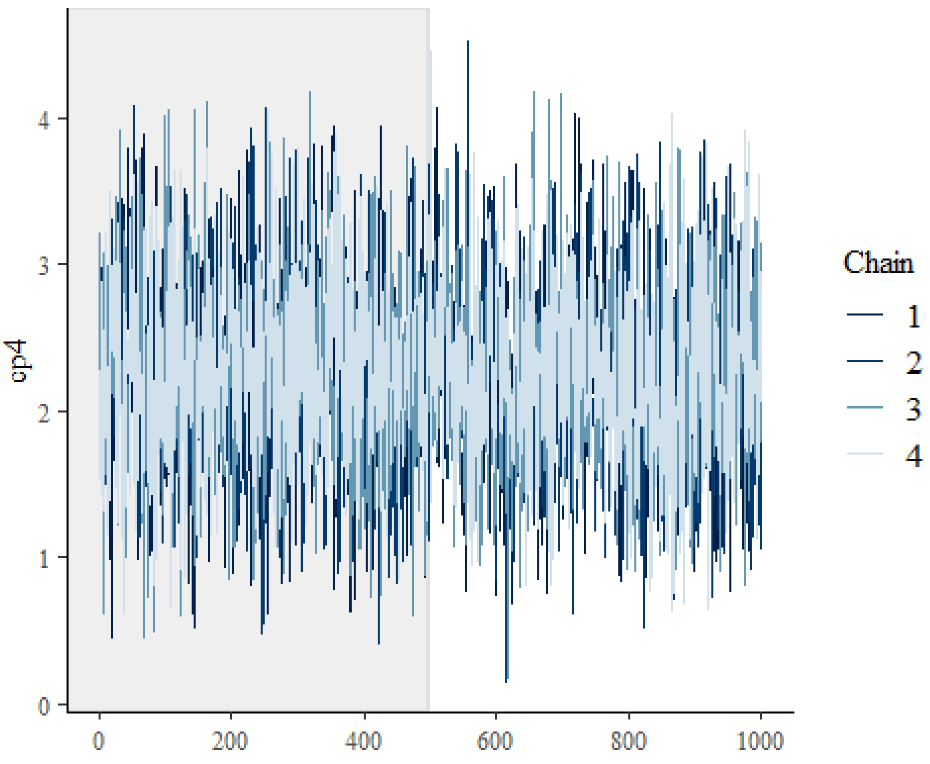
```
mcmc_trace(as.array(post1),  pars = c('cp3'), n_warmup = 500)
```



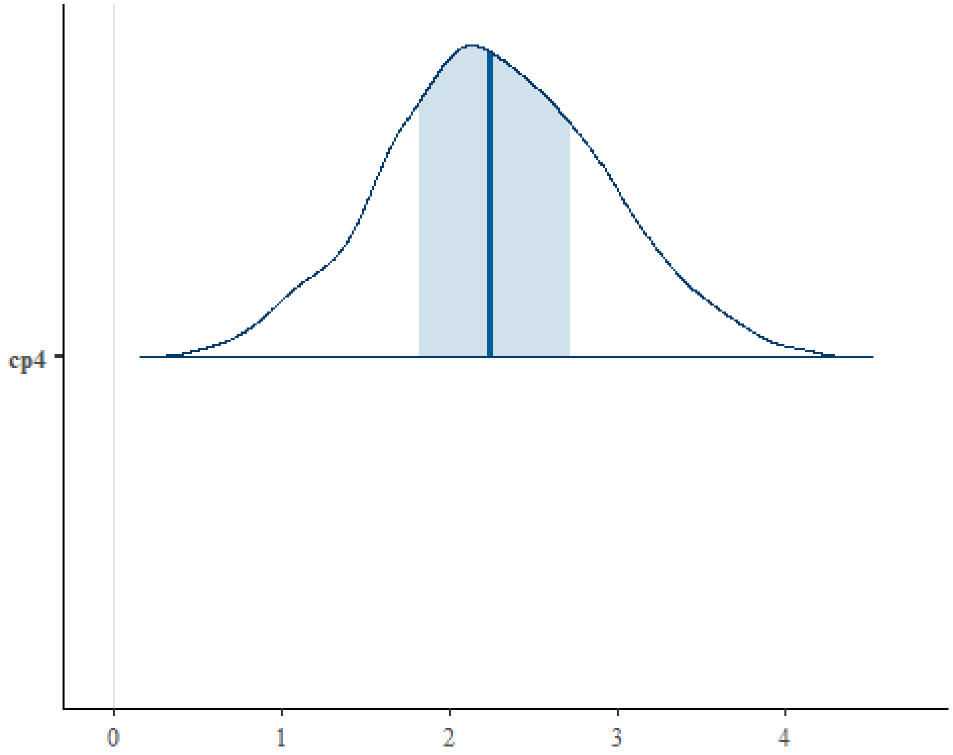
```
mcmc_areas(as.array(post1), pars = c('cp3'))
```



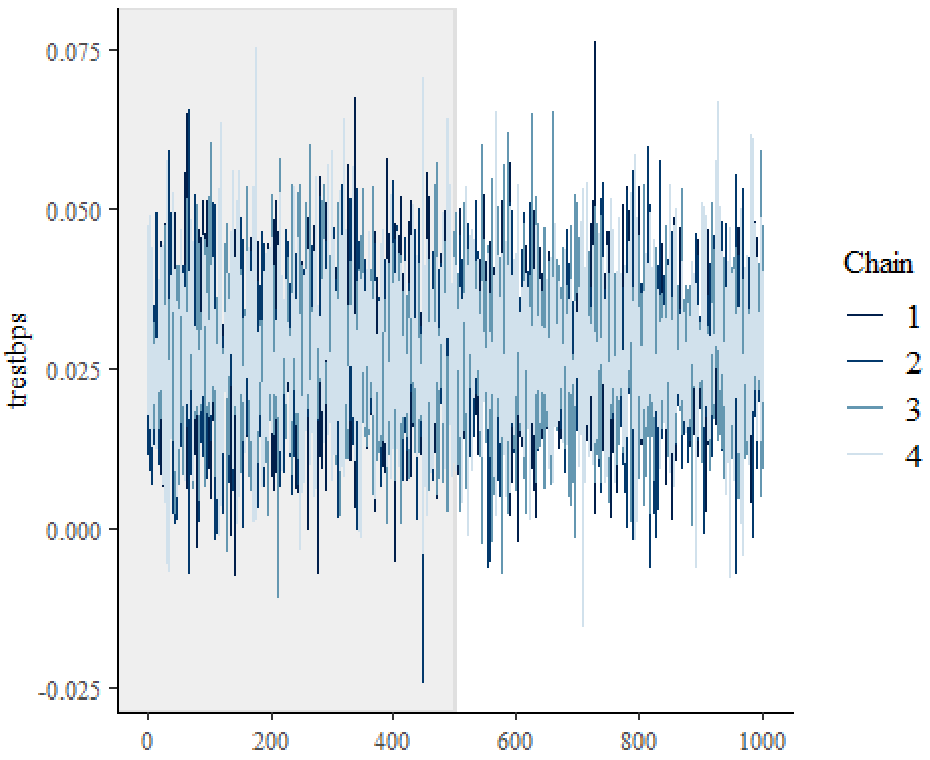
```
mcmc_trace(as.array(post1),  pars = c('cp4'), n_warmup = 500)
```



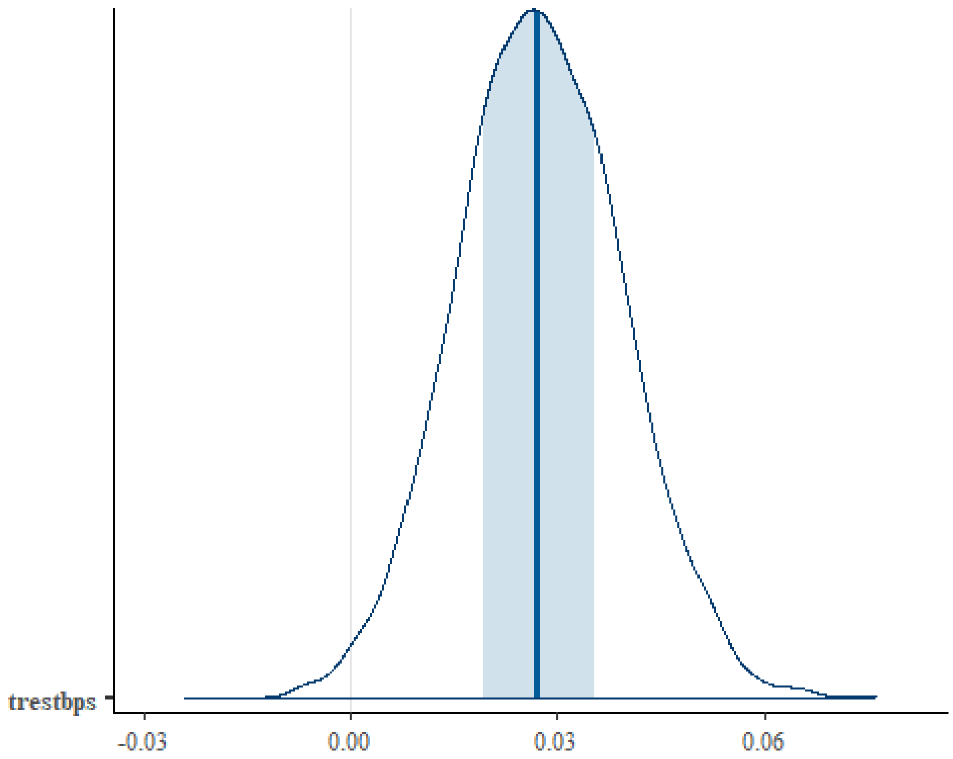
```
mcmc_areas(as.array(post1), pars = c('cp4'))
```



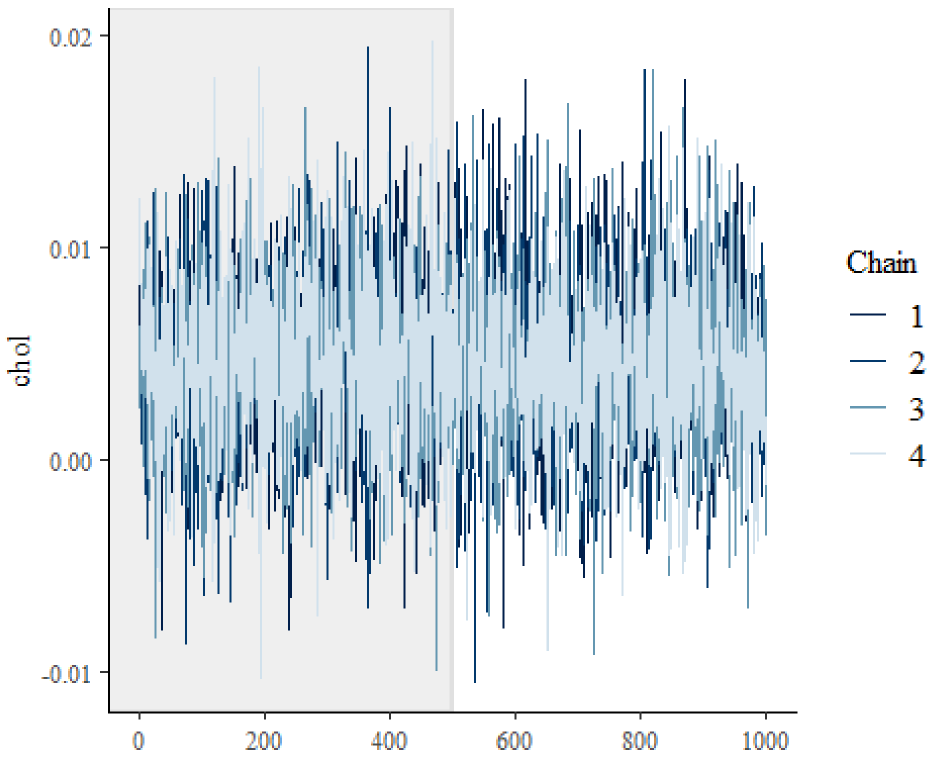
```
mcmc_trace(as.array(post1),  pars = c('trestbps'), n_warmup = 500)
```



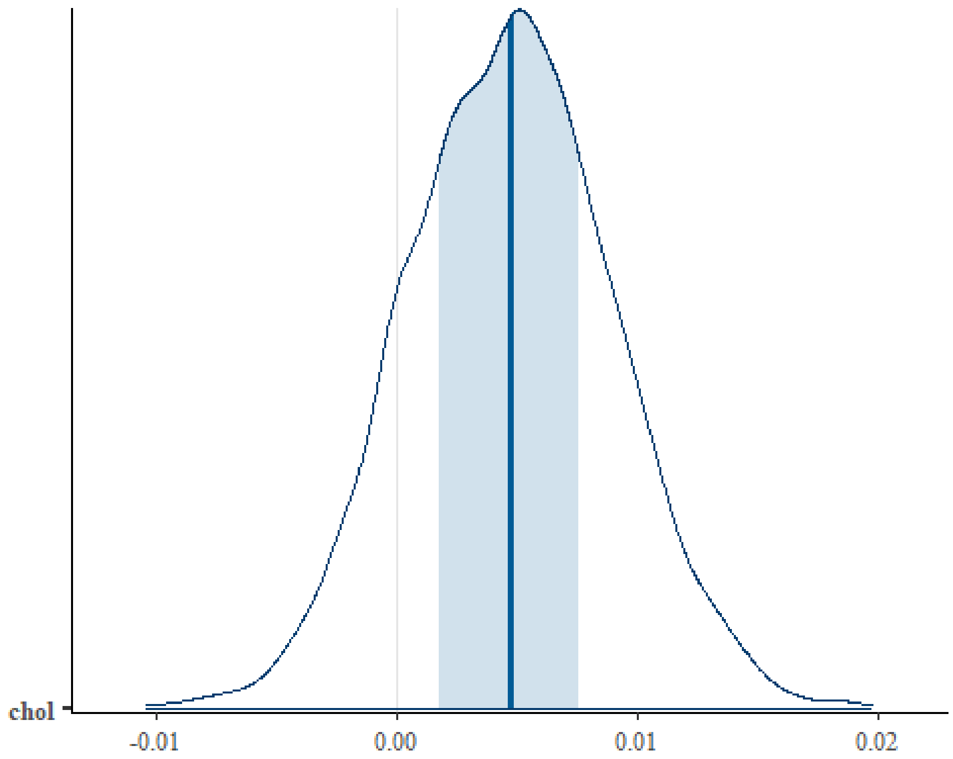
```
mcmc_areas(as.array(post1), pars = c('trestbps'))
```



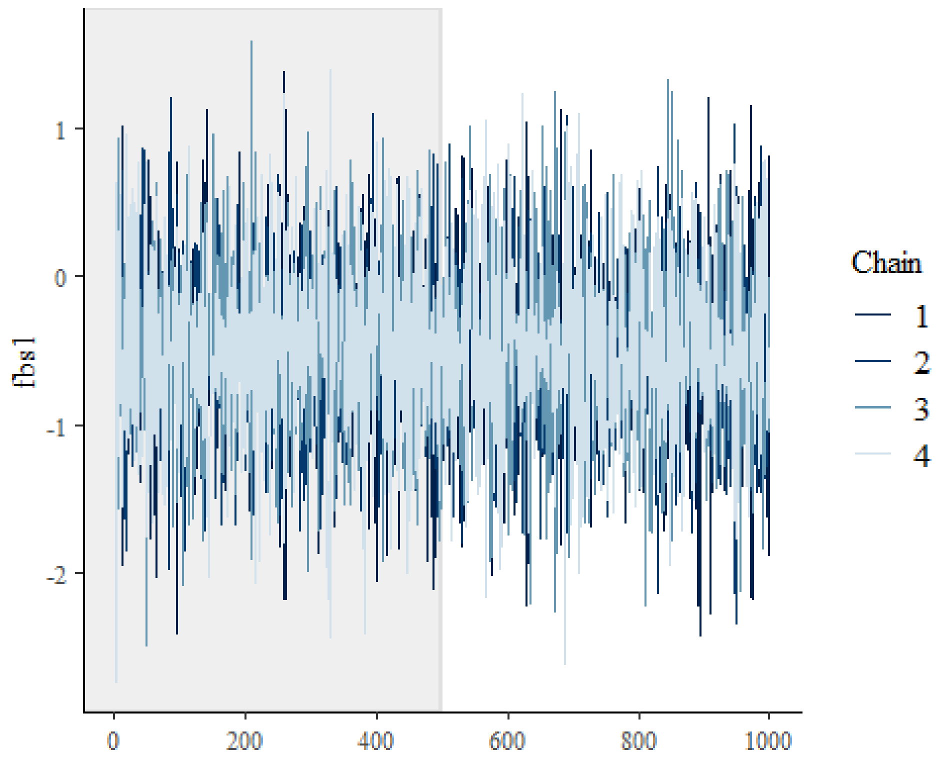
```
mcmc_trace(as.array(post1),  pars = c('chol'), n_warmup = 500)
```



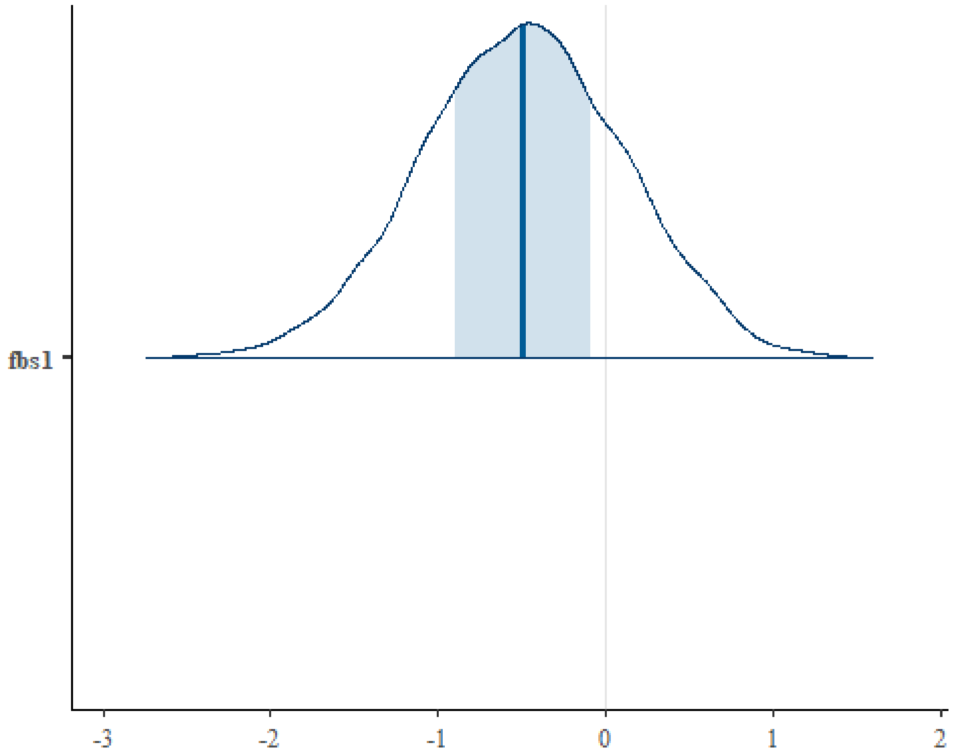
```
mcmc_areas(as.array(post1), pars = c('chol'))
```



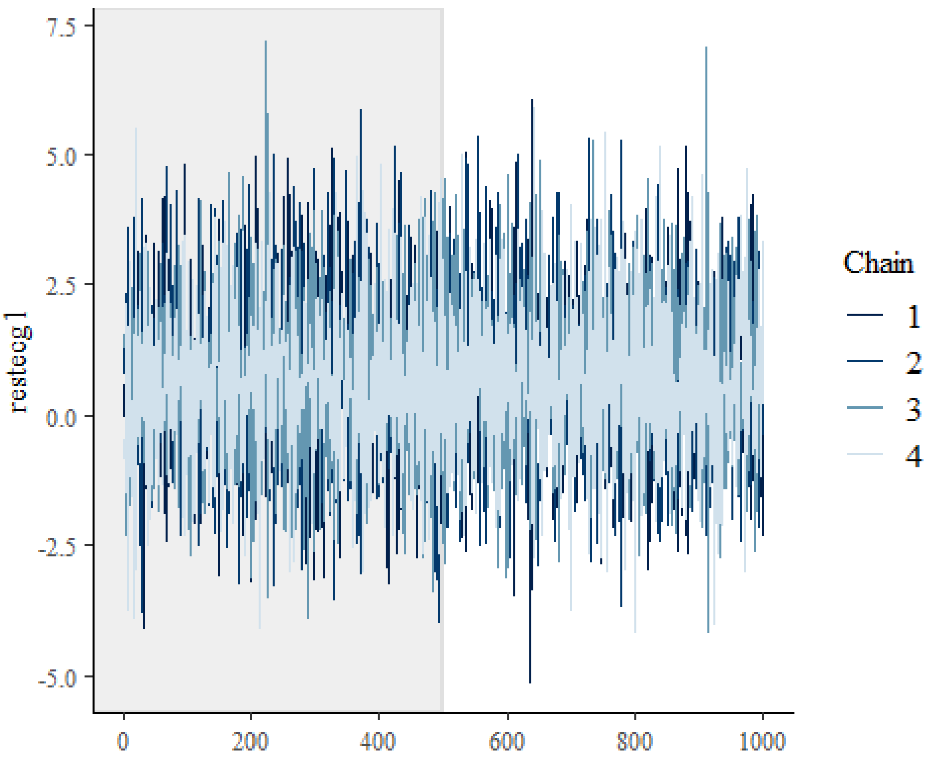
```
mcmc_trace(as.array(post1),  pars = c('fbs1'), n_warmup = 500)
```



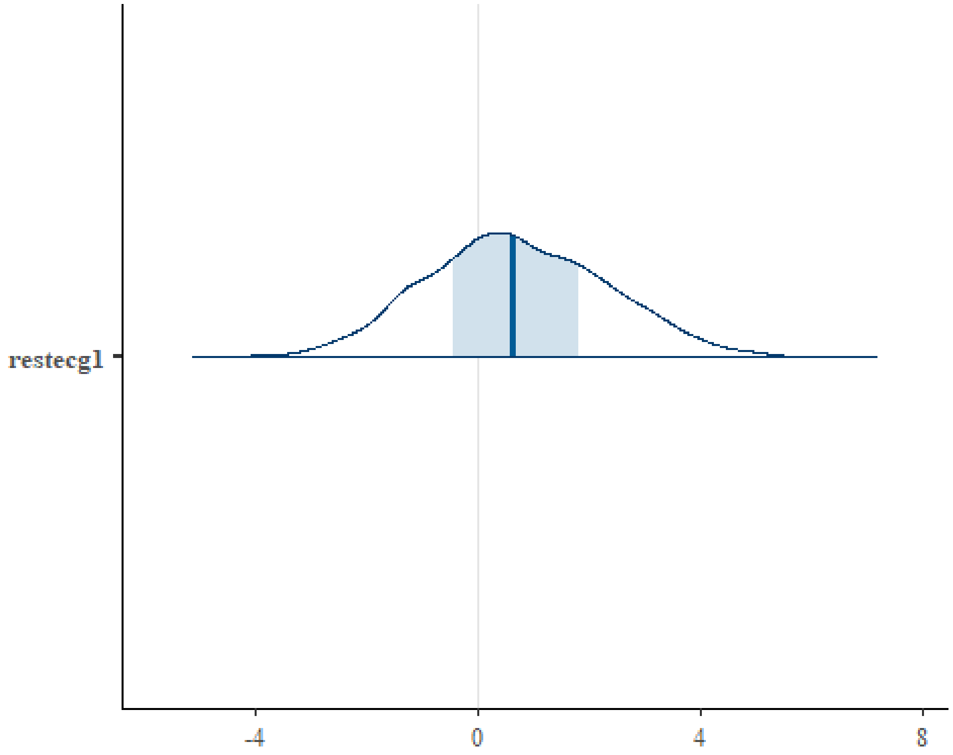
```
mcmc_areas(as.array(post1), pars = c('fbs1'))
```



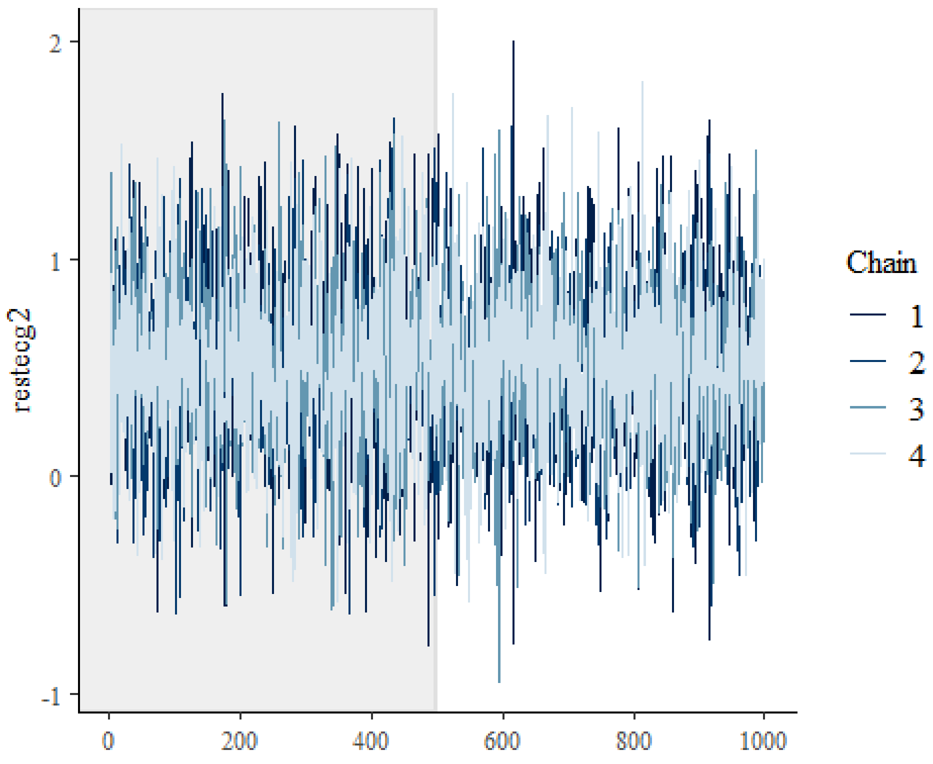
```
mcmc_trace(as.array(post1),  pars = c('restecg1'), n_warmup = 500)
```



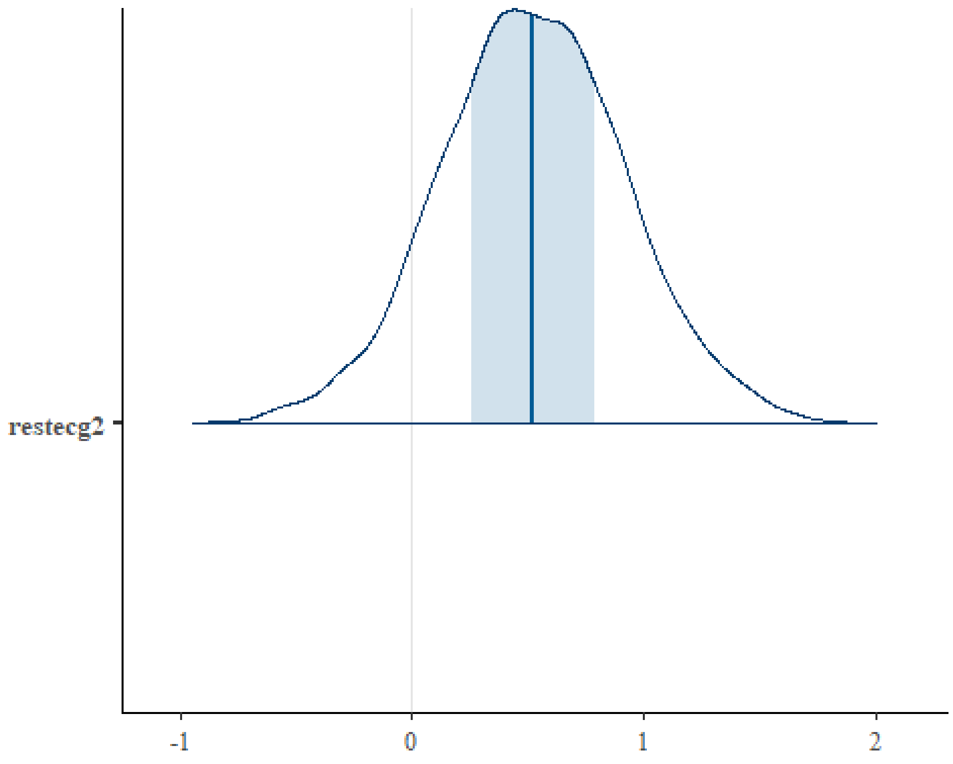
```r
mcmc_areas(as.array(post1), pars = c('restecg1'))
```



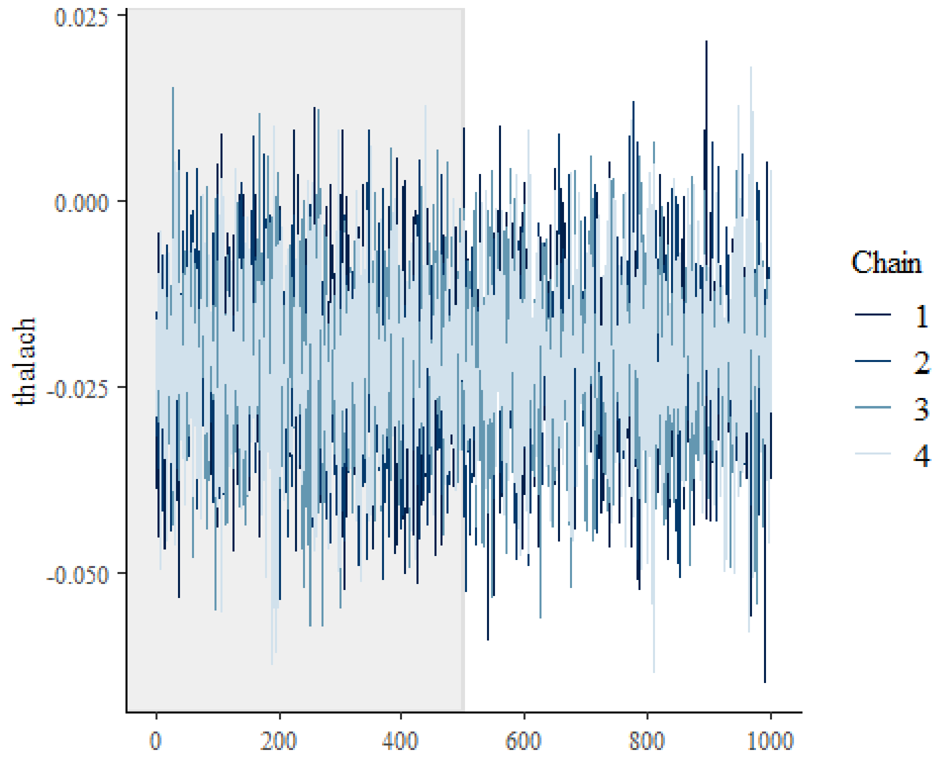
```r
mcmc_trace(as.array(post1),  pars = c('restecg2'), n_warmup = 500)
```



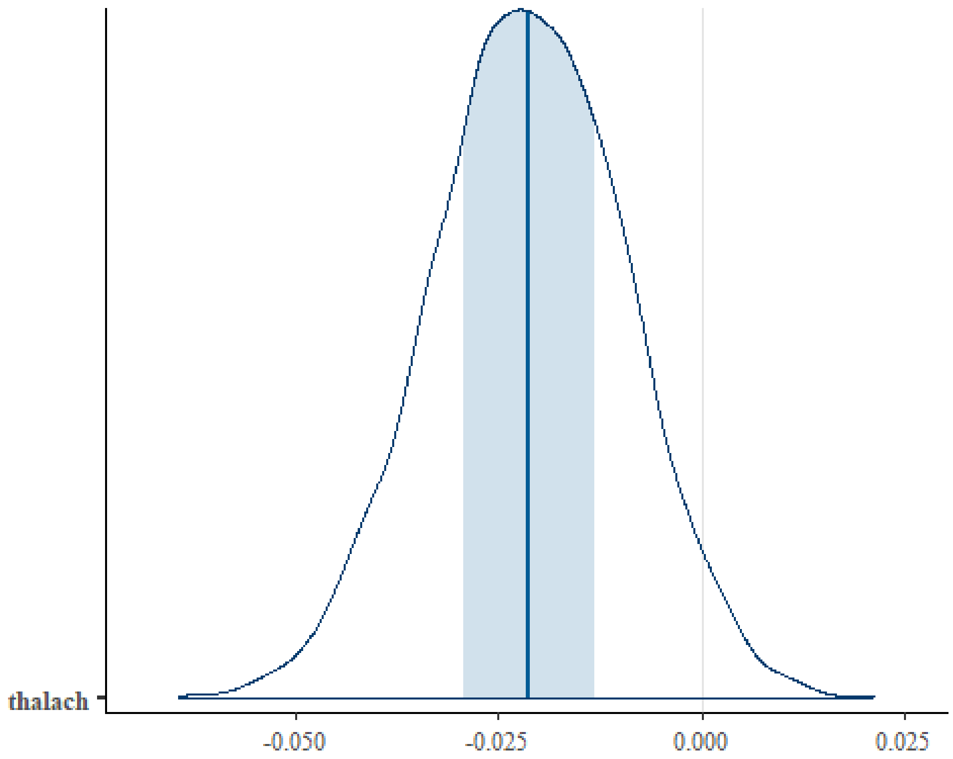
```r
mcmc_areas(as.array(post1), pars = c('restecg2'))
```



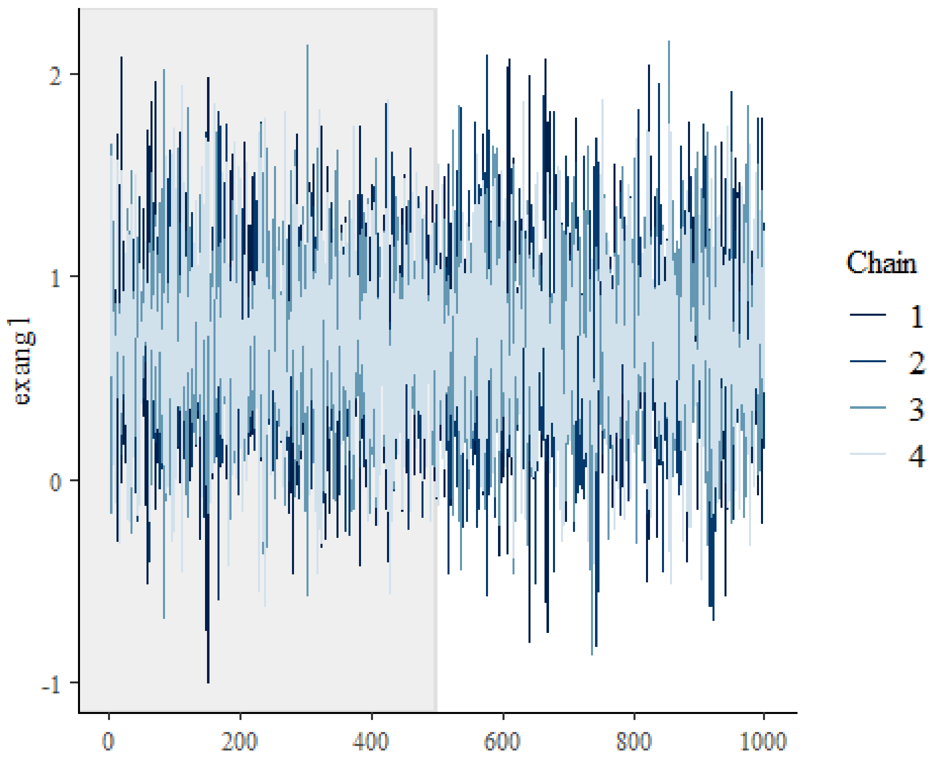
```r
mcmc_trace(as.array(post1),  pars = c('thalach'), n_warmup = 500)
```



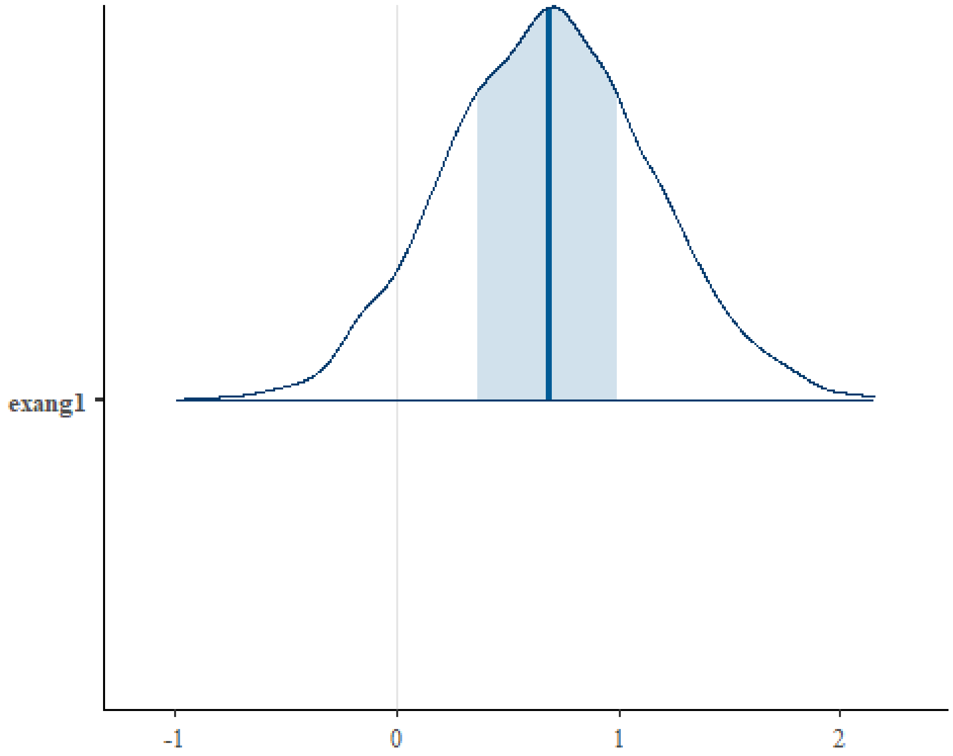
```r
mcmc_areas(as.array(post1), pars = c('thalach'))
```



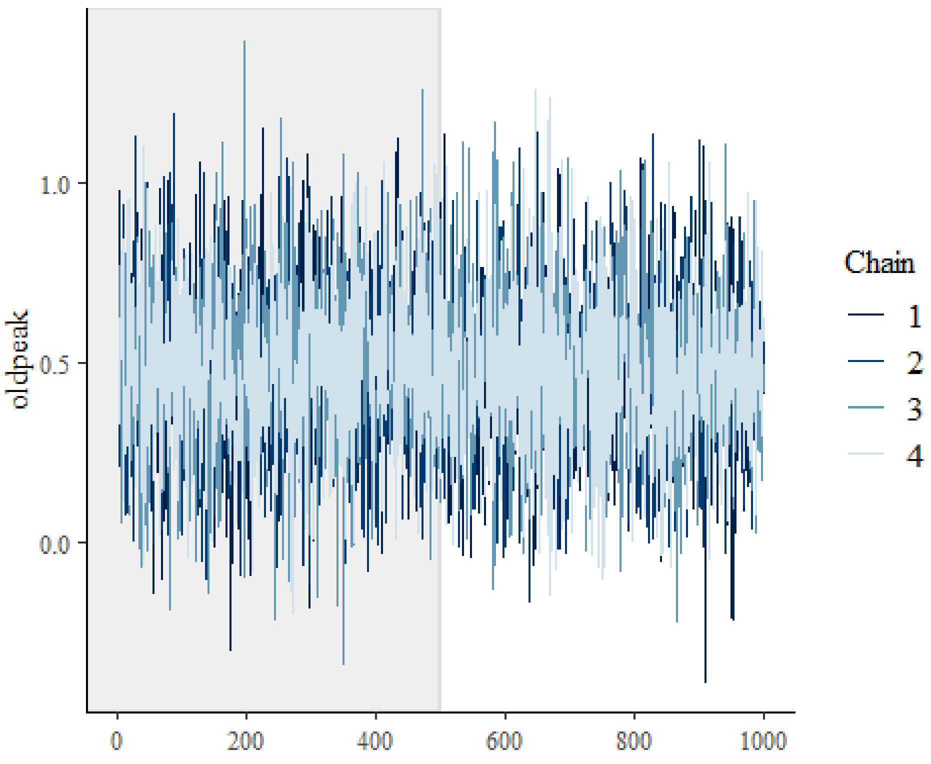
```r
mcmc_trace(as.array(post1),  pars = c('exang1'), n_warmup = 500)
```



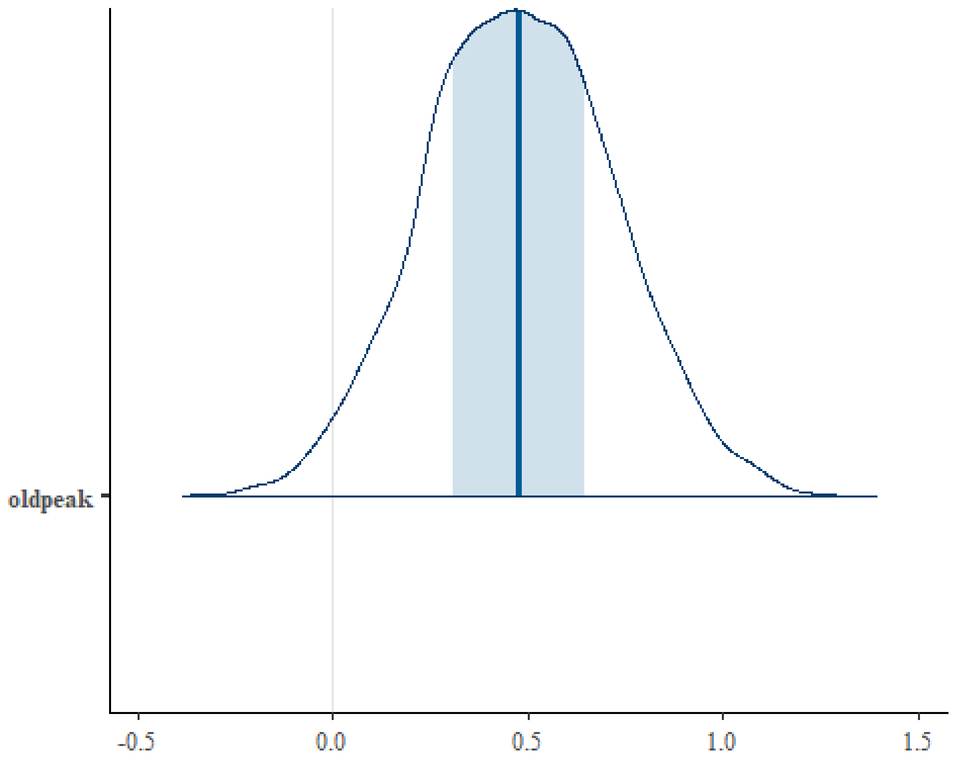
```
mcmc_areas(as.array(post1), pars = c('exang1'))
```



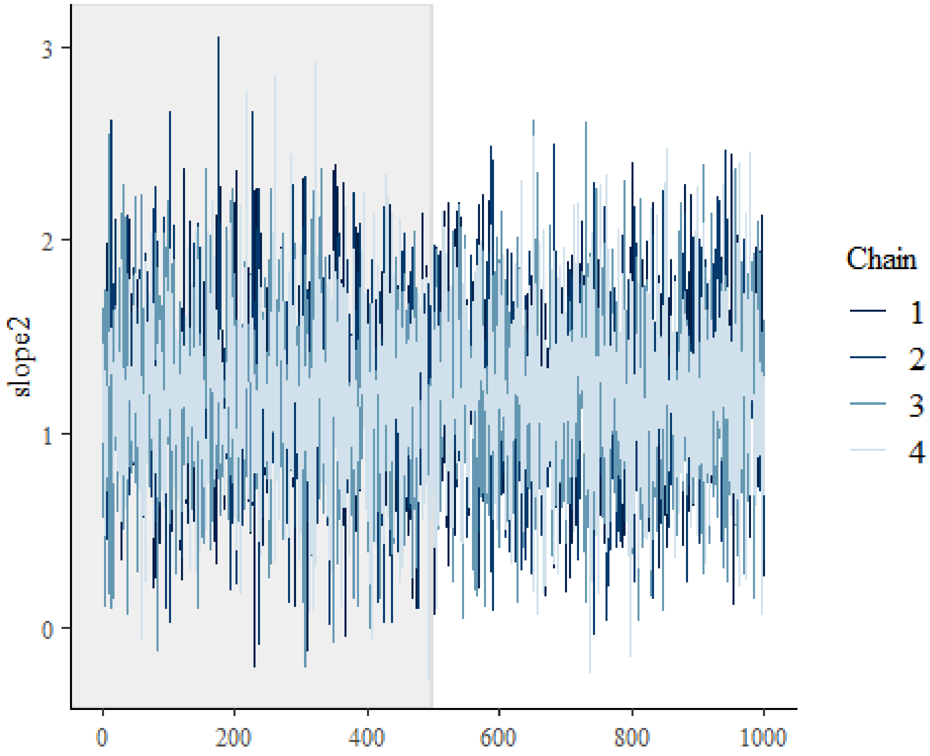
```
mcmc_trace(as.array(post1),  pars = c('oldpeak'), n_warmup = 500)
```



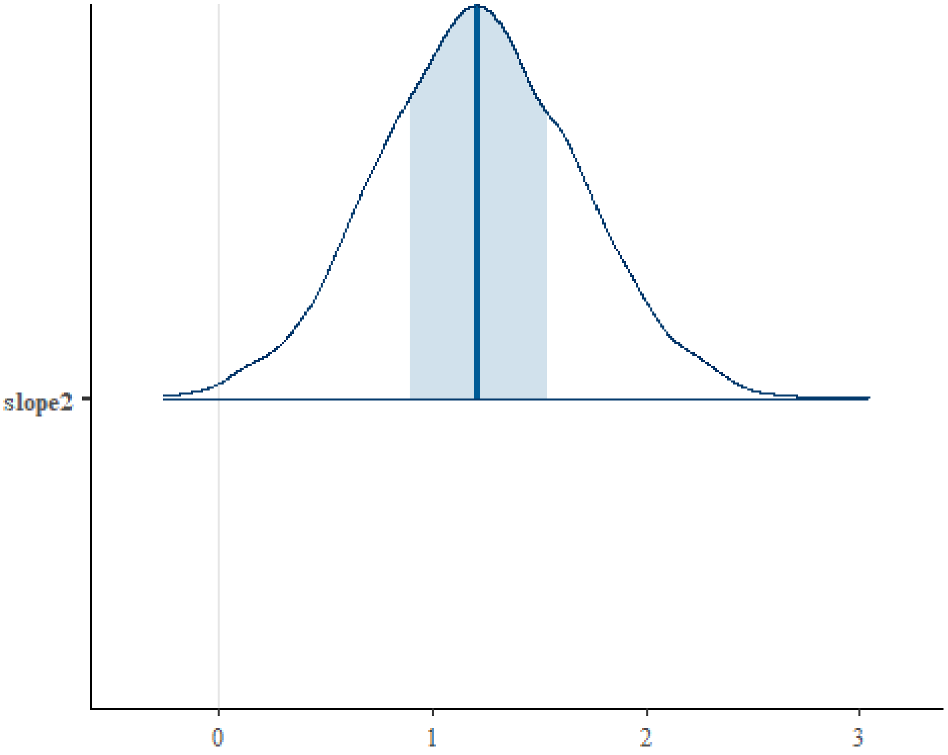
```r
mcmc_areas(as.array(post1), pars = c('oldpeak'))
```



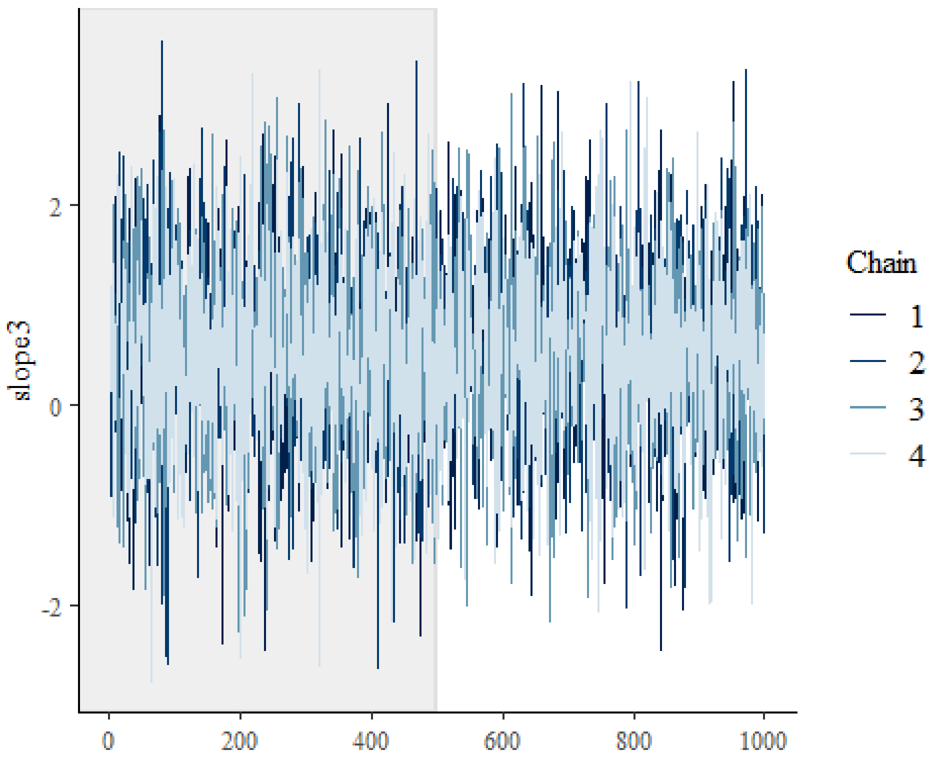
```r
mcmc_trace(as.array(post1),  pars = c('slope2'), n_warmup = 500)
```



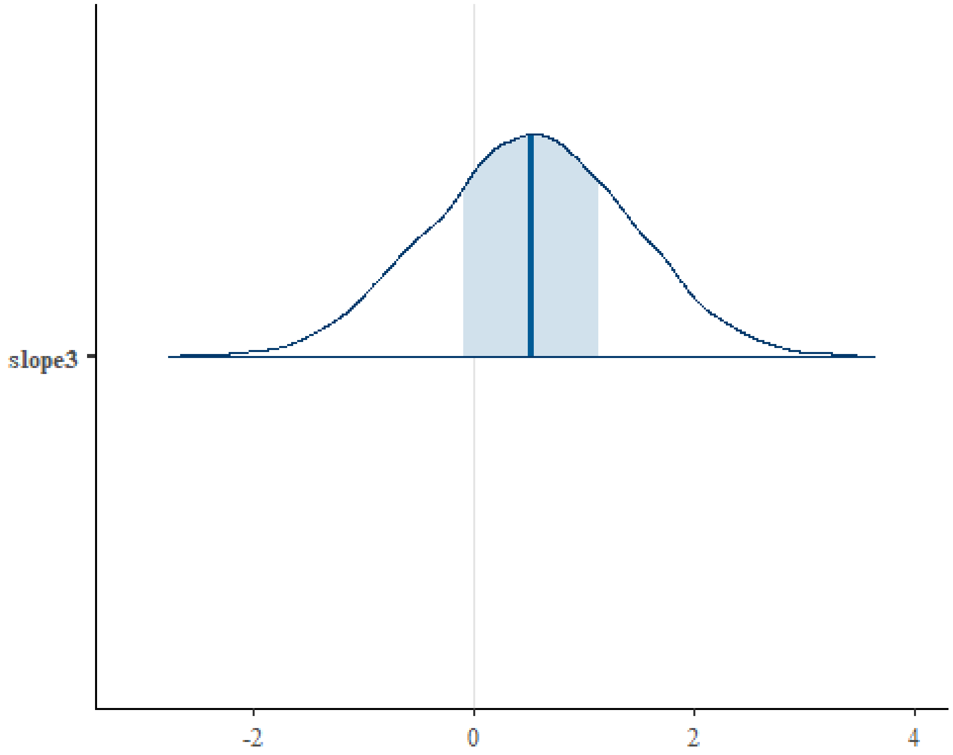
```
mcmc_areas(as.array(post1), pars = c('slope2'))
```



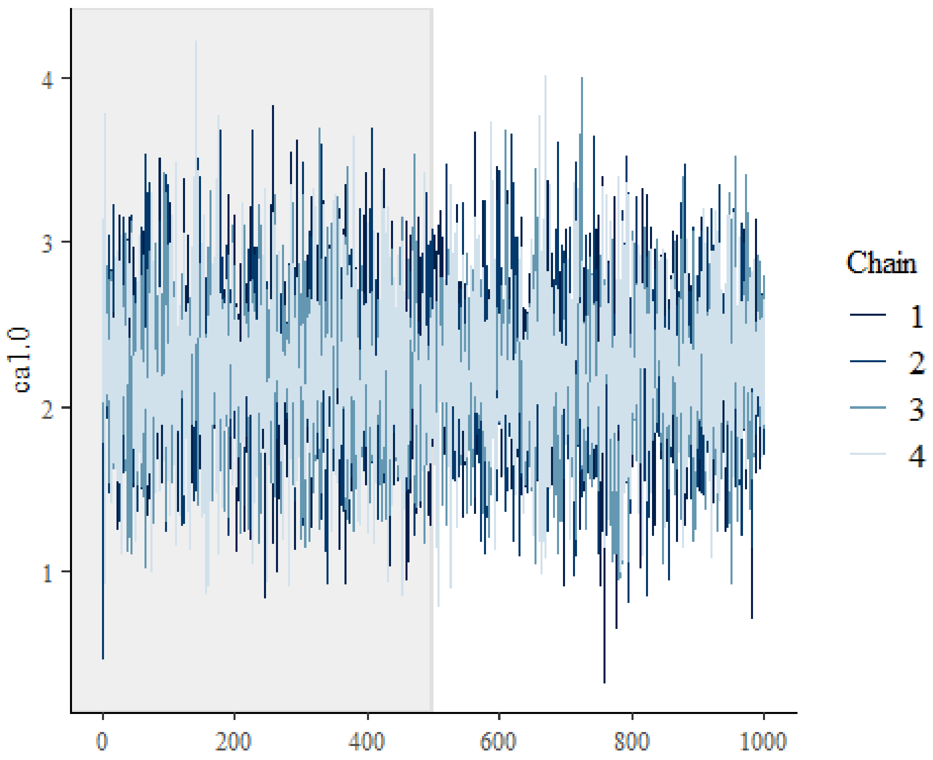
```
mcmc_trace(as.array(post1),  pars = c('slope3'), n_warmup = 500)
```



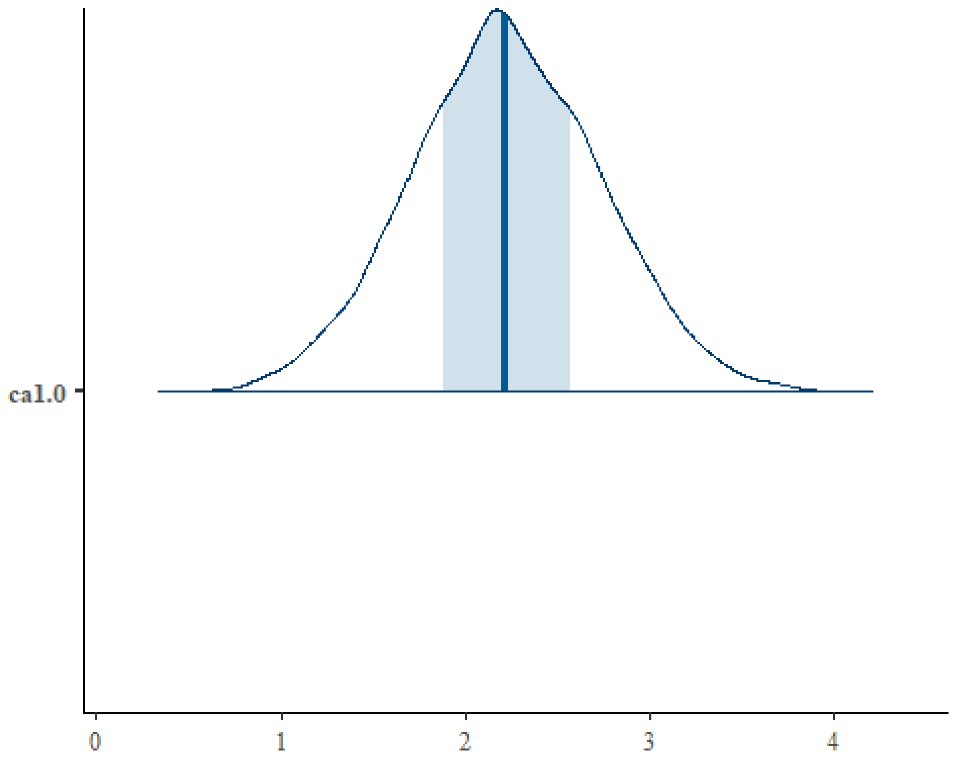
```
mcmc_areas(as.array(post1), pars = c('slope3'))
```



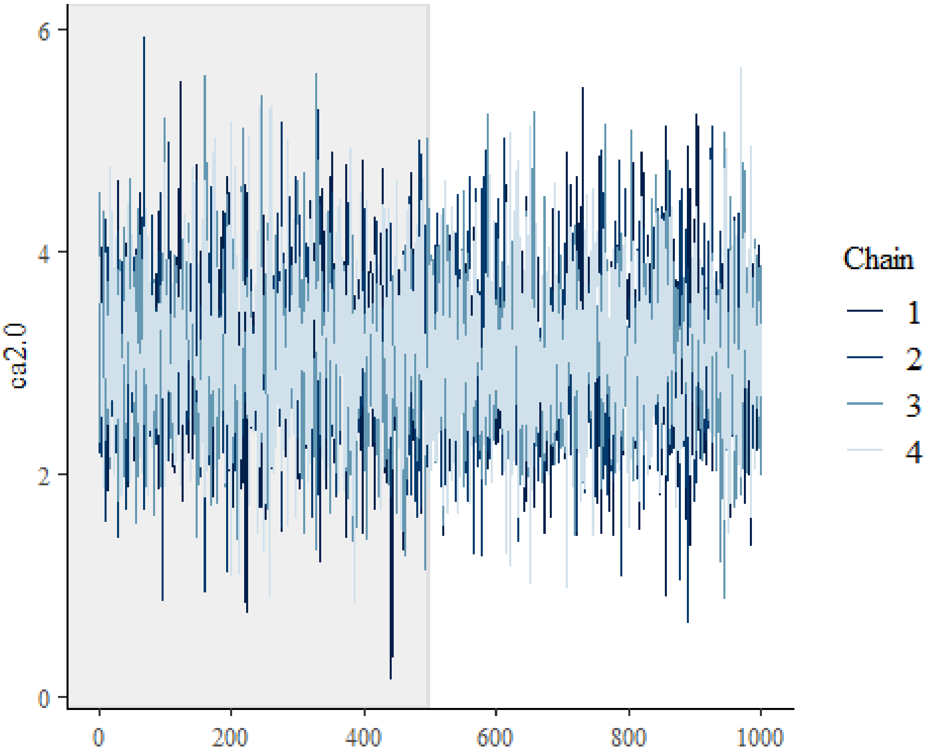
```
mcmc_trace(as.array(post1),  pars = c('ca1.0'), n_warmup = 500)
```



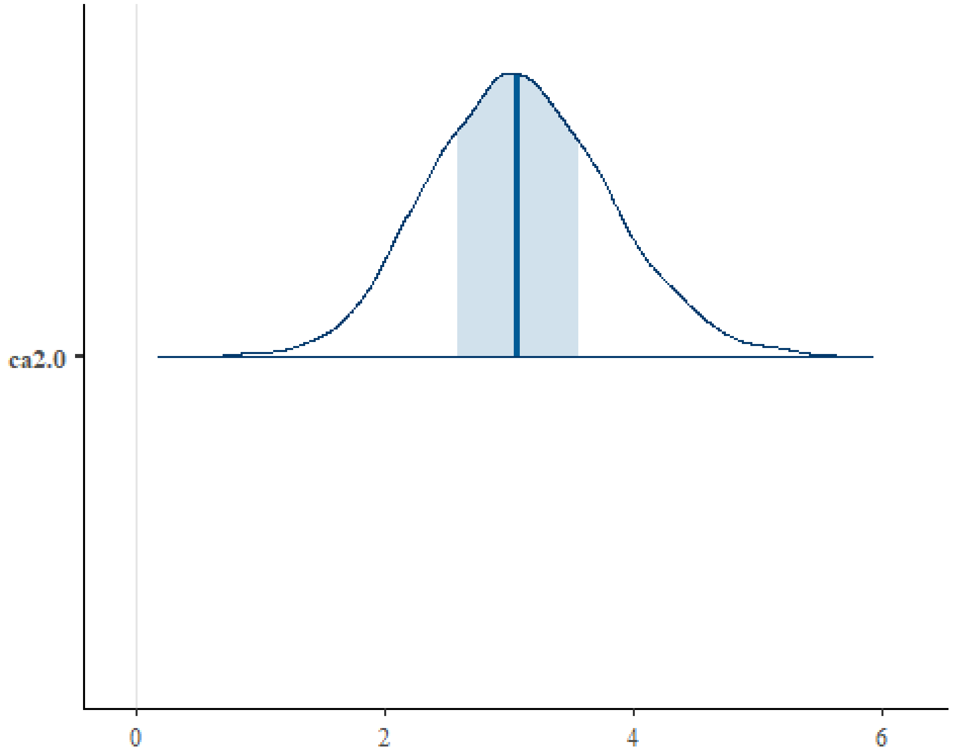
```
mcmc_areas(as.array(post1), pars = c('ca1.0'))
```



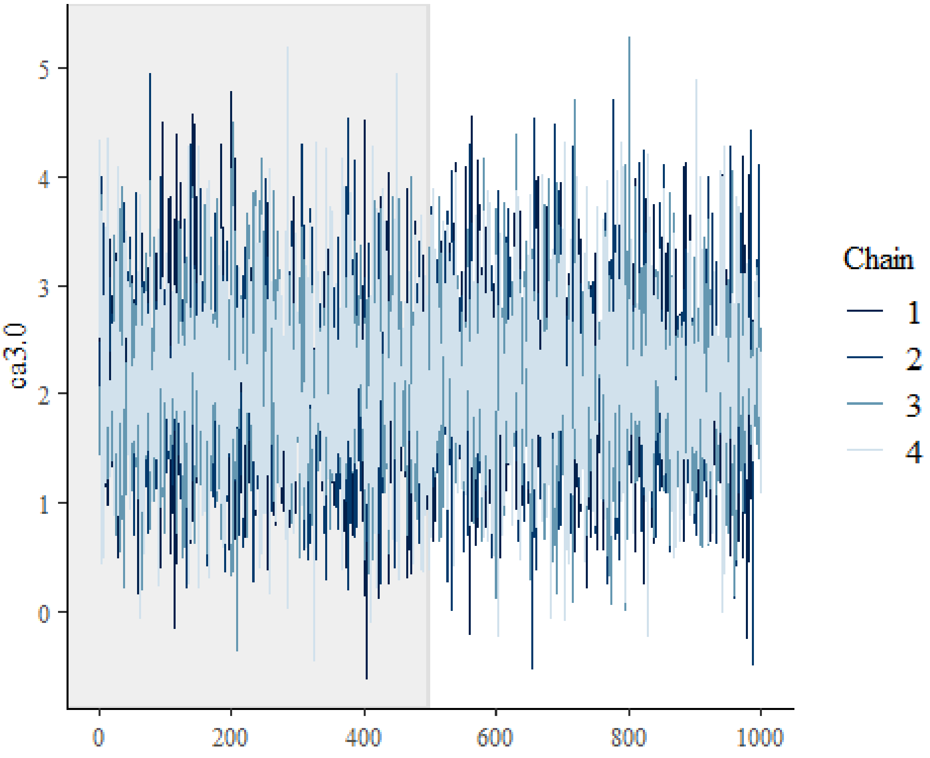
```
mcmc_trace(as.array(post1),  pars = c('ca2.0'), n_warmup = 500)
```



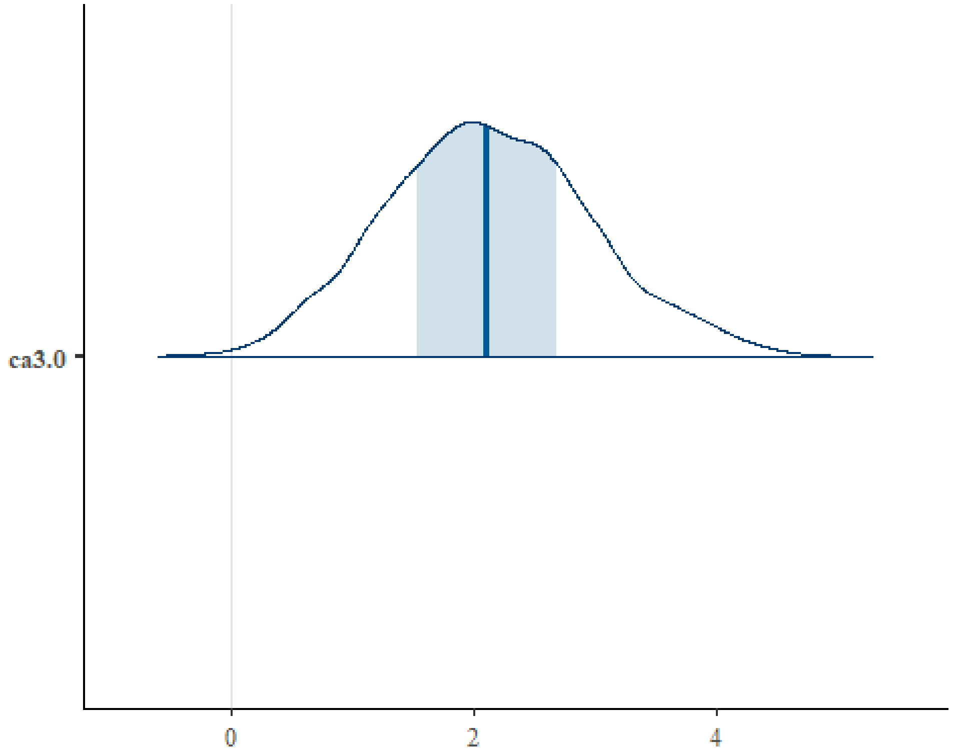
```
mcmc_areas(as.array(post1), pars = c('ca2.0'))
```



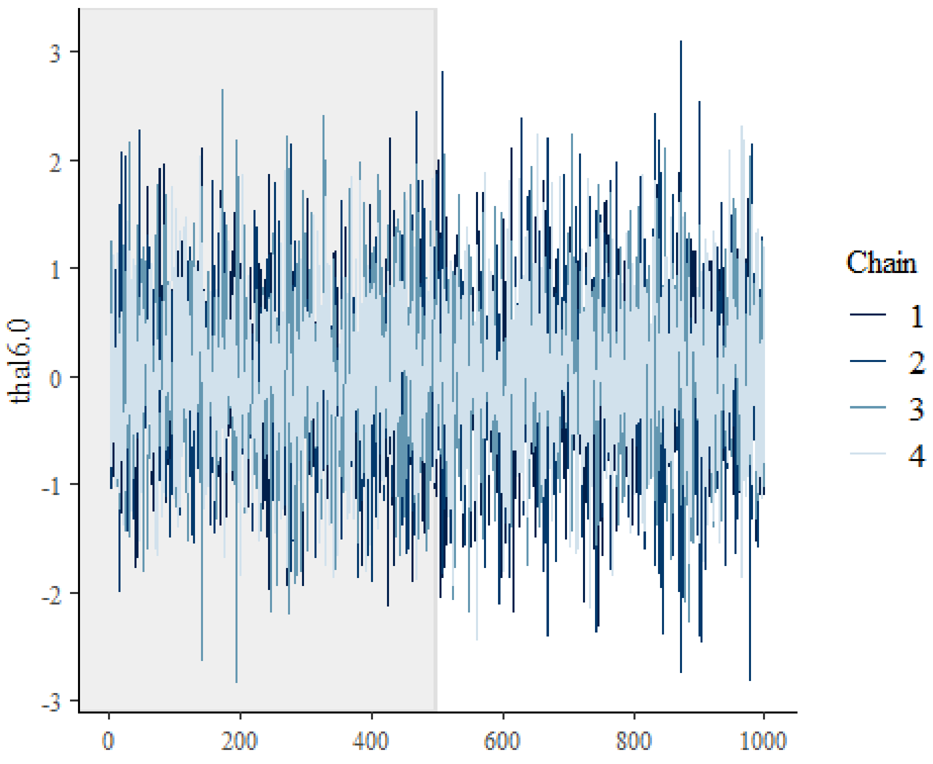
```
mcmc_trace(as.array(post1),  pars = c('ca3.0'), n_warmup = 500)
```



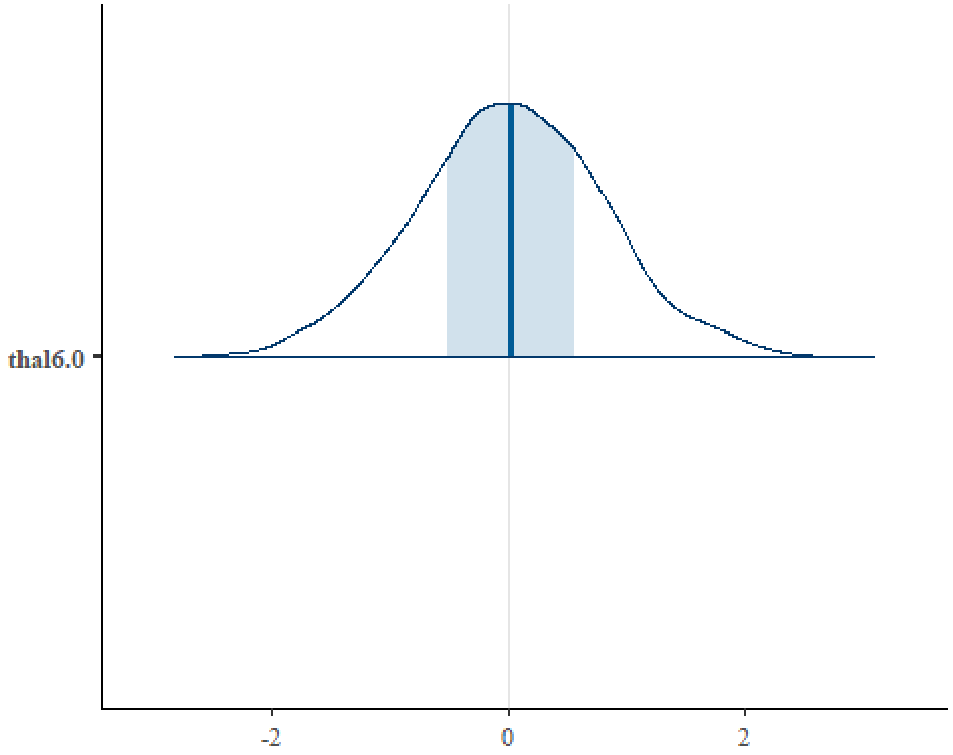
```
mcmc_areas(as.array(post1), pars = c('ca3.0'))
```



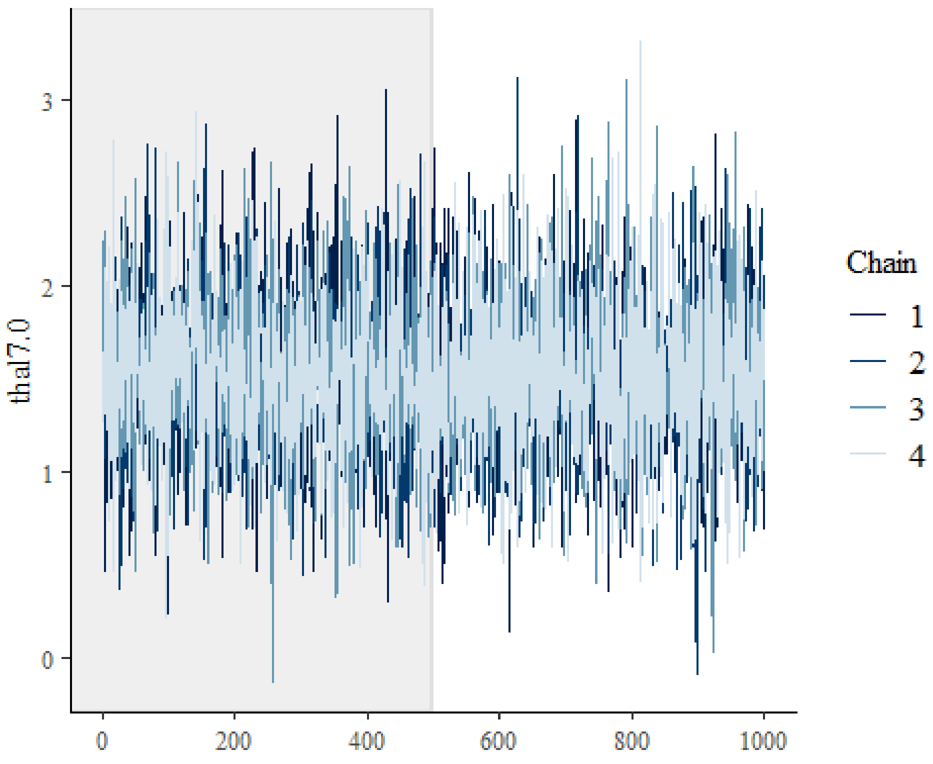
```
mcmc_trace(as.array(post1),  pars = c('thal6.0'), n_warmup = 500)
```



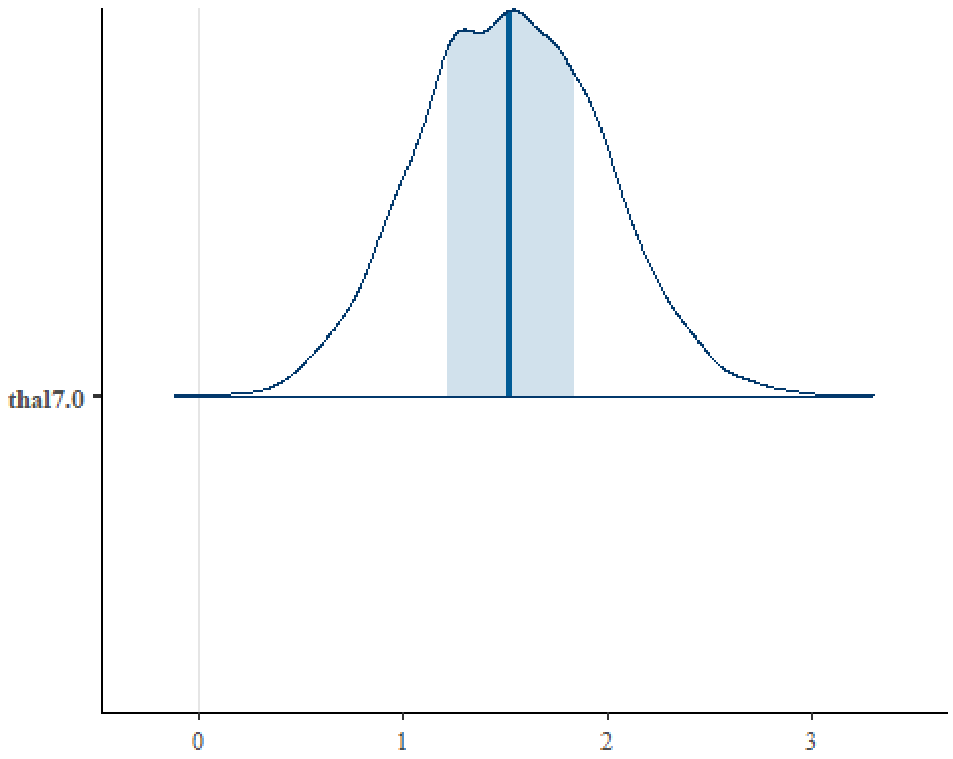
```
mcmc_areas(as.array(post1), pars = c('thal6.0'))
```



```
mcmc_trace(as.array(post1),  pars = c('thal7.0'), n_warmup = 500)
```



```
mcmc_areas(as.array(post1), pars = c('thal7.0'))
```



Analyze with No Cardiovascular disease and Cardiovascular disease with severity = 1

```
# Taken the class with 0 and 1 only
data4 <- data2[data2['class'] == 0 | data2['class'] == 1,]
post2 <- stan_glm(class ~ ., data = data4,
                  family = binomial(link = "logit"),
                  prior = normal_prior, prior_intercept = normal_prior)

##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 10 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: 3.449 seconds (Warm-up)
## Chain 1:                2.514 seconds (Sampling)
## Chain 1:                5.963 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.96 seconds (Warm-up)
## Chain 2:                3.181 seconds (Sampling)
## Chain 2:                6.141 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.824 seconds (Warm-up)
## Chain 3:                2.058 seconds (Sampling)
## Chain 3:                4.882 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.764 seconds (Warm-up)
## Chain 4:                2.022 seconds (Sampling)
## Chain 4:                4.786 seconds (Total)
## Chain 4:
```

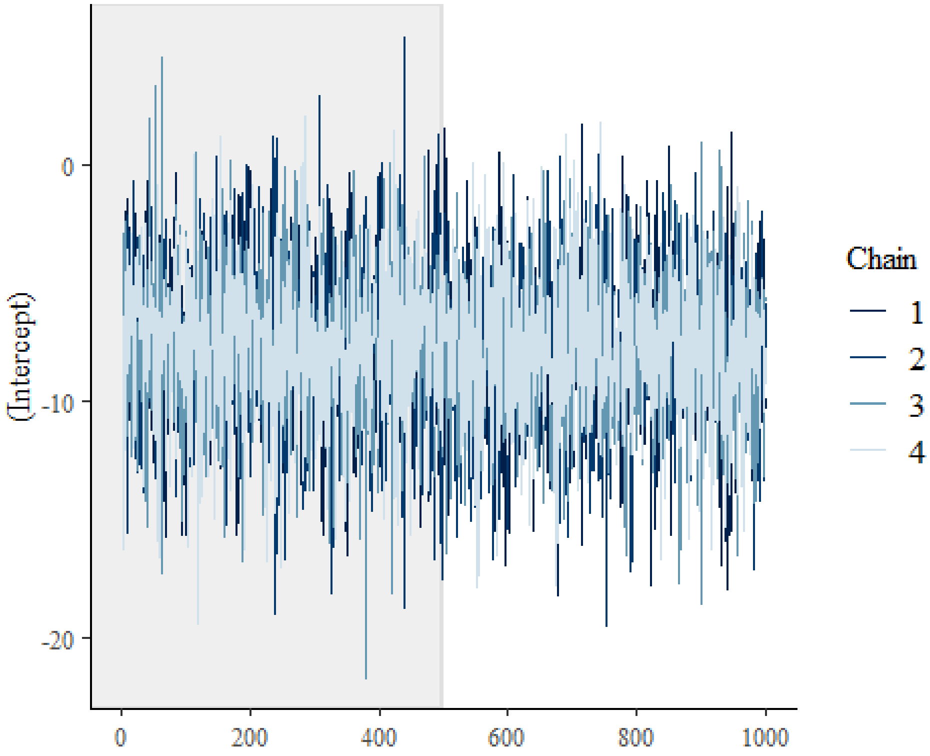
```
round(coef(post2), 2)
```

```
## (Intercept)         age        sex1         cp2         cp3         cp4 
##       -7.75        0.00        1.70        0.63       -0.07        1.78 
##    trestbps        chol        fbs1    restecg1    restecg2     thalach 
##        0.03        0.00       -1.48       -0.55        0.54       -0.01 
##      exang1     oldpeak      slope2      slope3       ca1.0       ca2.0 
##        0.50        0.18        1.10        0.20        1.72        2.02 
##       ca3.0     thal6.0     thal7.0 
##        1.11       -1.04        1.16
```

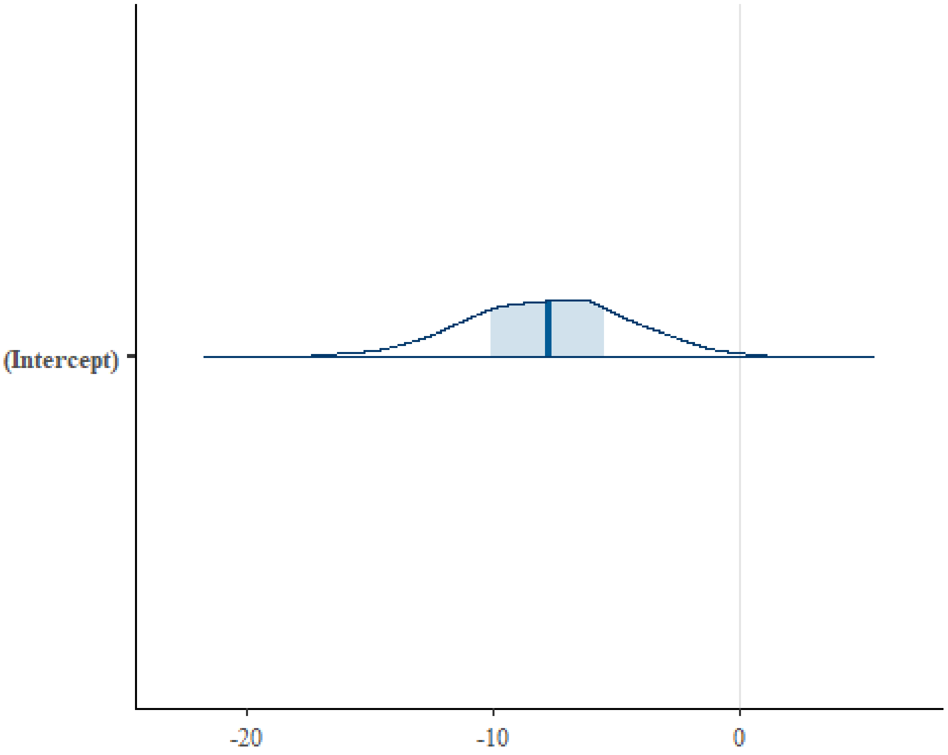
```
round(posterior_interval(post2, prob = 0.95), 2)
```

```
##               2.5% 97.5%
## (Intercept) -14.89 -1.29
## age          -0.06  0.05
## sex1          0.56  2.93
## cp2          -0.97  2.22
## cp3          -1.57  1.36
## cp4           0.47  3.20
## trestbps      0.00  0.05
## chol         -0.01  0.01
## fbs1         -3.16 -0.08
## restecg1     -5.05  3.63
## restecg2     -0.32  1.42
## thalach      -0.04  0.02
## exang1       -0.56  1.51
## oldpeak      -0.38  0.76
## slope2        0.07  2.15
## slope3       -2.23  2.28
## ca1.0         0.60  2.85
## ca2.0         0.34  3.77
## ca3.0        -1.29  3.41
## thal6.0      -3.21  0.91
## thal7.0       0.18  2.16
```

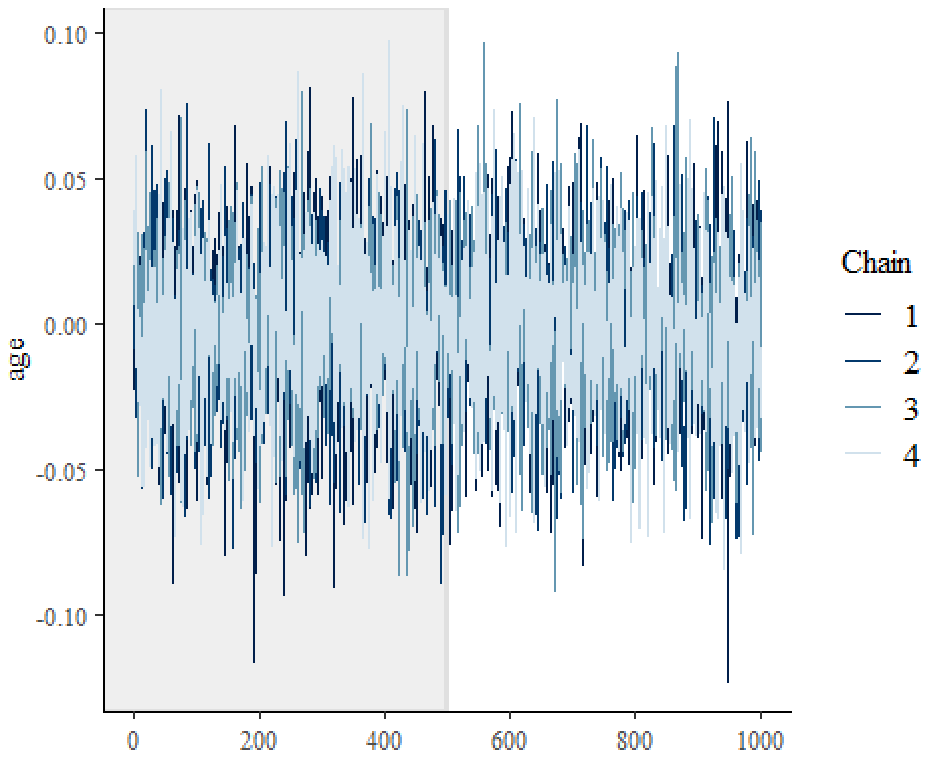
```
post1 <- post2  # to make reuse of the code of plotting
mcmc_trace(as.array(post1),  pars = c('(Intercept)'), n_warmup = 500)
```



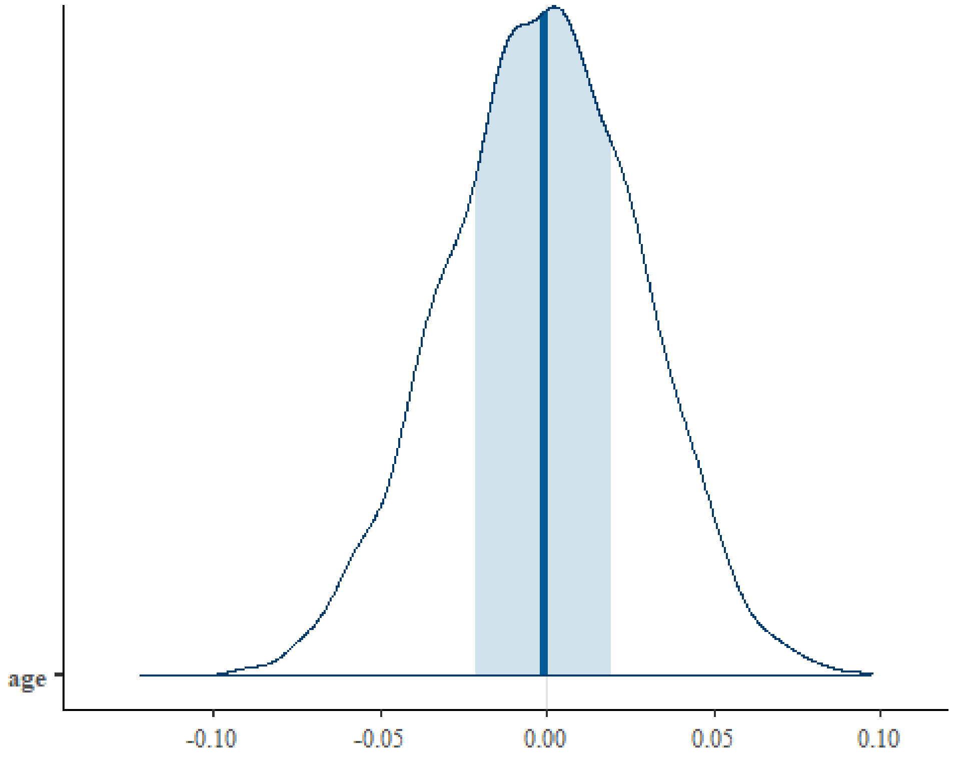
```
mcmc_areas(as.array(post1), pars = c('(Intercept)'))
```



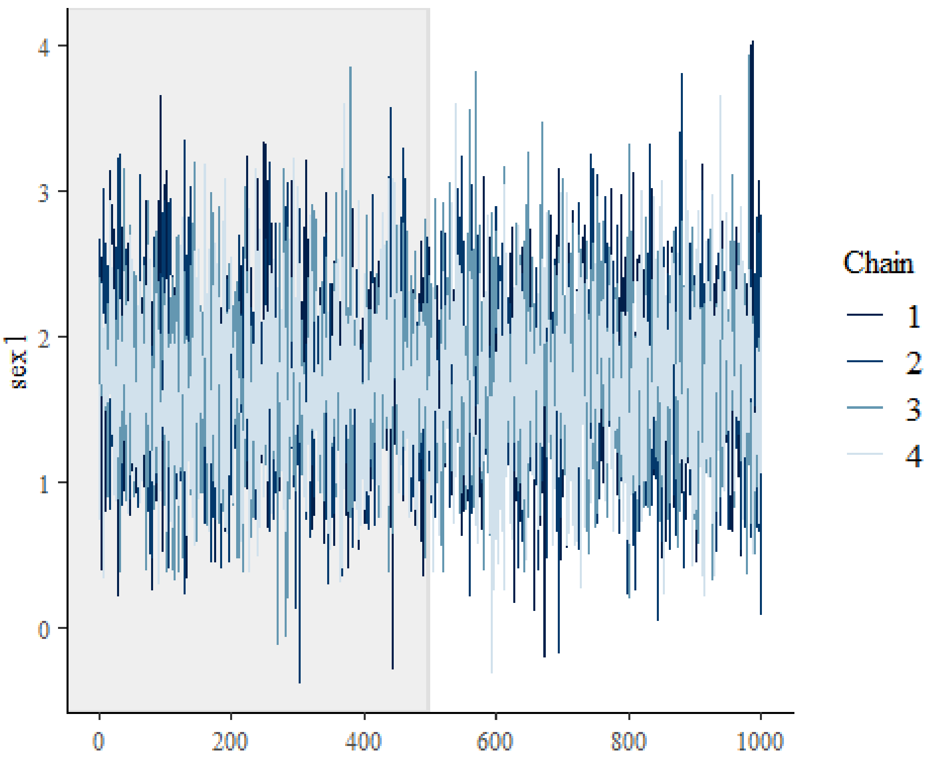
```
mcmc_trace(as.array(post1),  pars = c('age'), n_warmup = 500)
```



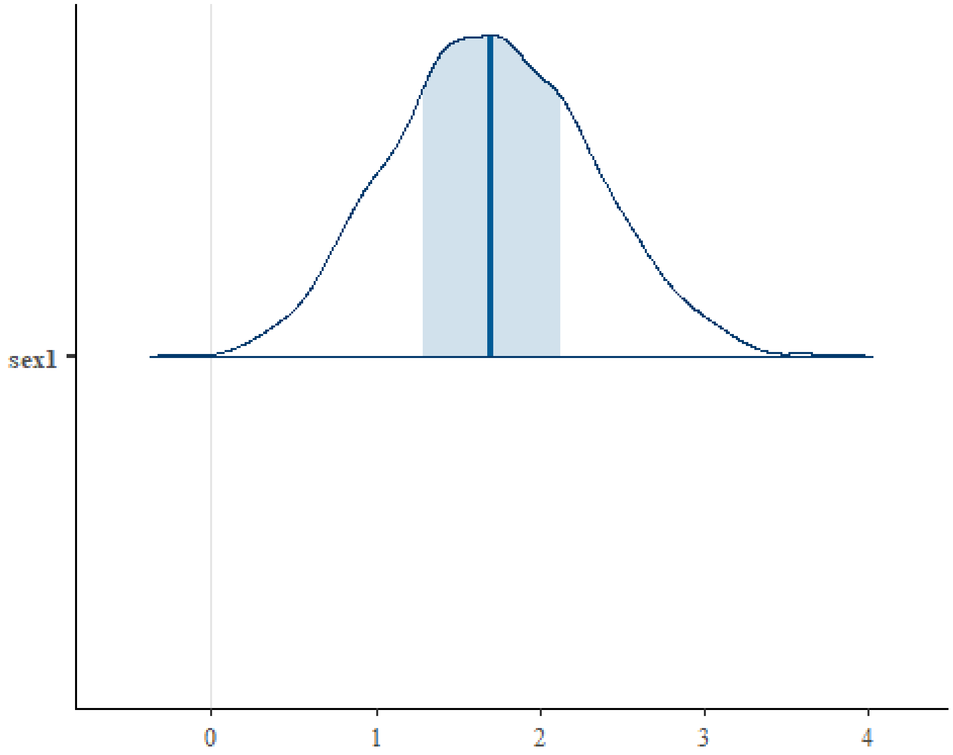
```r
mcmc_areas(as.array(post1), pars = c('age'))
```



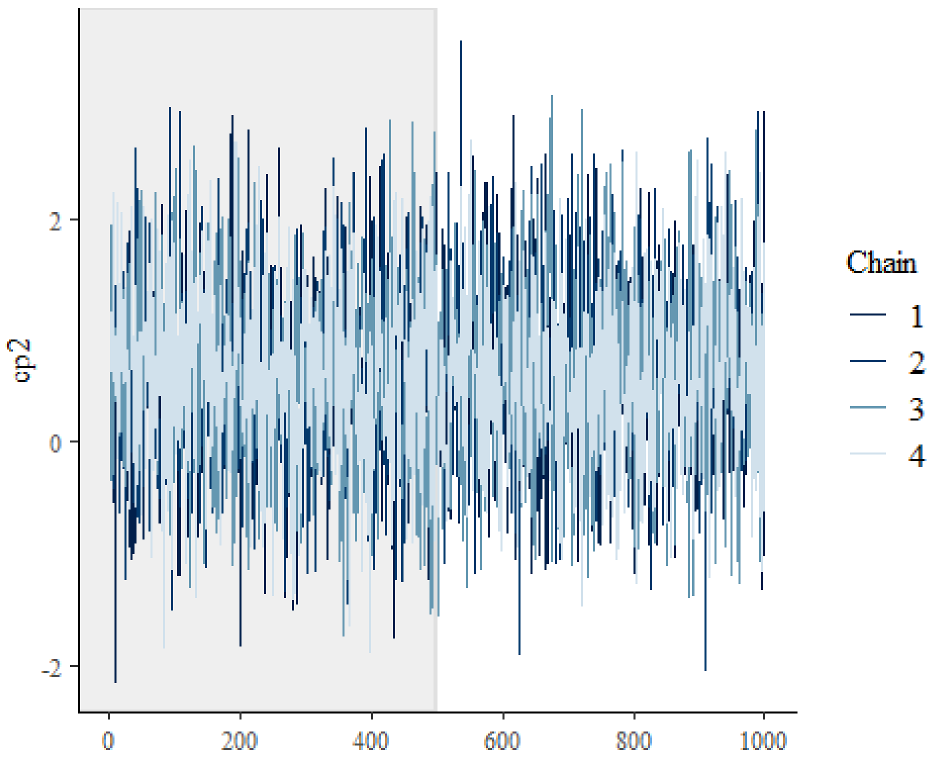
```r
mcmc_trace(as.array(post1),  pars = c('sex1'), n_warmup = 500)
```



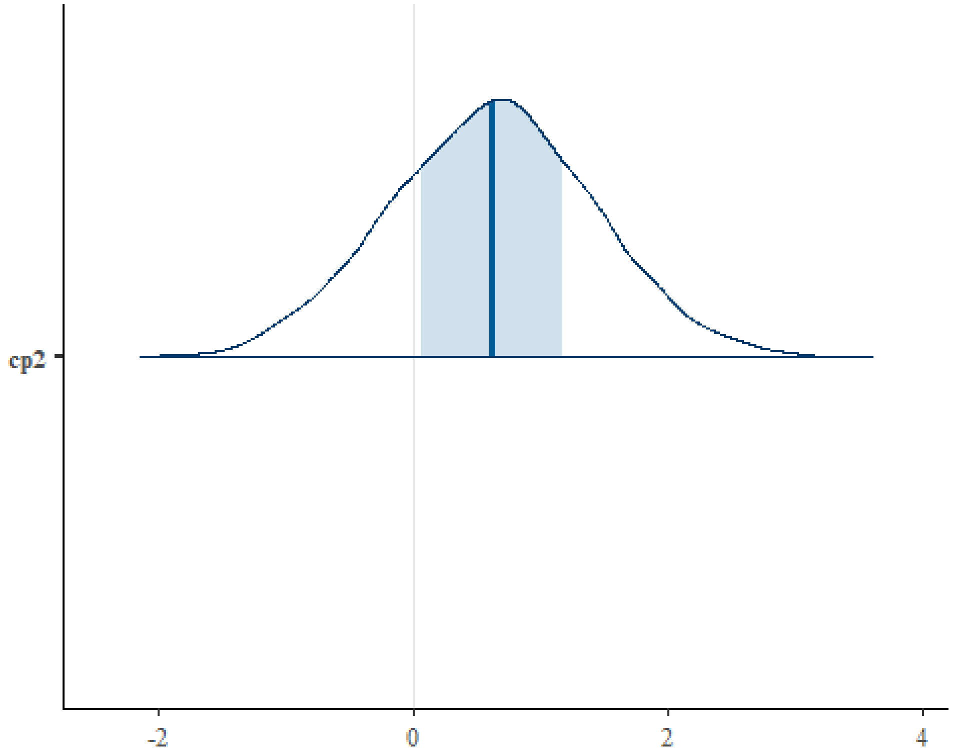
```
mcmc_areas(as.array(post1), pars = c('sex1'))
```



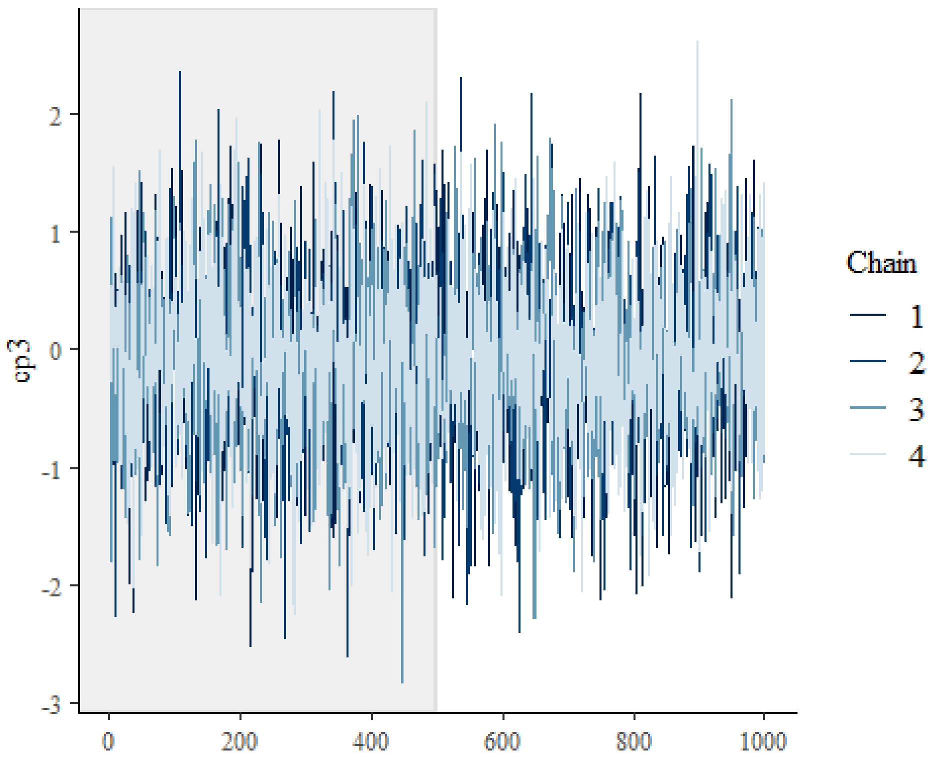
```
mcmc_trace(as.array(post1),  pars = c('cp2'), n_warmup = 500)
```



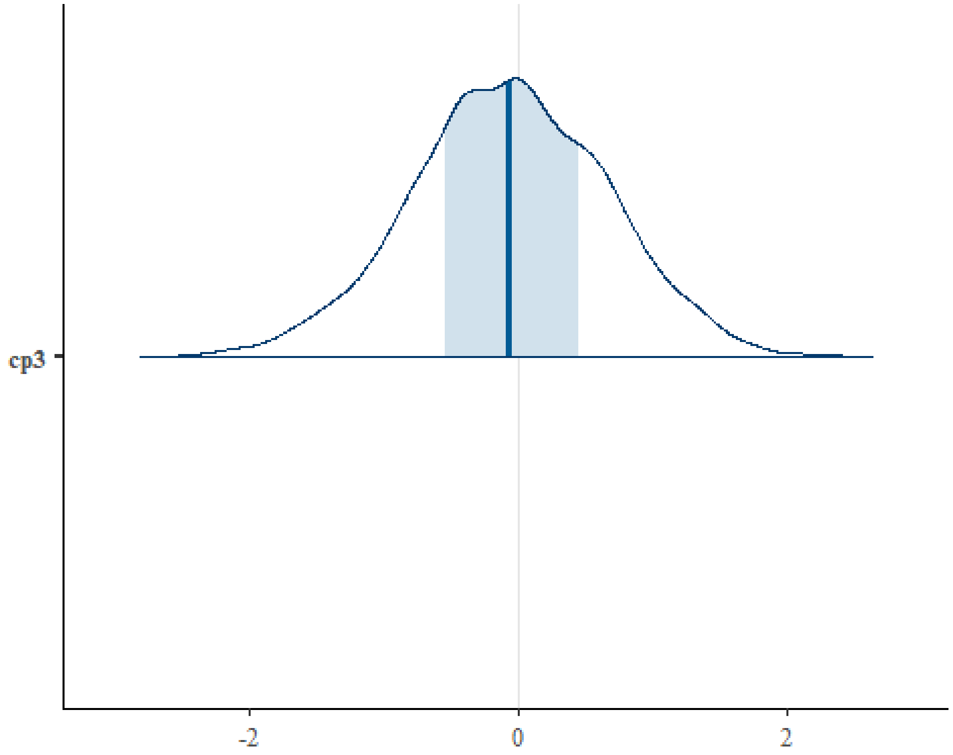
```
mcmc_areas(as.array(post1), pars = c('cp2'))
```



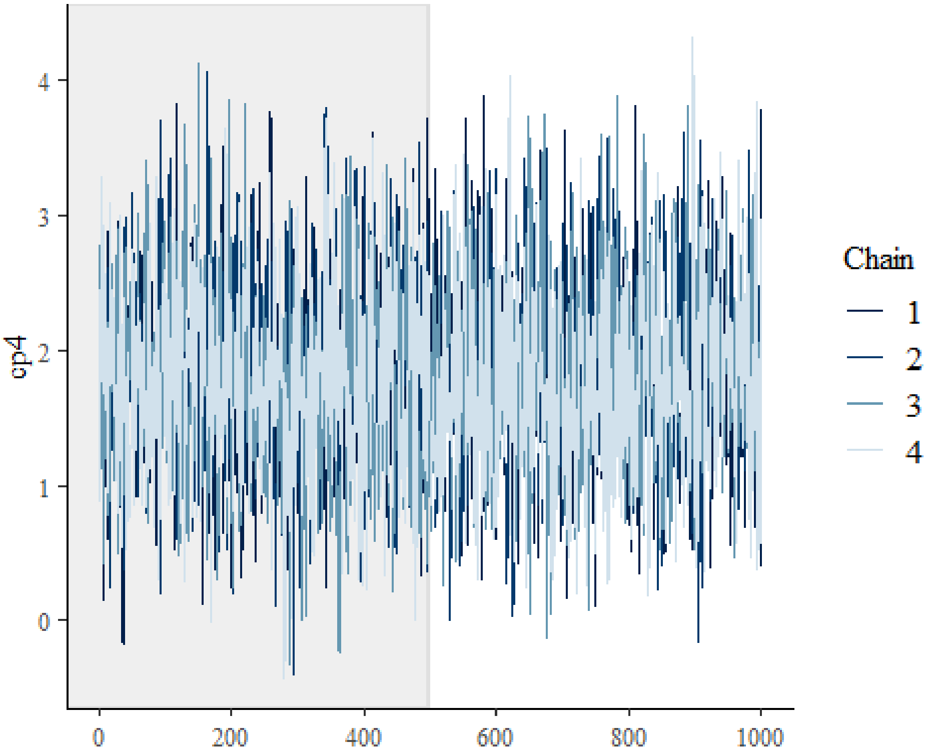
```
mcmc_trace(as.array(post1),  pars = c('cp3'), n_warmup = 500)
```



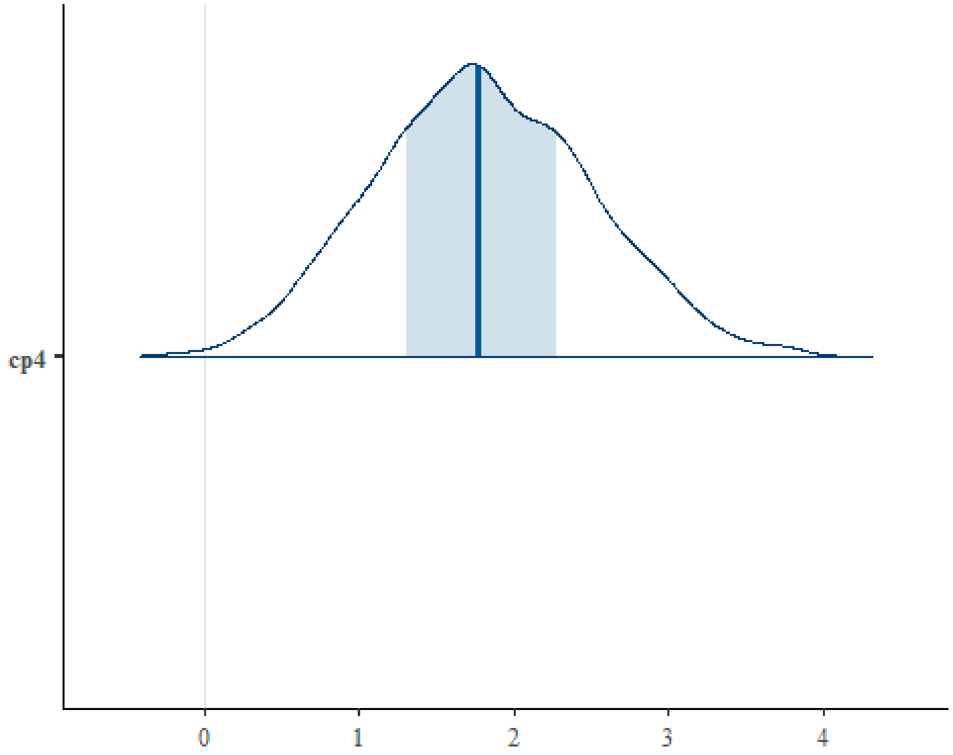
```
mcmc_areas(as.array(post1), pars = c('cp3'))
```



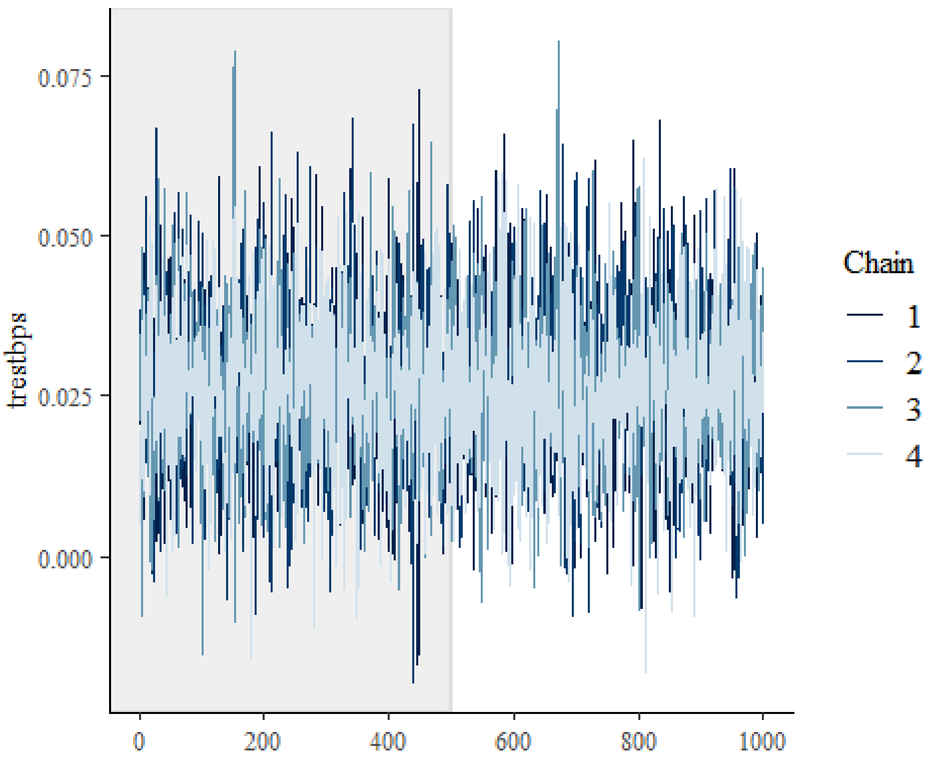
```
mcmc_trace(as.array(post1),  pars = c('cp4'), n_warmup = 500)
```



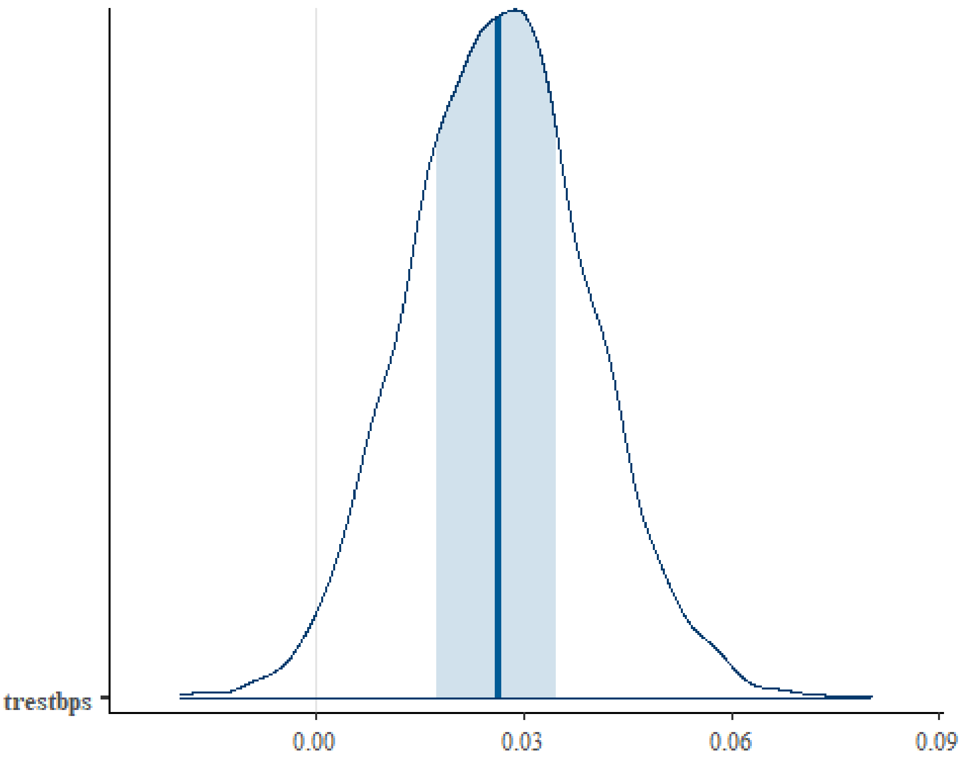
```
mcmc_areas(as.array(post1), pars = c('cp4'))
```



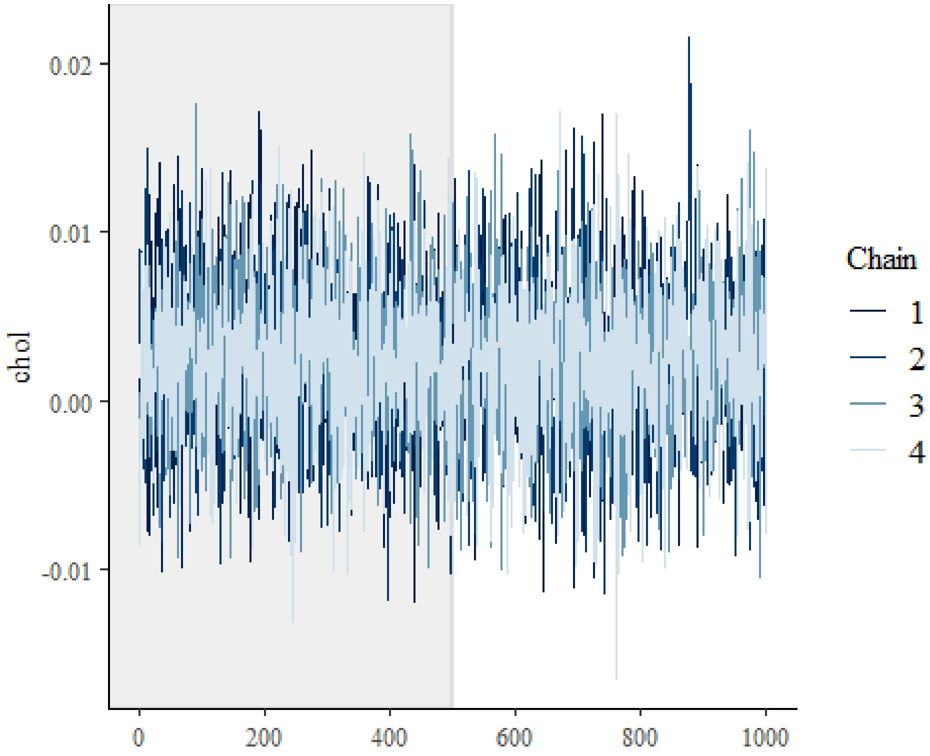
```
mcmc_trace(as.array(post1),  pars = c('trestbps'), n_warmup = 500)
```



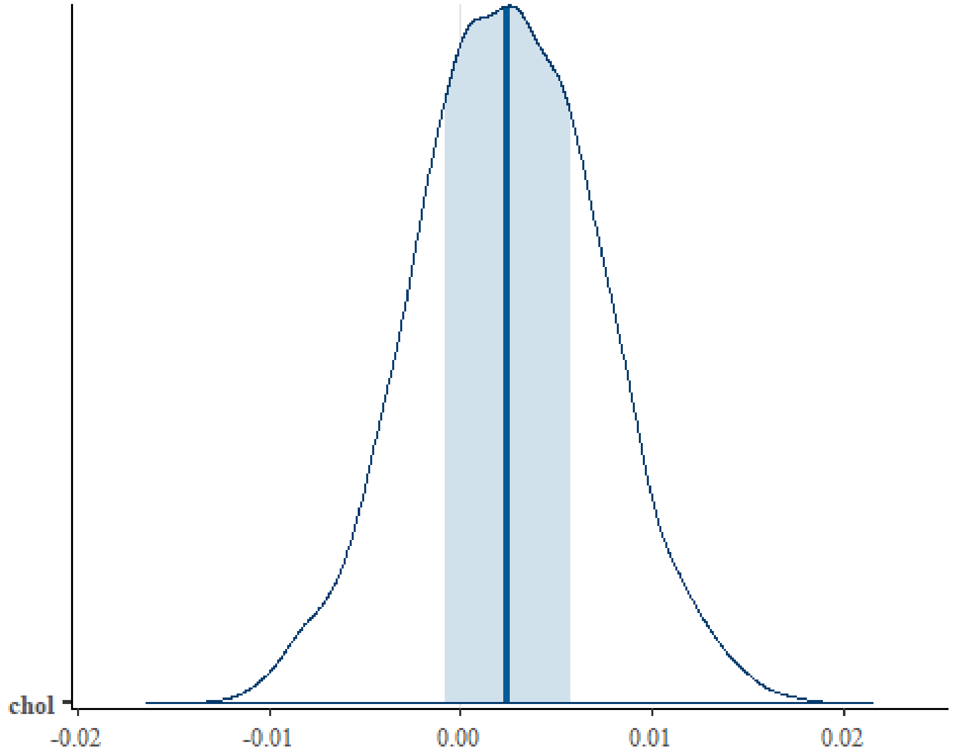
```
mcmc_areas(as.array(post1), pars = c('trestbps'))
```



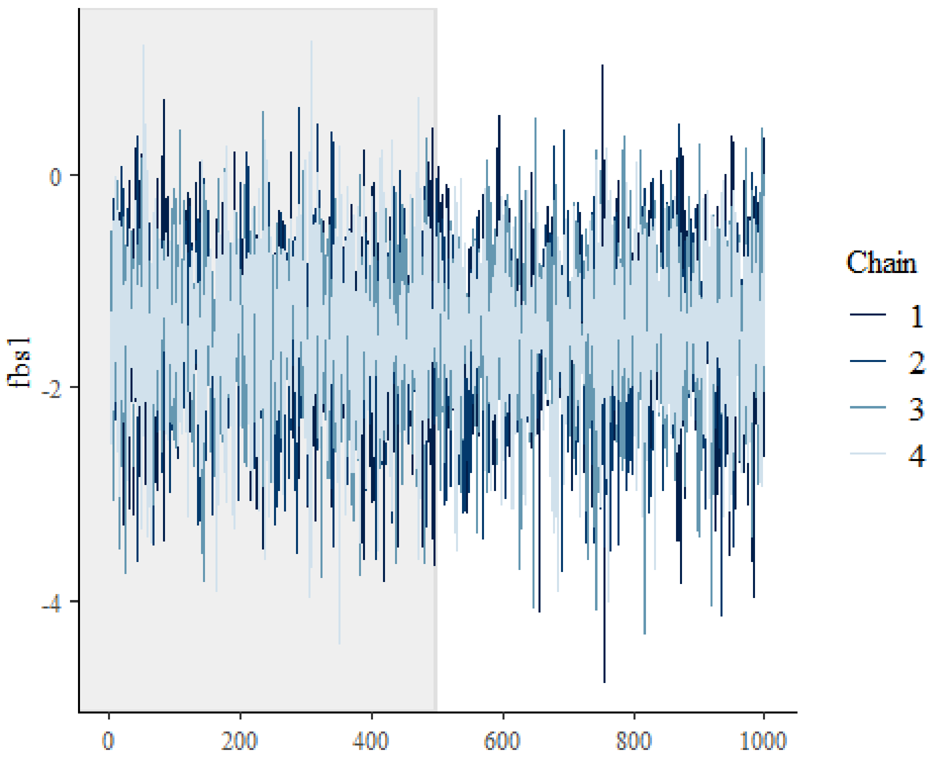
```
mcmc_trace(as.array(post1),  pars = c('chol'), n_warmup = 500)
```



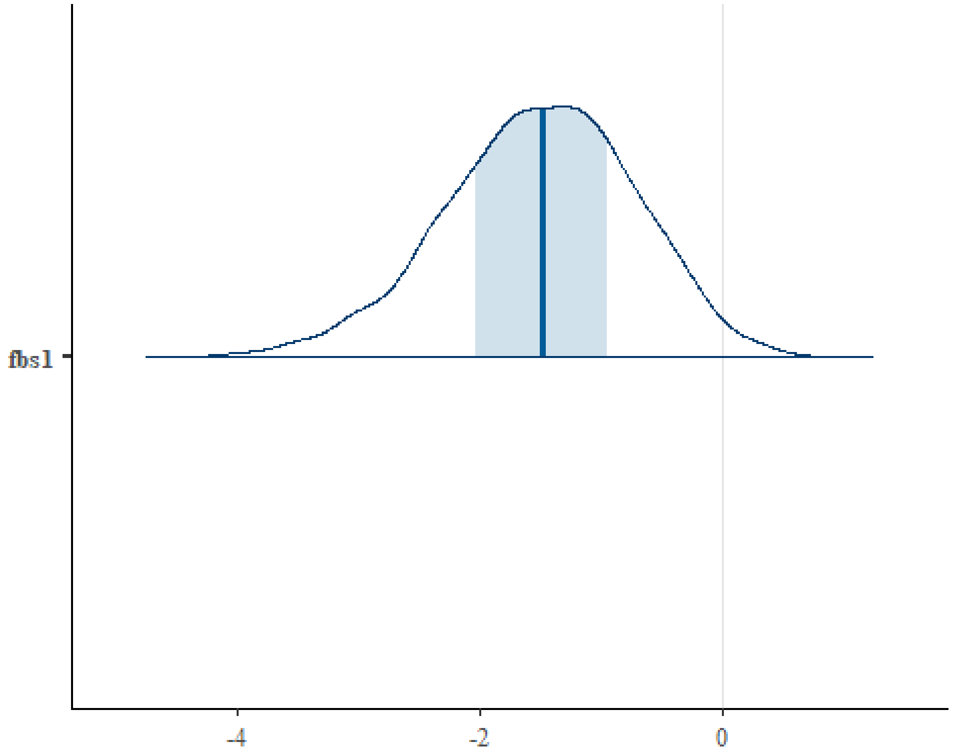
```
mcmc_areas(as.array(post1), pars = c('chol'))
```



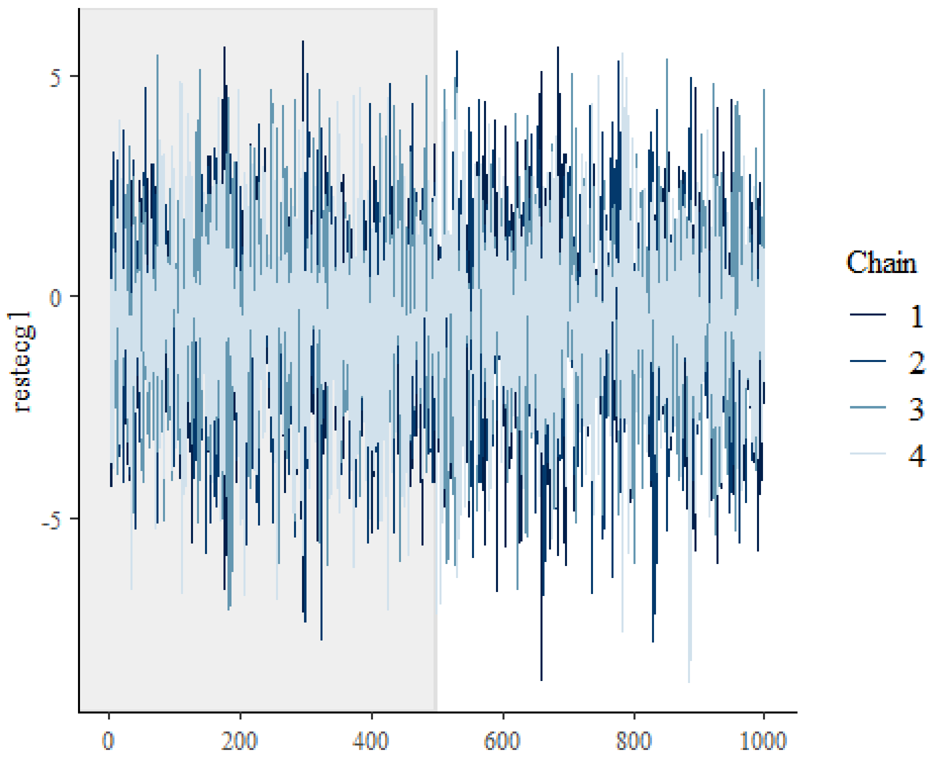
```
mcmc_trace(as.array(post1),  pars = c('fbs1'), n_warmup = 500)
```



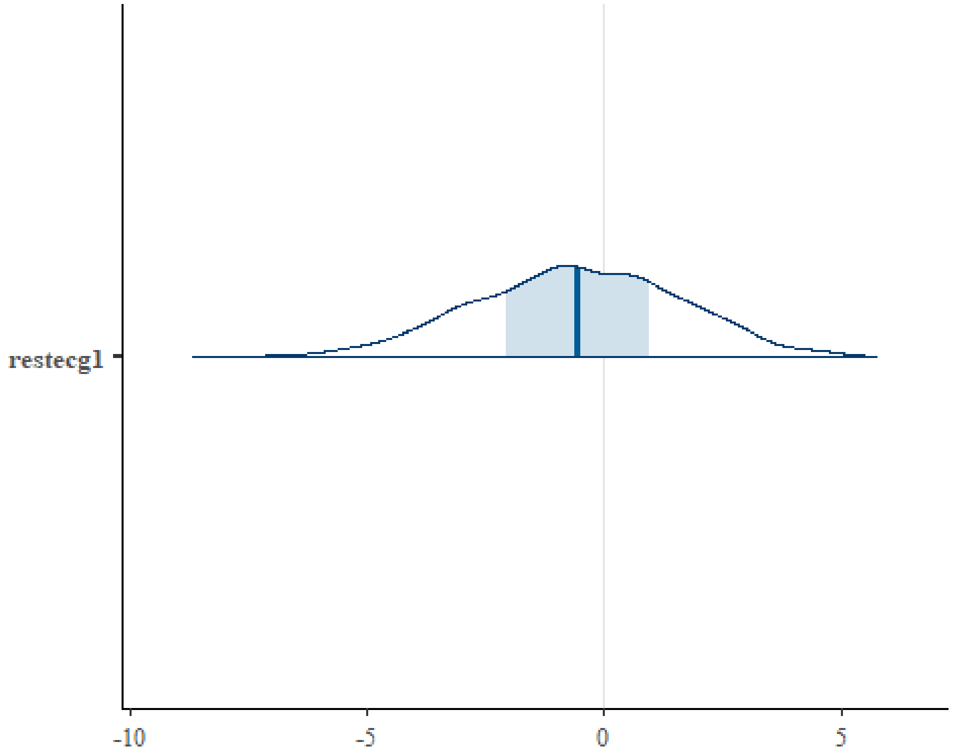
```
mcmc_areas(as.array(post1), pars = c('fbs1'))
```



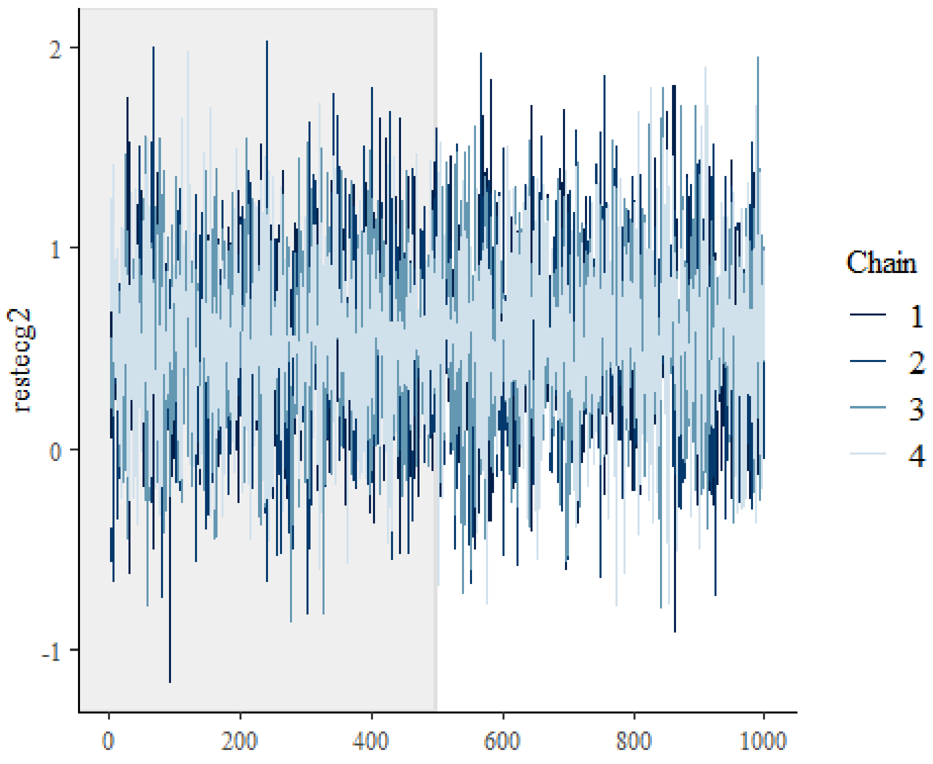
```
mcmc_trace(as.array(post1),  pars = c('restecg1'), n_warmup = 500)
```



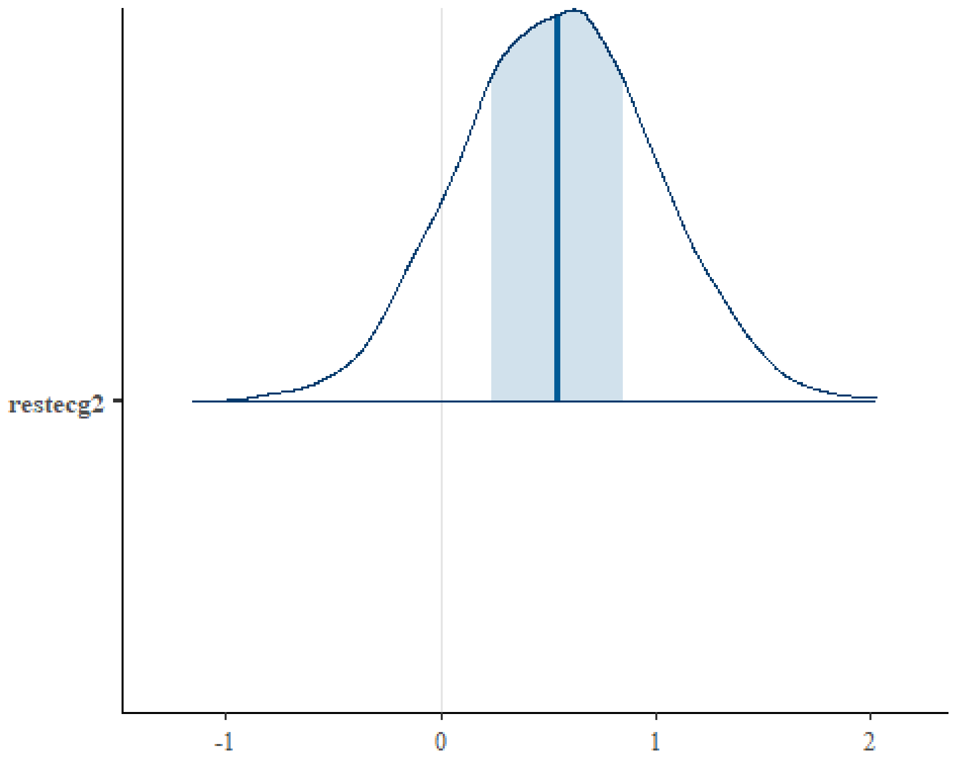
```
mcmc_areas(as.array(post1), pars = c('restecg1'))
```



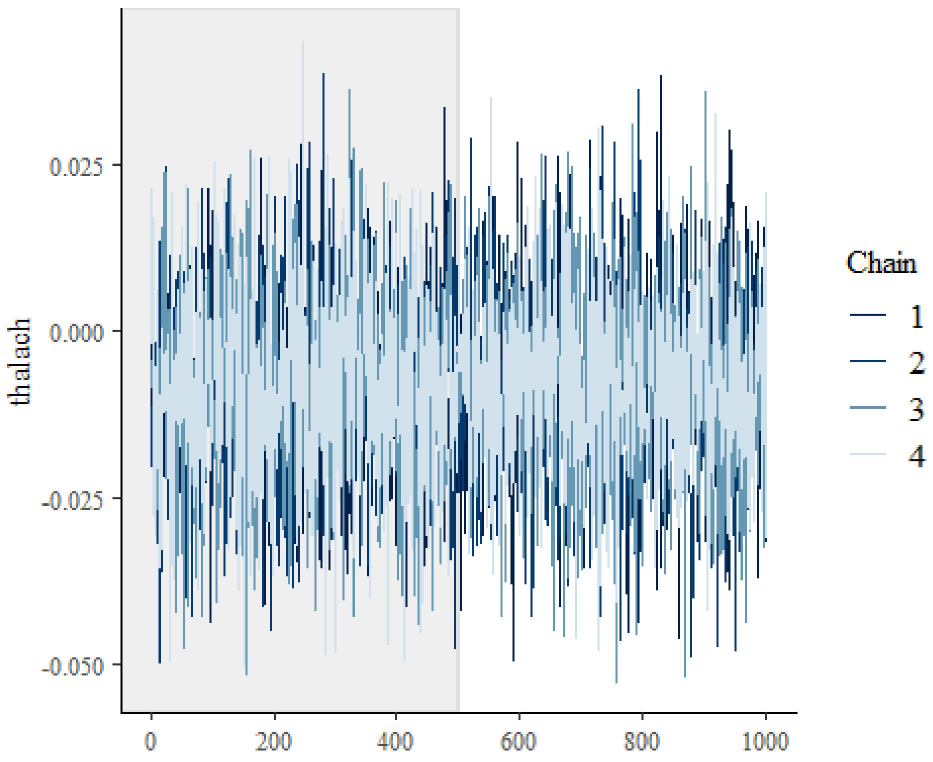
```
mcmc_trace(as.array(post1),  pars = c('restecg2'), n_warmup = 500)
```



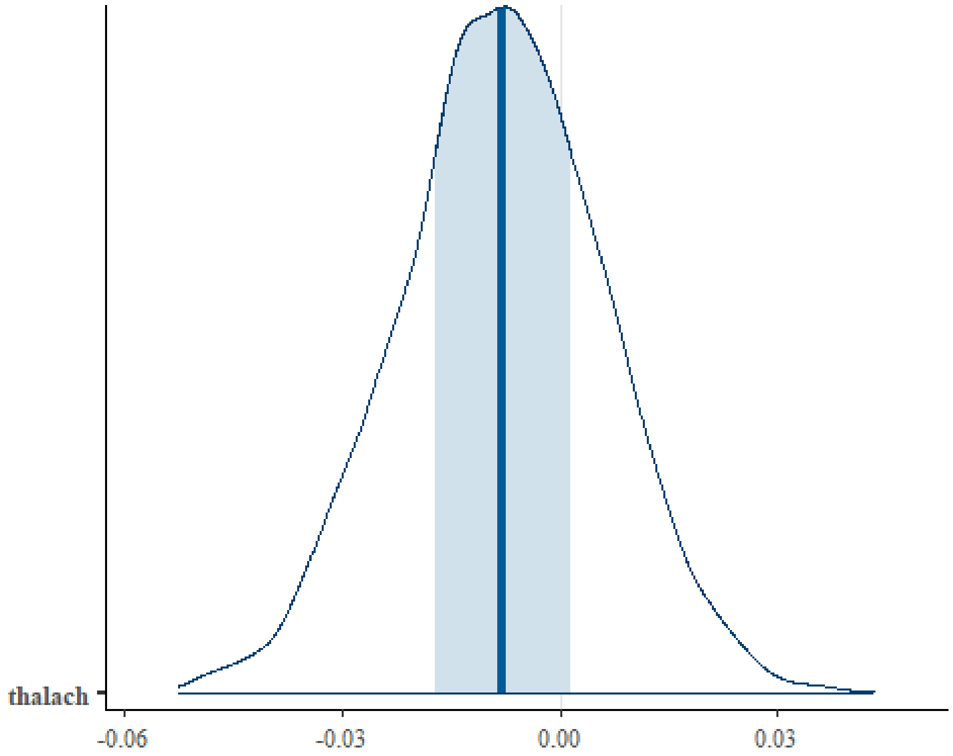
```
mcmc_areas(as.array(post1), pars = c('restecg2'))
```



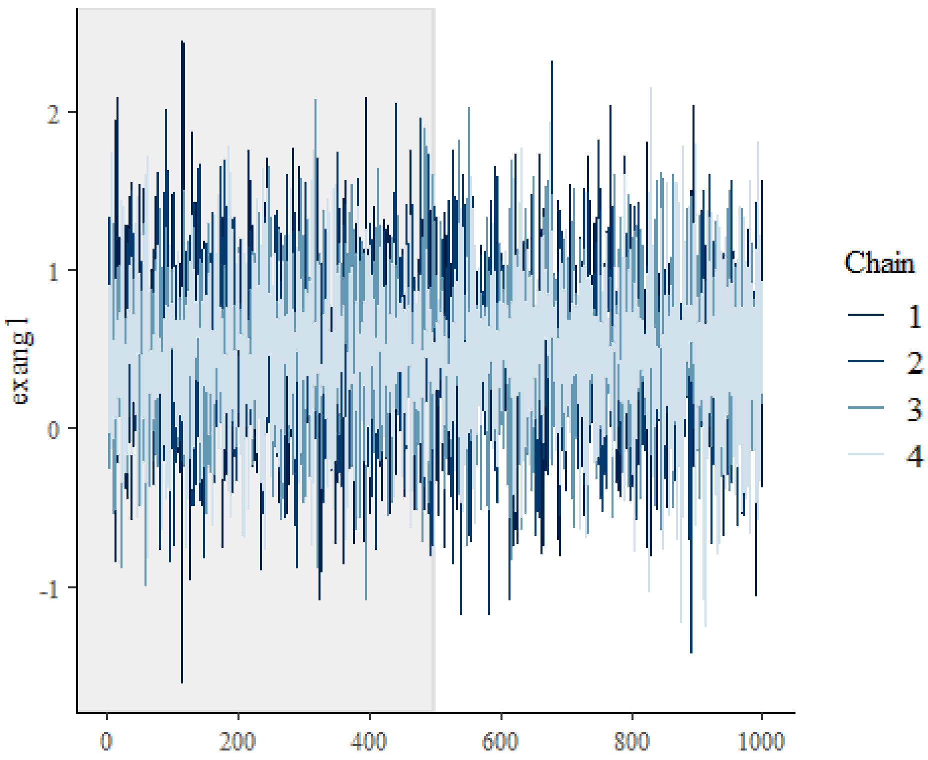
```
mcmc_trace(as.array(post1),  pars = c('thalach'), n_warmup = 500)
```



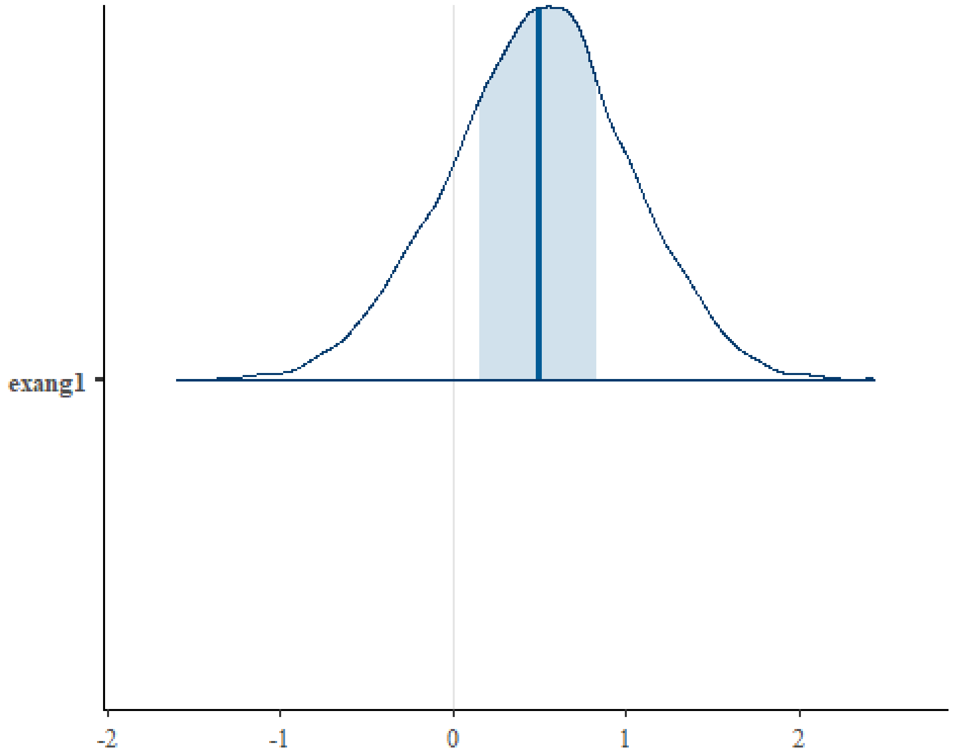
```
mcmc_areas(as.array(post1), pars = c('thalach'))
```



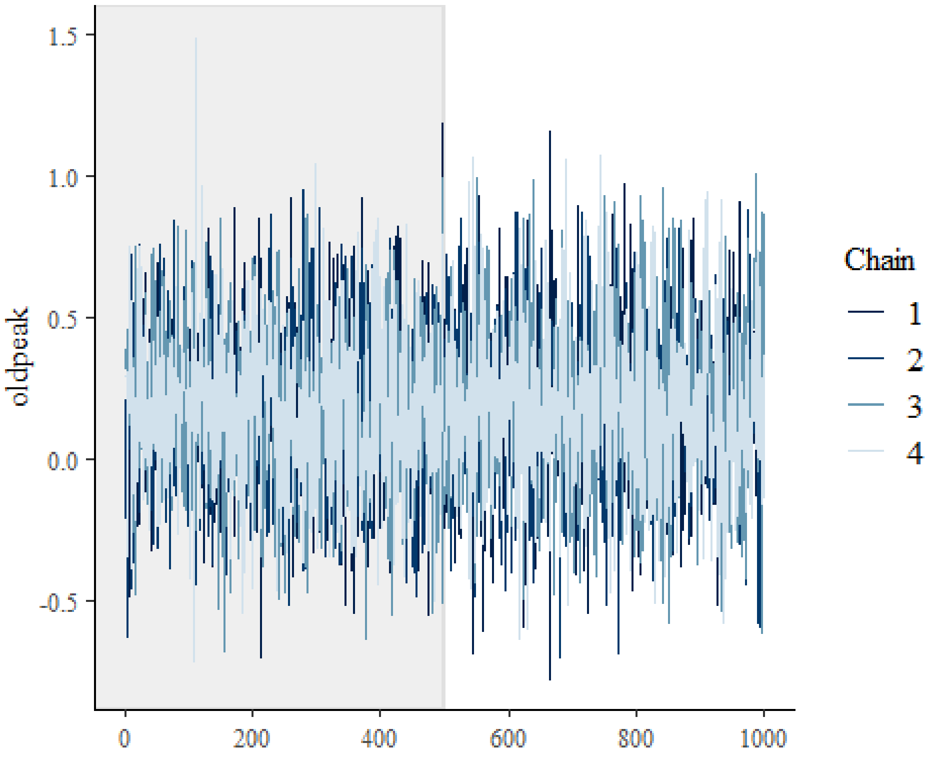
```
mcmc_trace(as.array(post1),  pars = c('exang1'), n_warmup = 500)
```



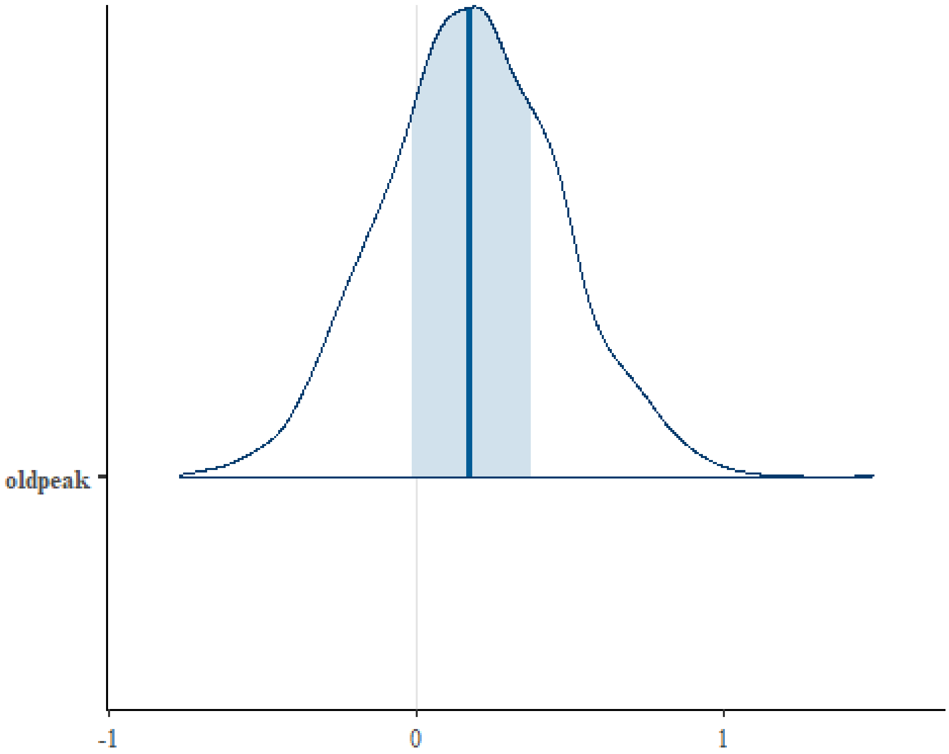
```
mcmc_areas(as.array(post1), pars = c('exang1'))
```



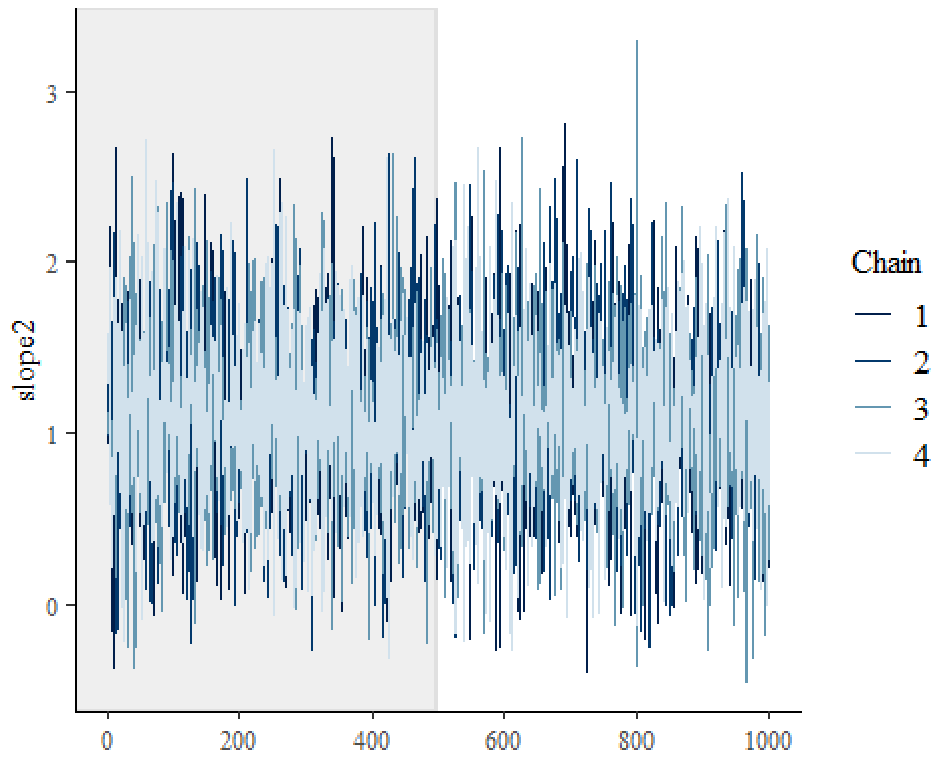
```
mcmc_trace(as.array(post1),  pars = c('oldpeak'), n_warmup = 500)
```



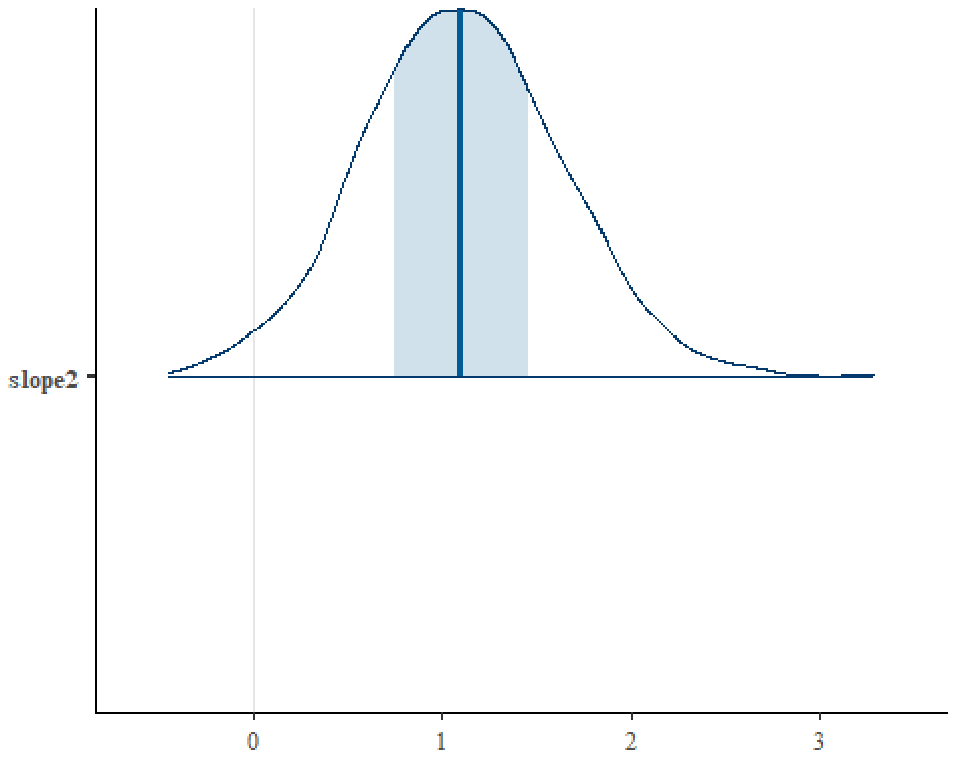
```
mcmc_areas(as.array(post1), pars = c('oldpeak'))
```



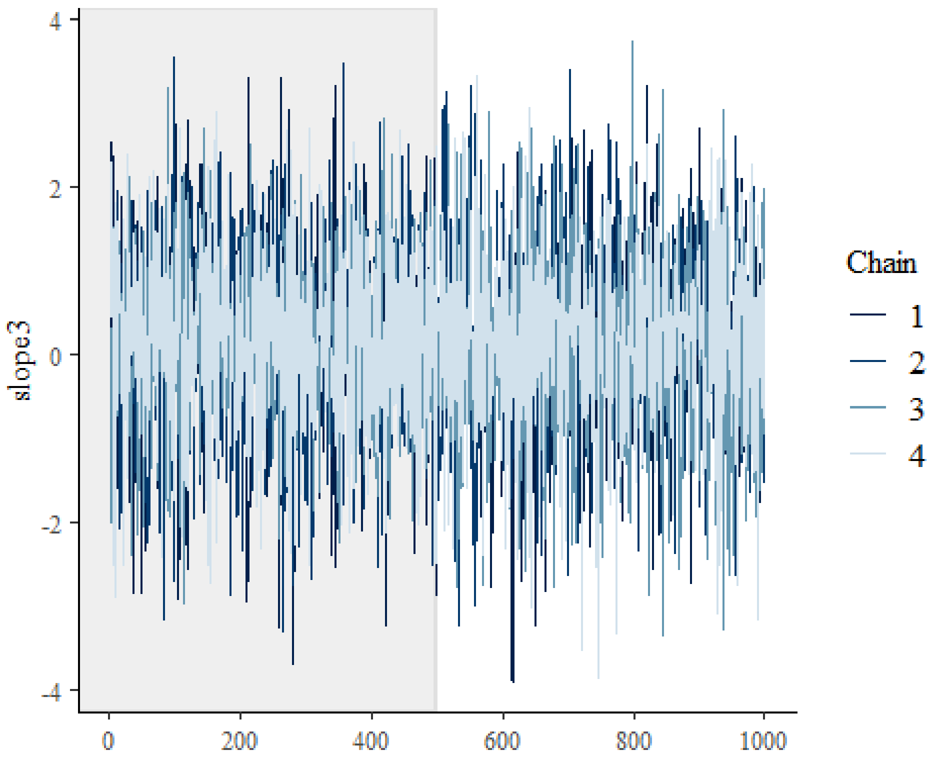
```
mcmc_trace(as.array(post1),  pars = c('slope2'), n_warmup = 500)
```



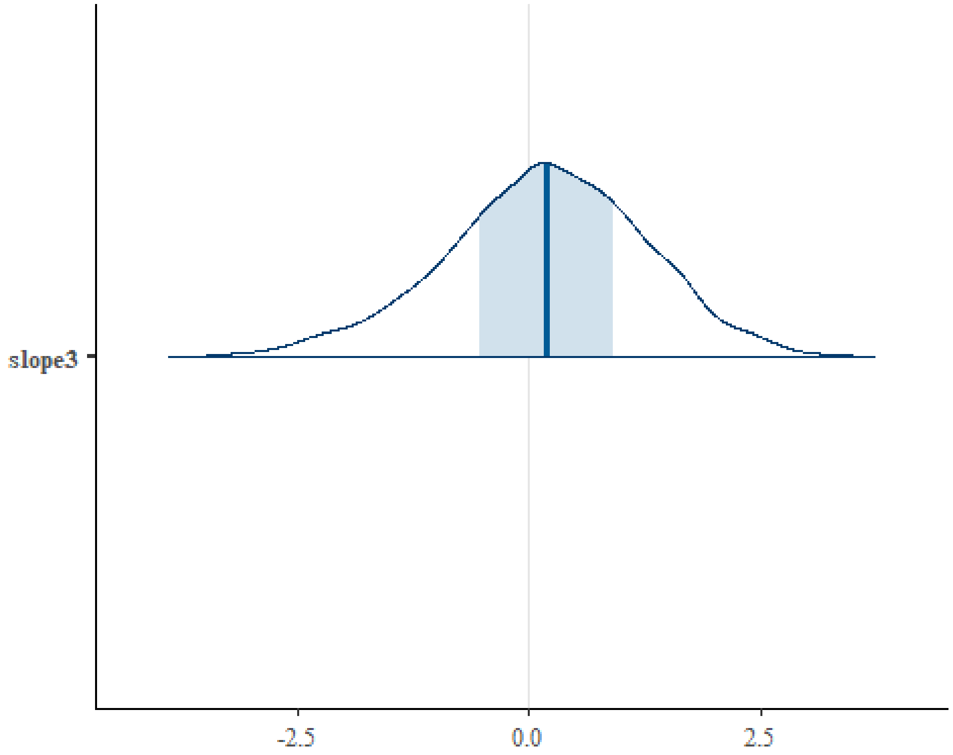
```
mcmc_areas(as.array(post1), pars = c('slope2'))
```



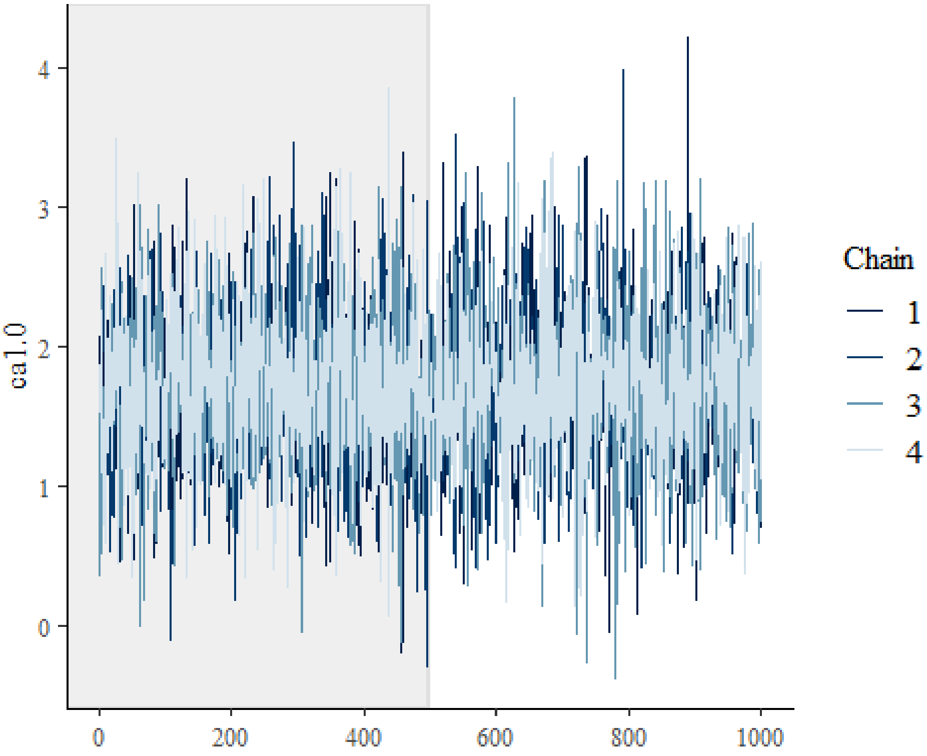
```
mcmc_trace(as.array(post1),  pars = c('slope3'), n_warmup = 500)
```



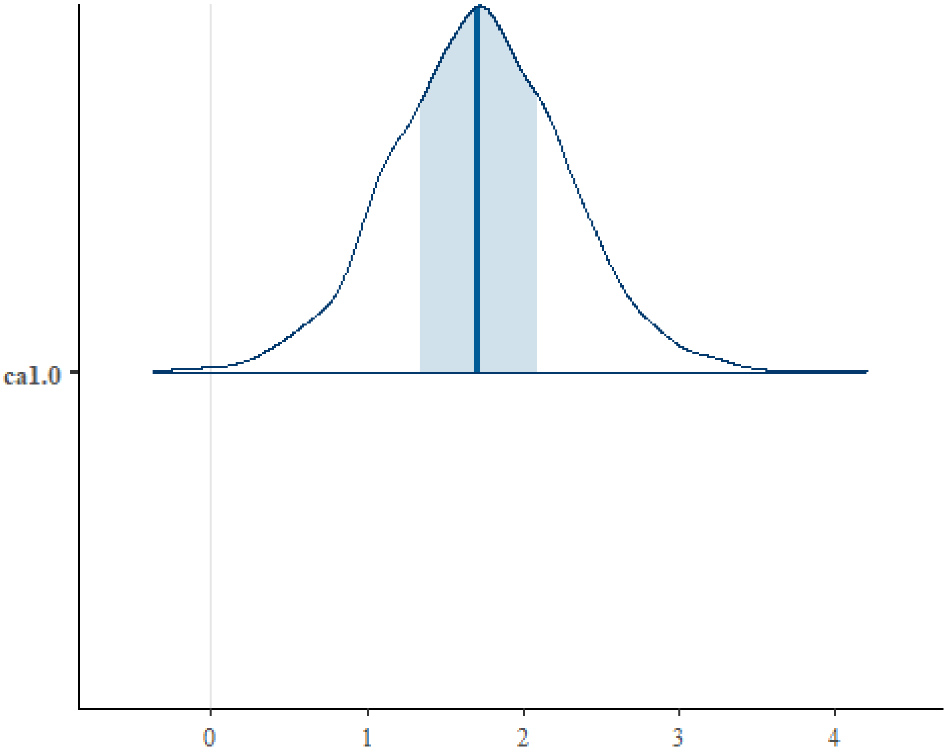
```
mcmc_areas(as.array(post1), pars = c('slope3'))
```



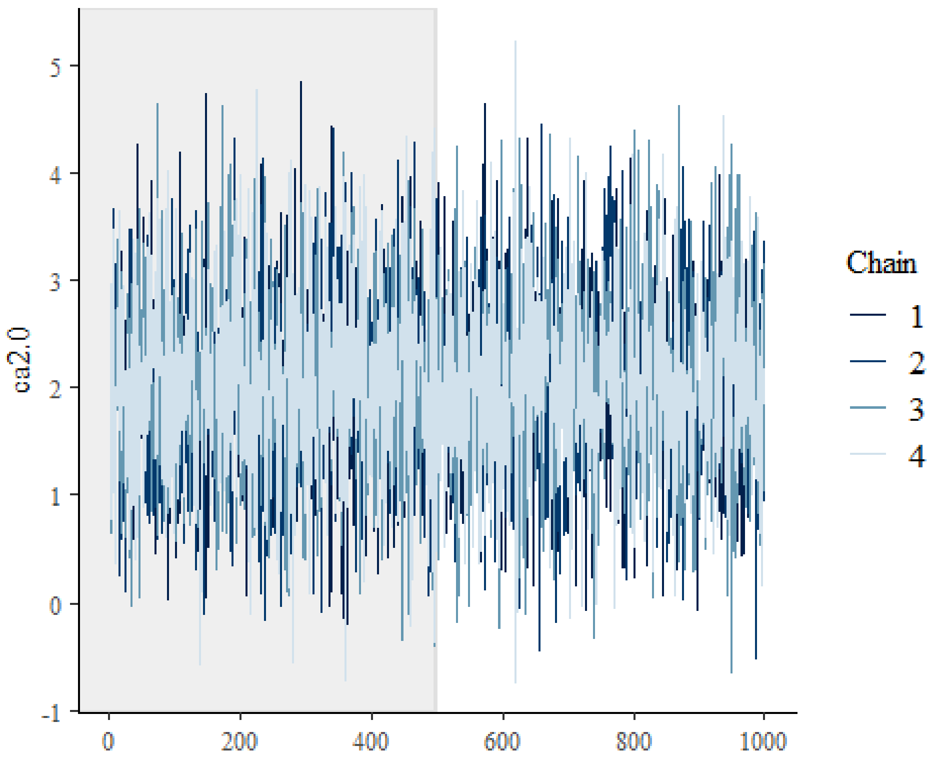
```
mcmc_trace(as.array(post1),  pars = c('ca1.0'), n_warmup = 500)
```



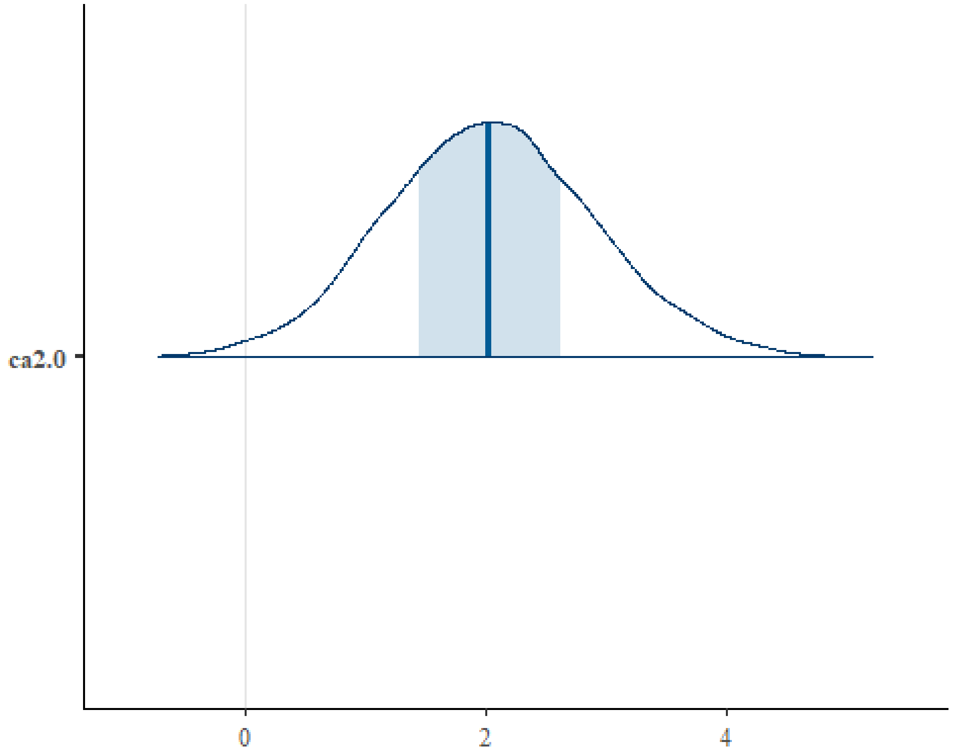
```
mcmc_areas(as.array(post1), pars = c('ca1.0'))
```



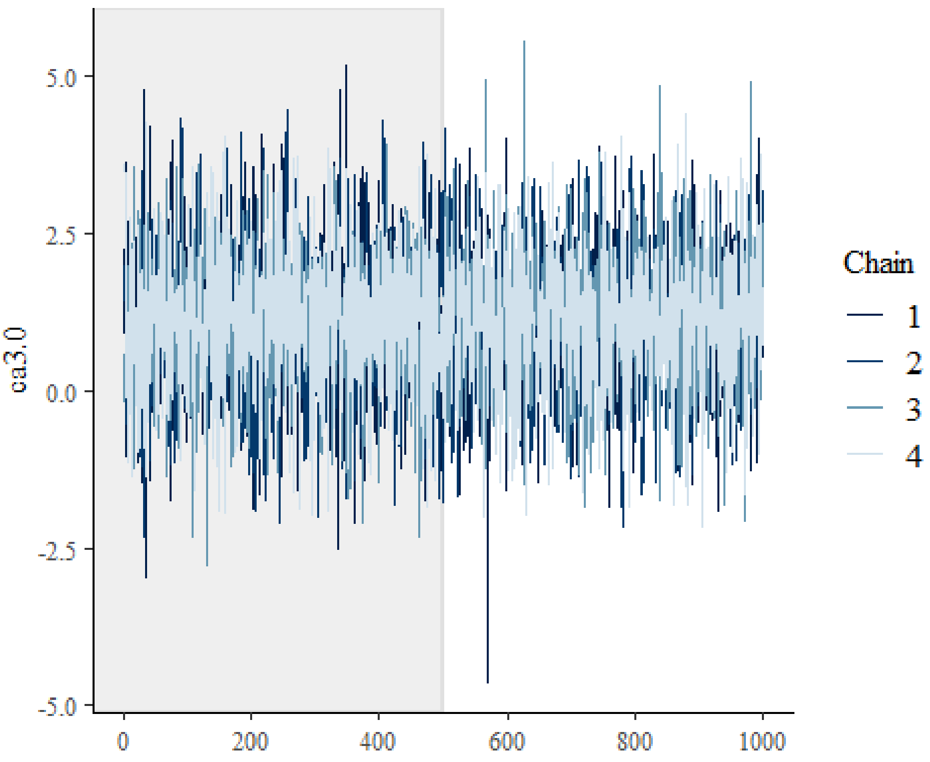
```
mcmc_trace(as.array(post1),  pars = c('ca2.0'), n_warmup = 500)
```



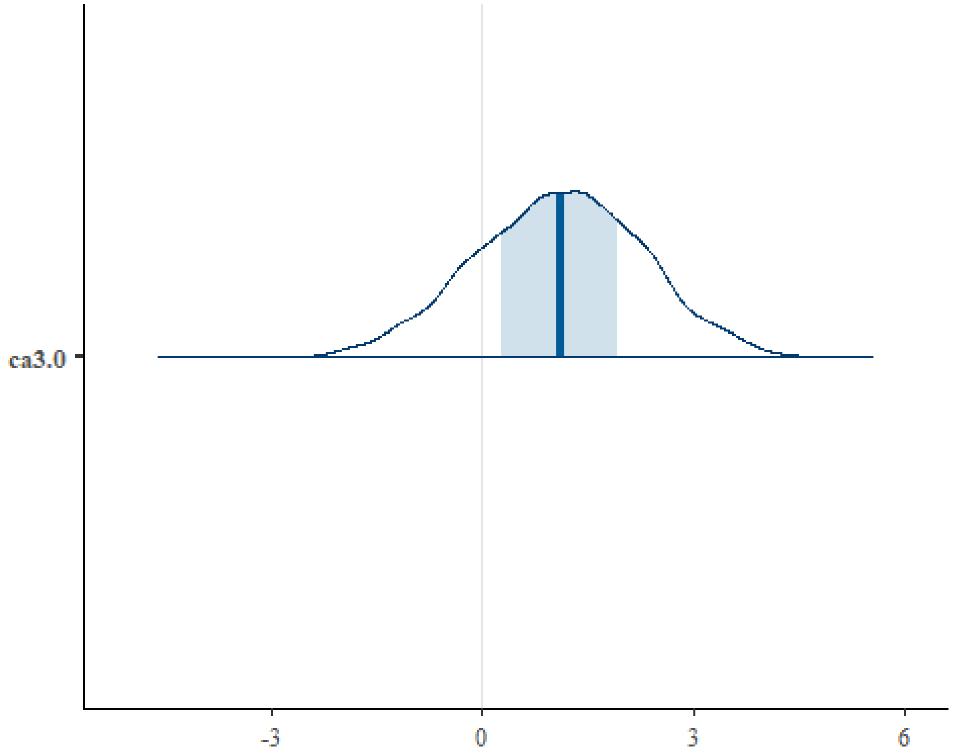
```
mcmc_areas(as.array(post1), pars = c('ca2.0'))
```



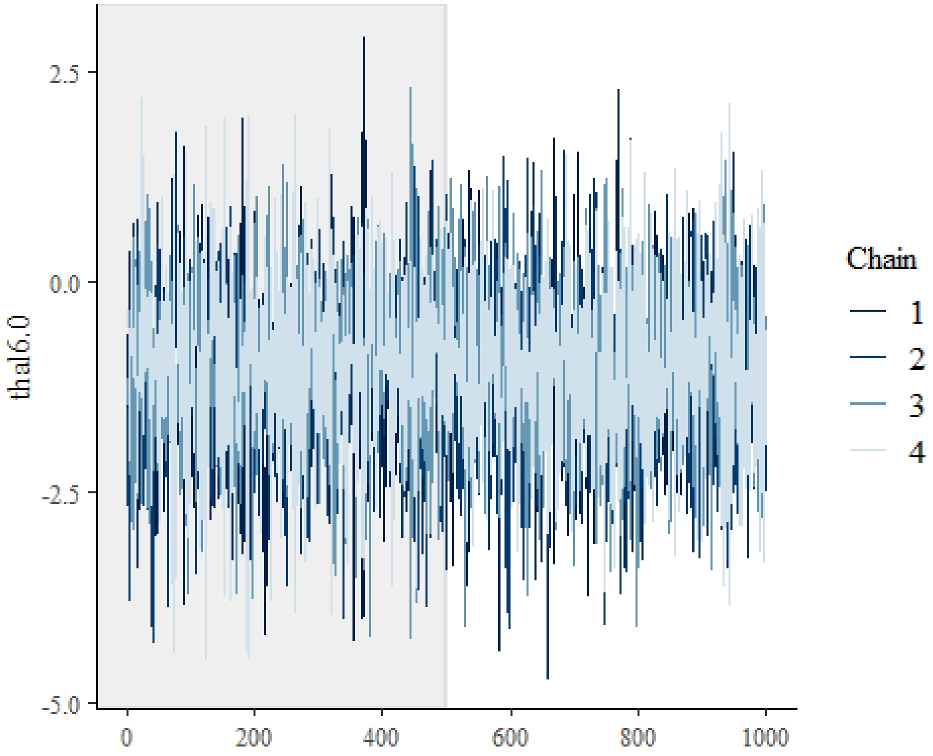
```
mcmc_trace(as.array(post1),  pars = c('ca3.0'), n_warmup = 500)
```



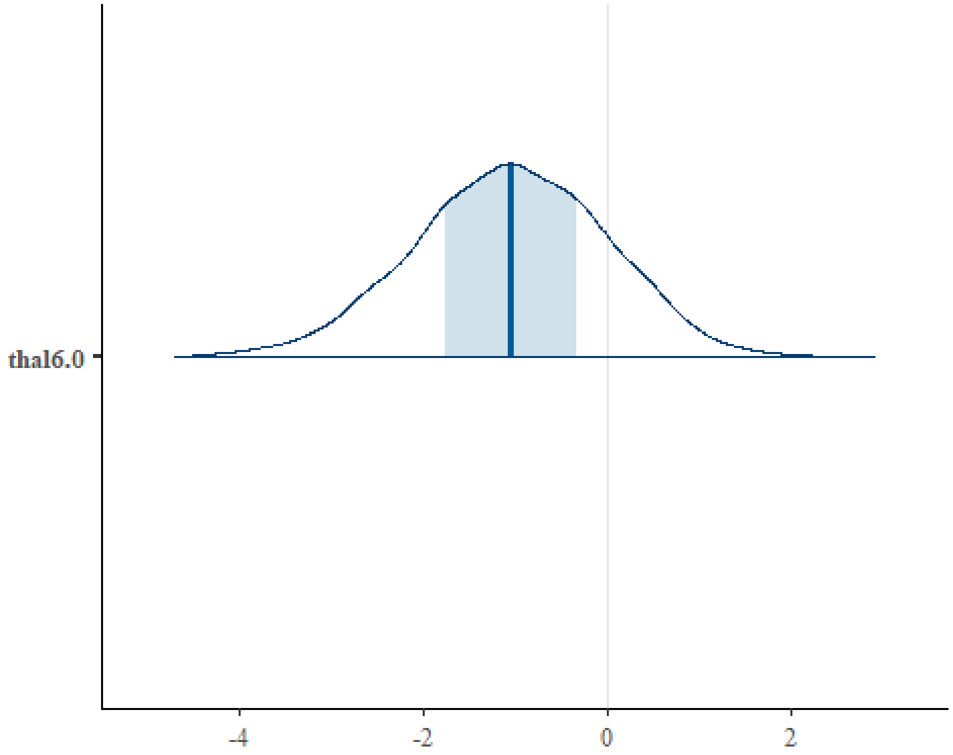
```
mcmc_areas(as.array(post1), pars = c('ca3.0'))
```



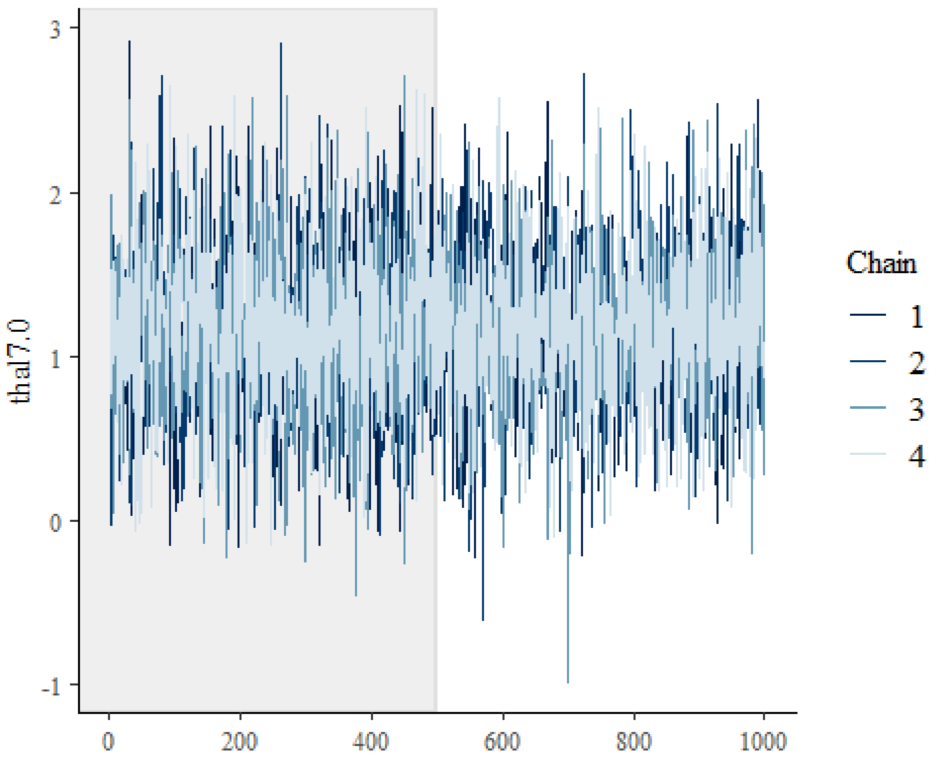
```
mcmc_trace(as.array(post1),  pars = c('thal6.0'), n_warmup = 500)
```



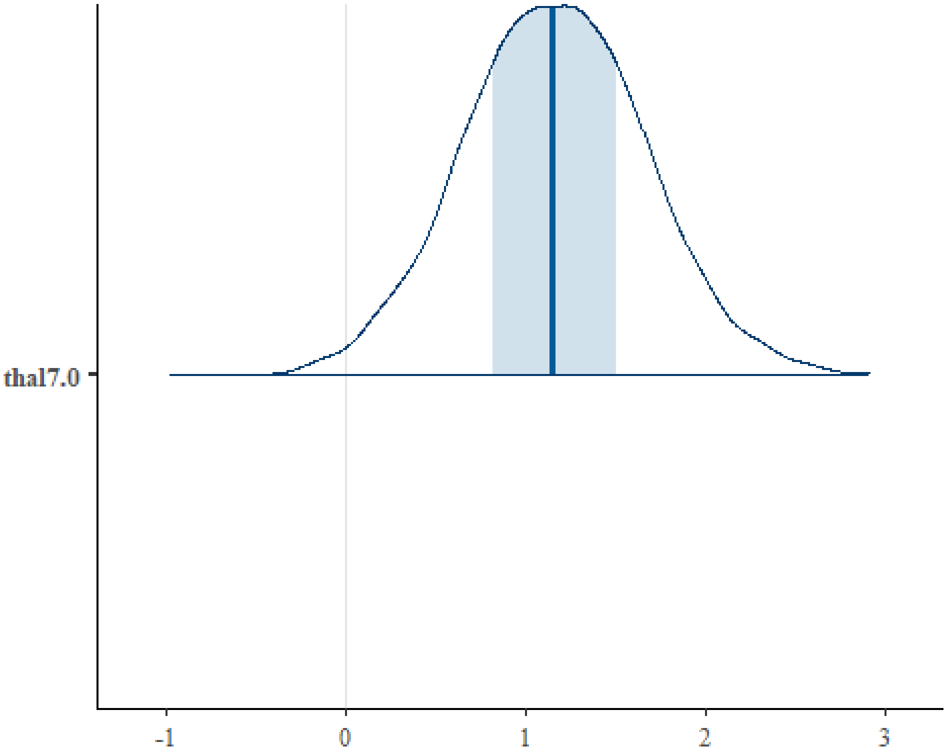
```
mcmc_areas(as.array(post1), pars = c('thal6.0'))
```



```
mcmc_trace(as.array(post1),  pars = c('thal7.0'), n_warmup = 500)
```



```
mcmc_areas(as.array(post1), pars = c('thal7.0'))
```



Analyze with No Cardiovascular disease and Cardiovascular disease with severity = 2

```
# Taken the class with 0 and 1 only
data5 <- data2[data2['class'] == 0 | data2['class'] == 2,]
post3 <- stan_glm(class ~ ., data = data5,
                  family = binomial(link = "logit"),
                  prior = normal_prior, prior_intercept = normal_prior)

##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.52 seconds (Warm-up)
## Chain 1:                1.89 seconds (Sampling)
## Chain 1:                4.41 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.29 seconds (Warm-up)
## Chain 2:                1.97 seconds (Sampling)
## Chain 2:                4.26 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.604 seconds (Warm-up)
## Chain 3:                1.846 seconds (Sampling)
## Chain 3:                4.45 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.099 seconds (Warm-up)
## Chain 4:                2.017 seconds (Sampling)
## Chain 4:                4.116 seconds (Total)
## Chain 4:
```

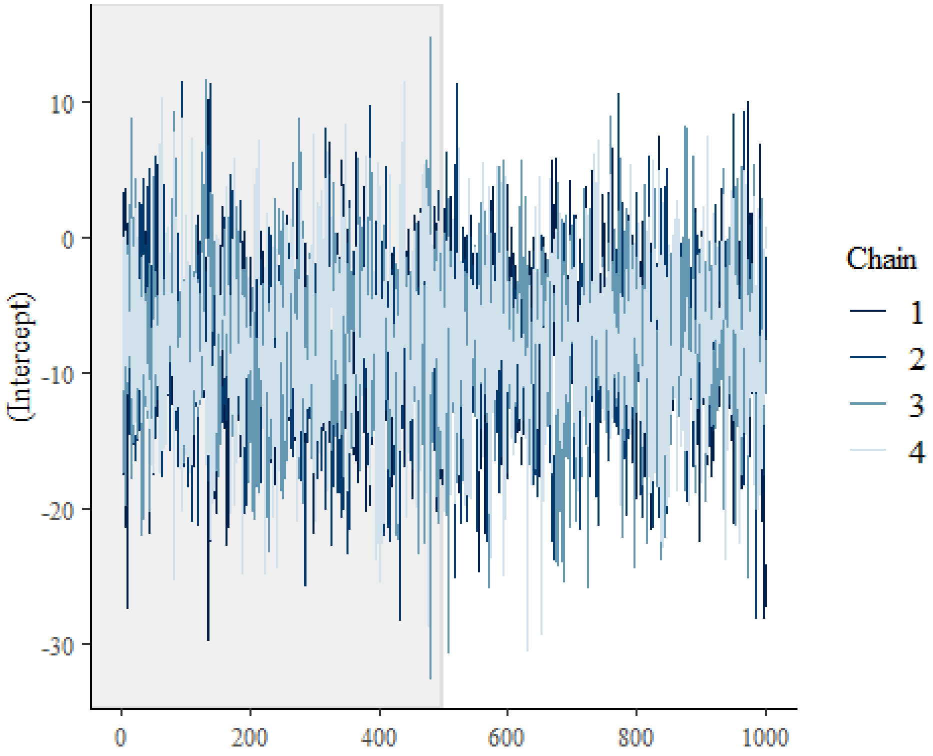
```
round(coef(post3), 2)
```

```
## (Intercept)         age        sex1         cp2         cp3         cp4 
##       -7.56       -0.01        1.61        0.47       -0.30        3.21 
##    trestbps        chol        fbs1    restecg1    restecg2     thalach 
##        0.02        0.01        0.63        0.74       -0.36       -0.04 
##      exang1     oldpeak      slope2      slope3       ca1.0       ca2.0 
##        1.24        1.09        1.42        0.62        4.38        3.09 
##       ca3.0     thal6.0     thal7.0 
##        2.84        1.67        1.91
```

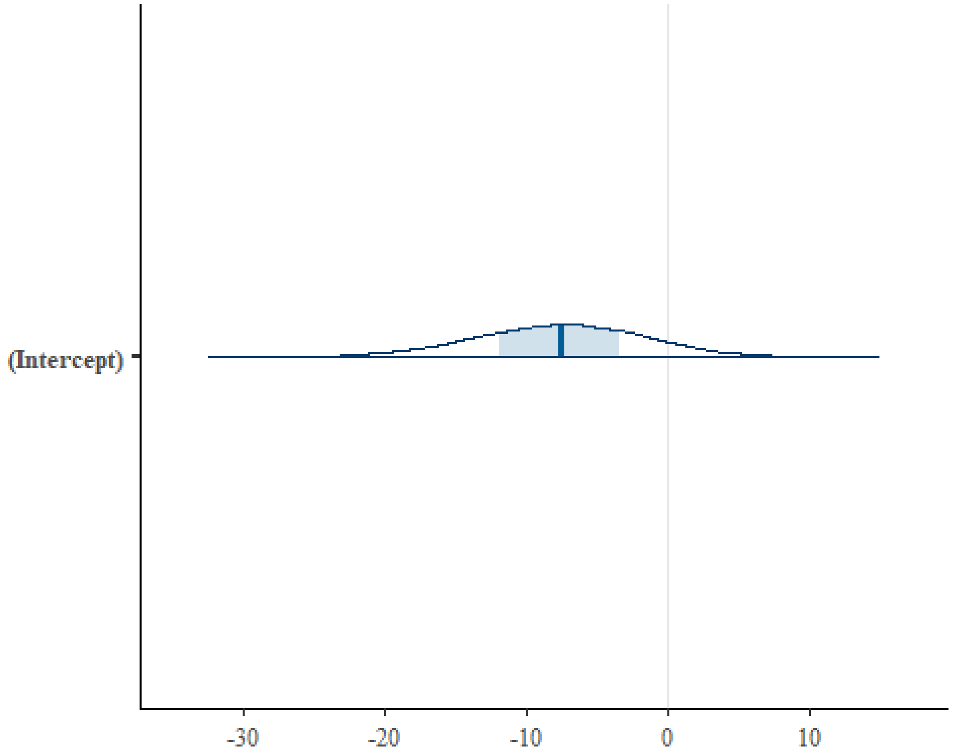
```
round(posterior_interval(post3, prob = 0.95), 2)
```

```
##               2.5% 97.5%
## (Intercept) -20.62  4.39
## age          -0.12  0.11
## sex1         -0.27  3.66
## cp2          -2.94  3.65
## cp3          -2.73  2.25
## cp4           1.04  5.57
## trestbps     -0.03  0.07
## chol         -0.01  0.02
## fbs1         -1.28  2.49
## restecg1     -3.25  4.88
## restecg2     -2.04  1.28
## thalach      -0.09  0.00
## exang1       -0.37  2.84
## oldpeak       0.22  2.07
## slope2       -0.39  3.29
## slope3       -2.40  3.58
## ca1.0         2.47  6.53
## ca2.0         0.69  5.69
## ca3.0         0.45  5.41
## thal6.0      -0.57  3.99
## thal7.0       0.24  3.65
```

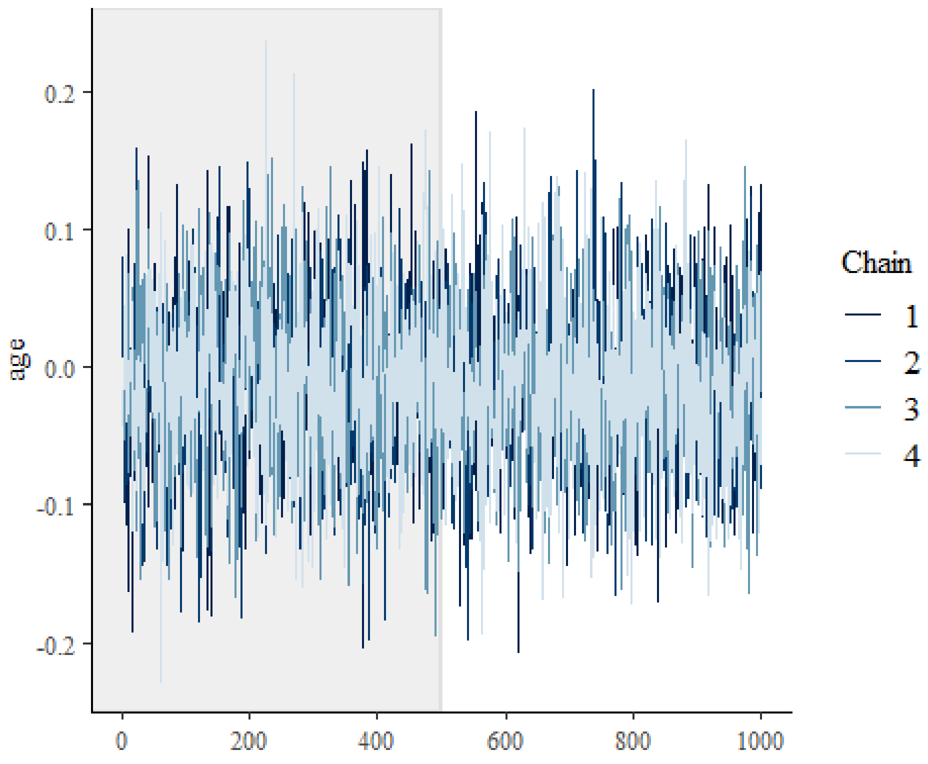
```
post1 <- post3  # to make reuse of the code of plotting

mcmc_trace(as.array(post1),  pars = c('(Intercept)'), n_warmup = 500)
```



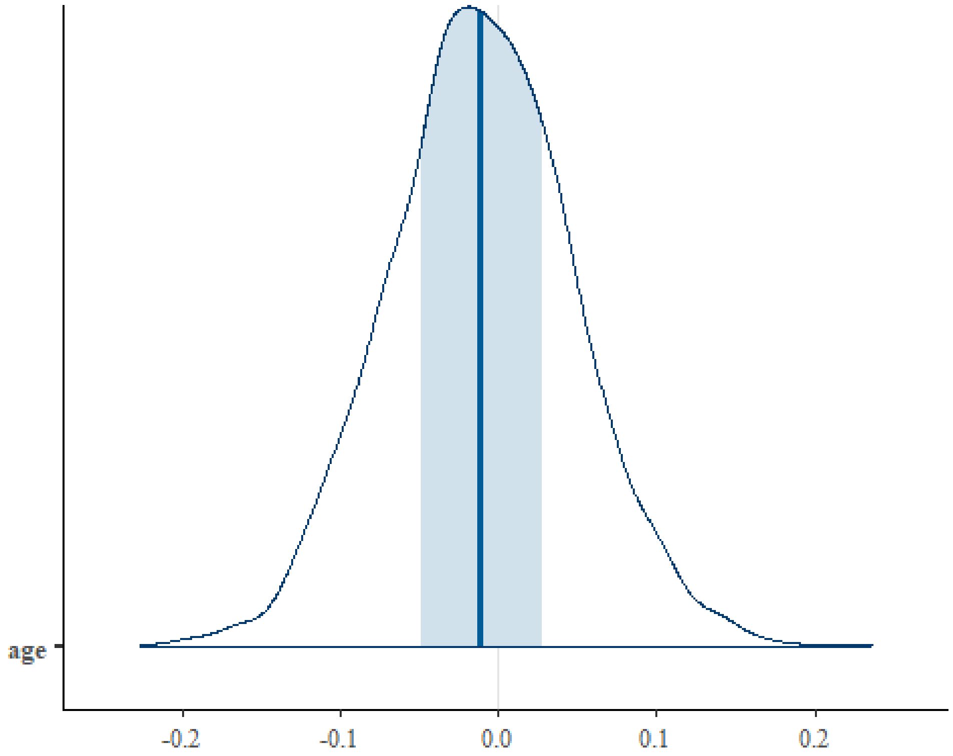
```
mcmc_areas(as.array(post1), pars = c('(Intercept)'))
```



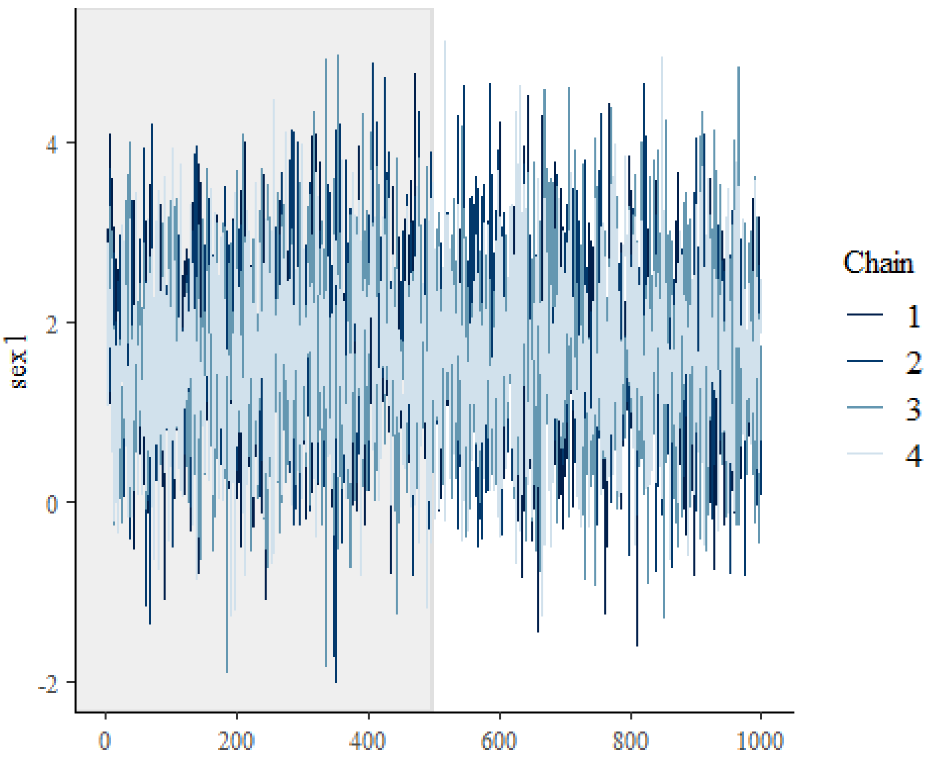
```
mcmc_trace(as.array(post1),  pars = c('age'), n_warmup = 500)
```



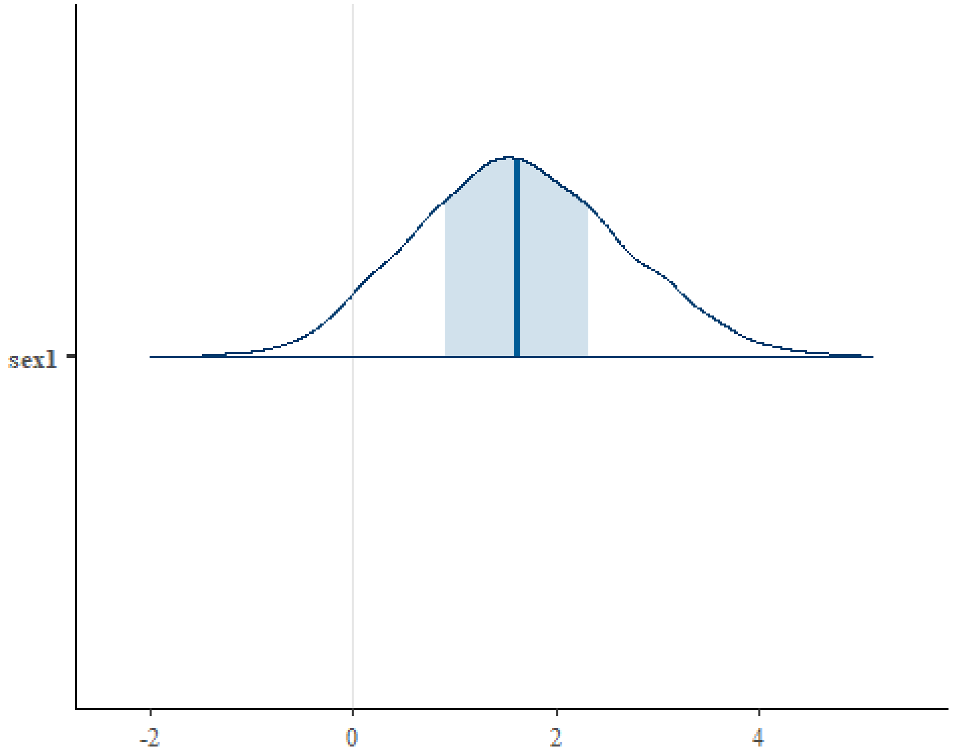
```
mcmc_areas(as.array(post1), pars = c('age'))
```



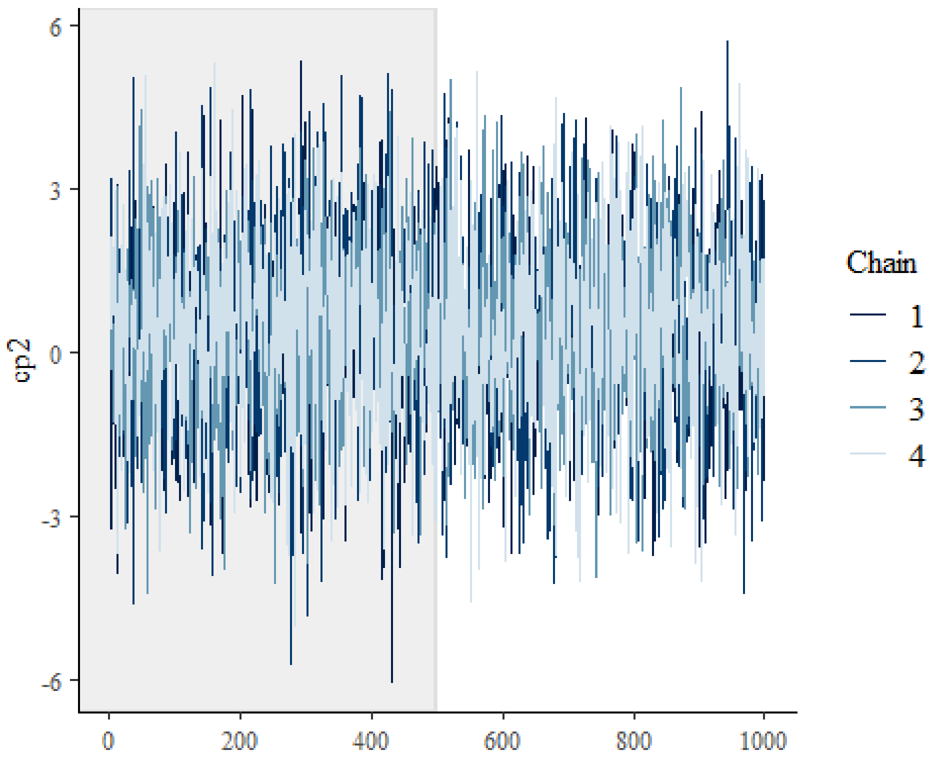
```
mcmc_trace(as.array(post1),  pars = c('sex1'), n_warmup = 500)
```



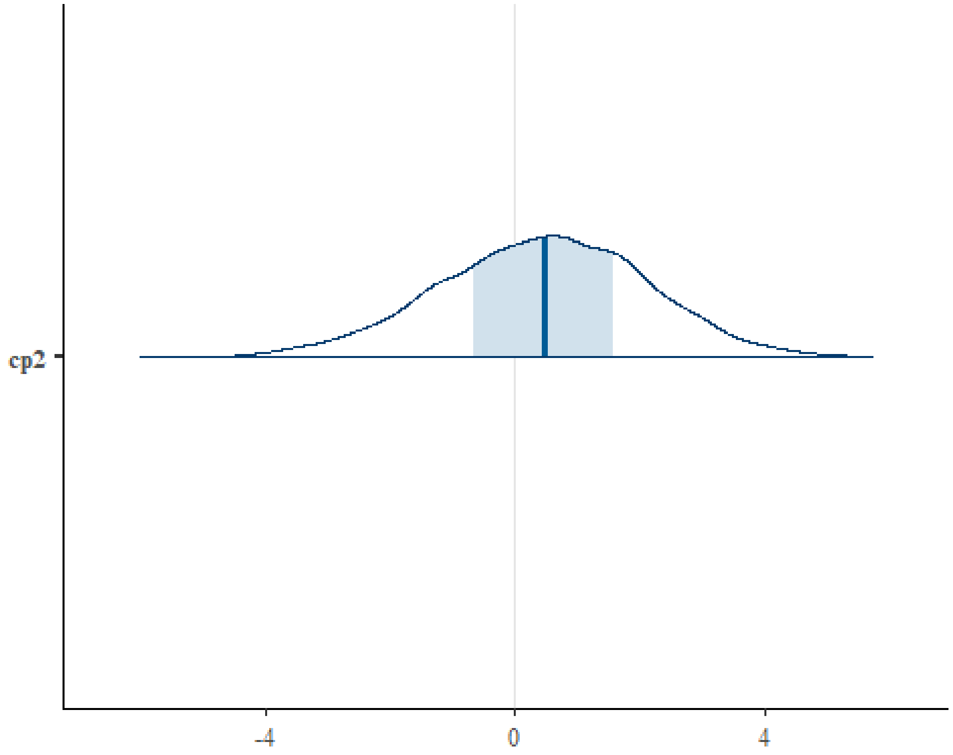
```
mcmc_areas(as.array(post1), pars = c('sex1'))
```



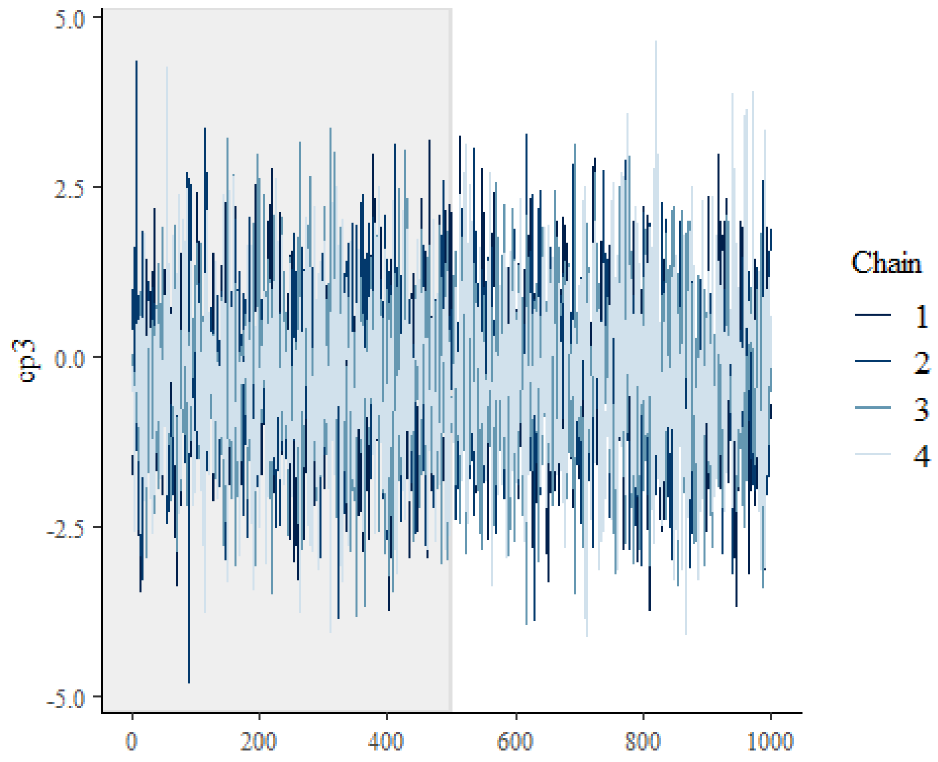
```
mcmc_trace(as.array(post1),  pars = c('cp2'), n_warmup = 500)
```



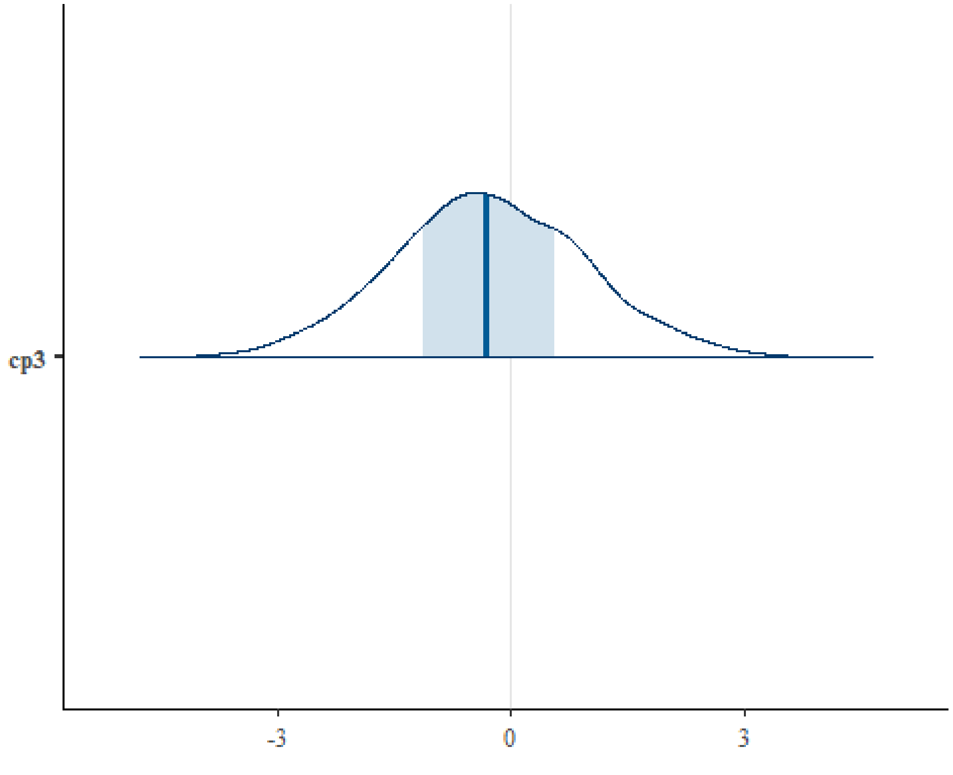
```
mcmc_areas(as.array(post1), pars = c('cp2'))
```



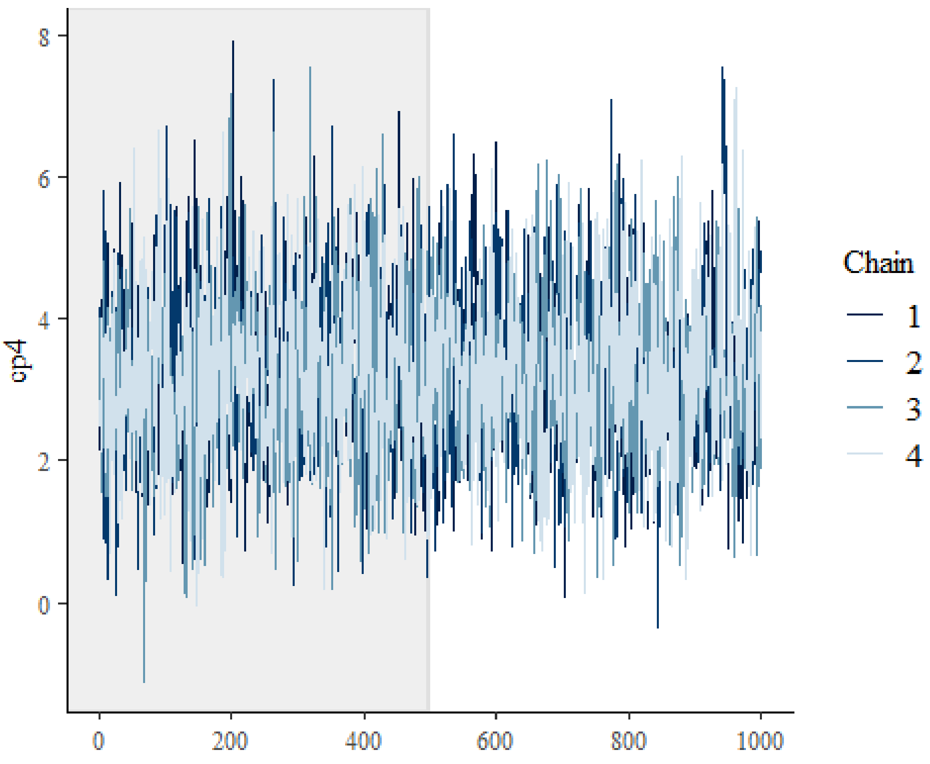
```
mcmc_trace(as.array(post1),  pars = c('cp3'), n_warmup = 500)
```



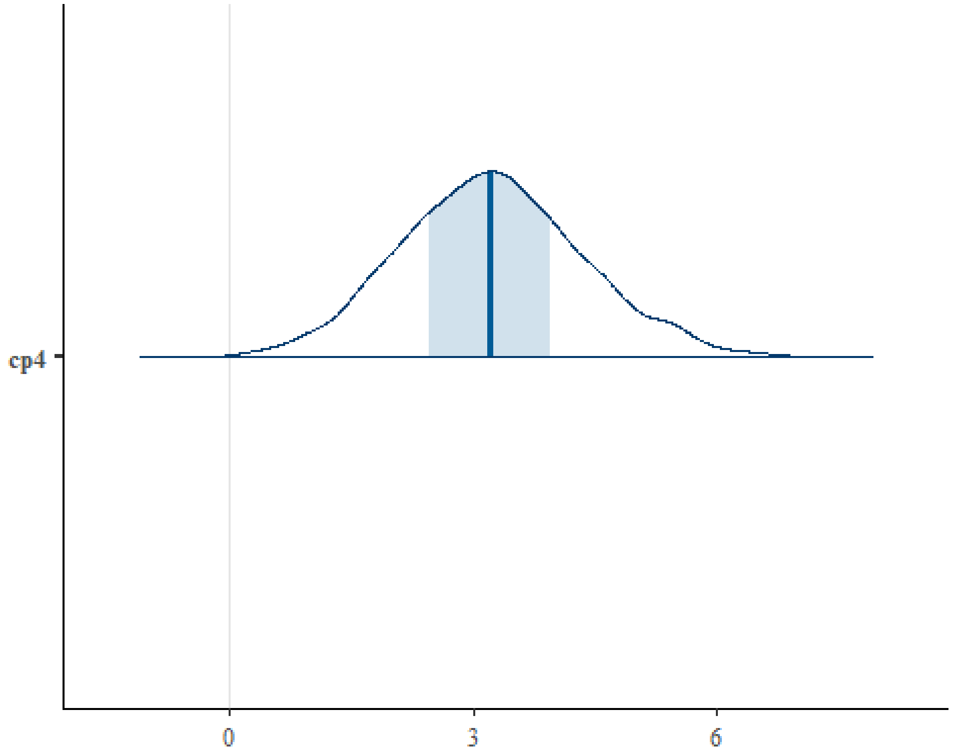
```
mcmc_areas(as.array(post1), pars = c('cp3'))
```



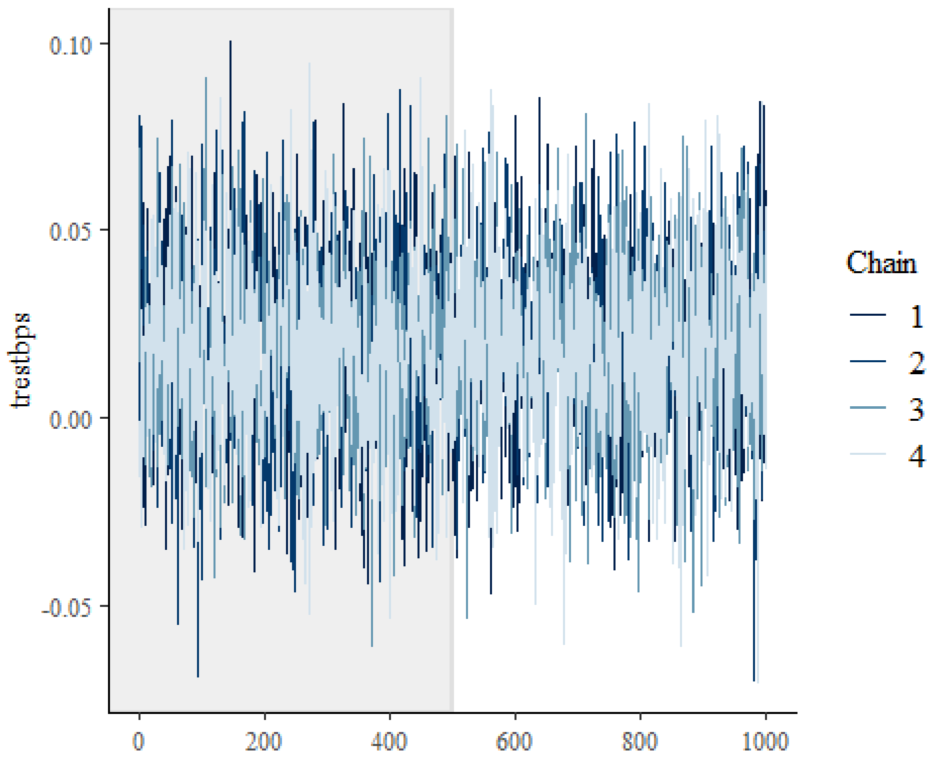
```
mcmc_trace(as.array(post1),  pars = c('cp4'), n_warmup = 500)
```



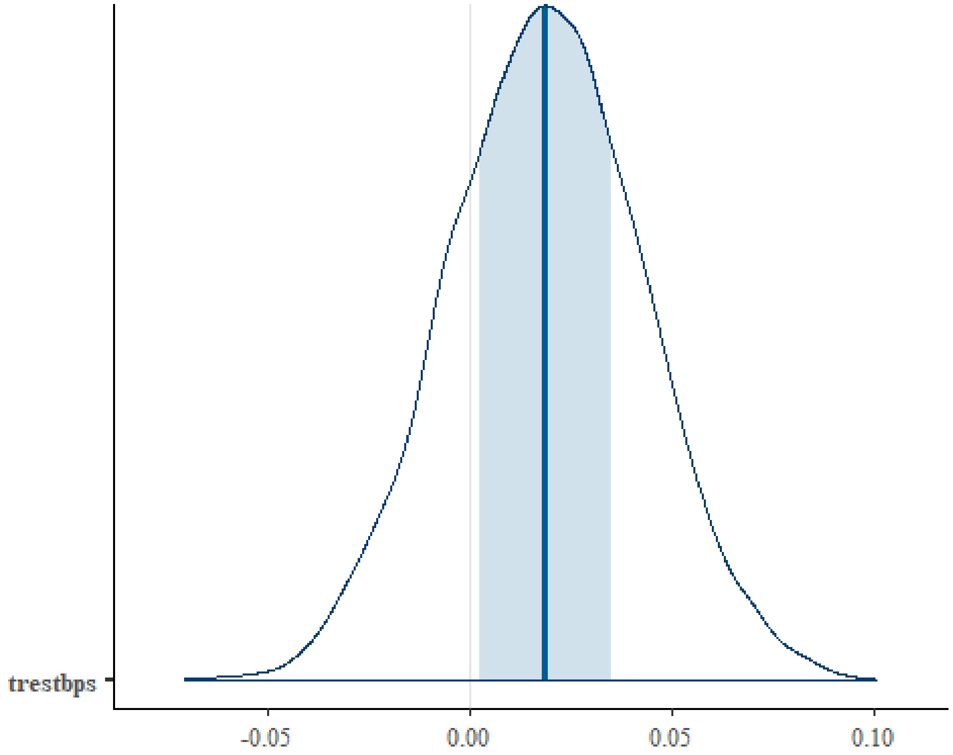
```r
mcmc_areas(as.array(post1), pars = c('cp4'))
```



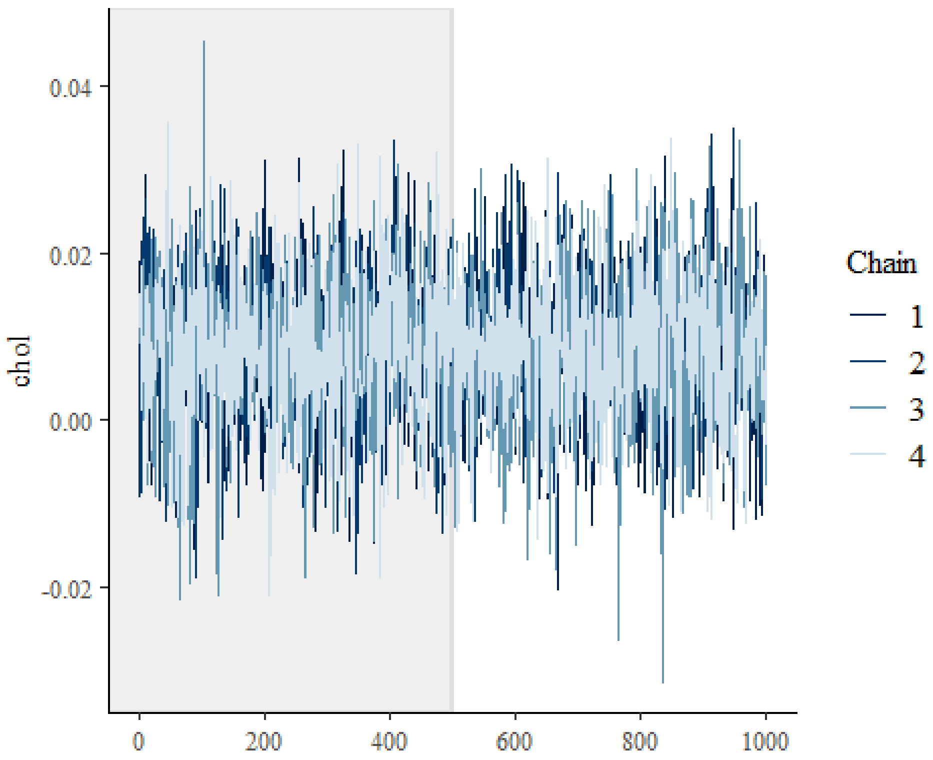
```r
mcmc_trace(as.array(post1),  pars = c('trestbps'), n_warmup = 500)
```



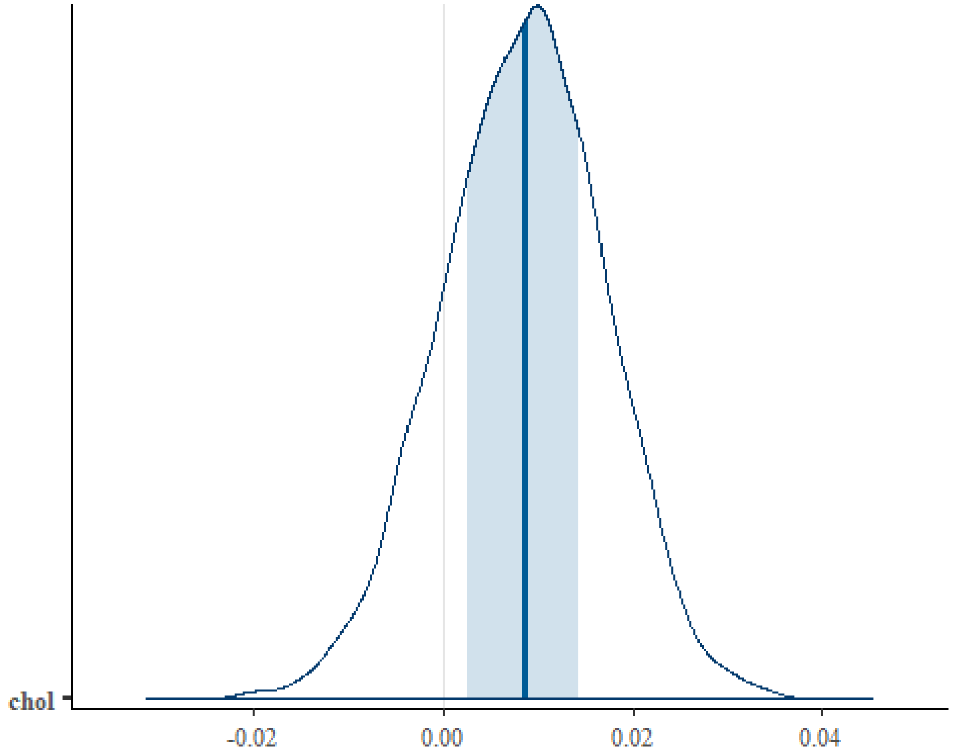
```
mcmc_areas(as.array(post1), pars = c('trestbps'))
```



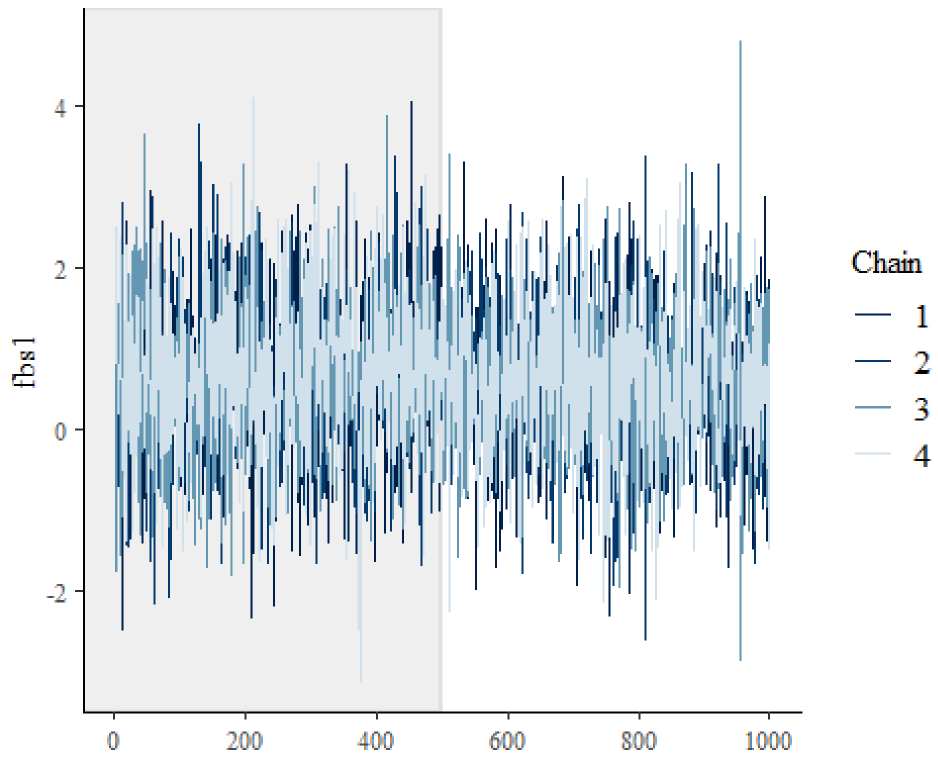
```
mcmc_trace(as.array(post1),  pars = c('chol'), n_warmup = 500)
```



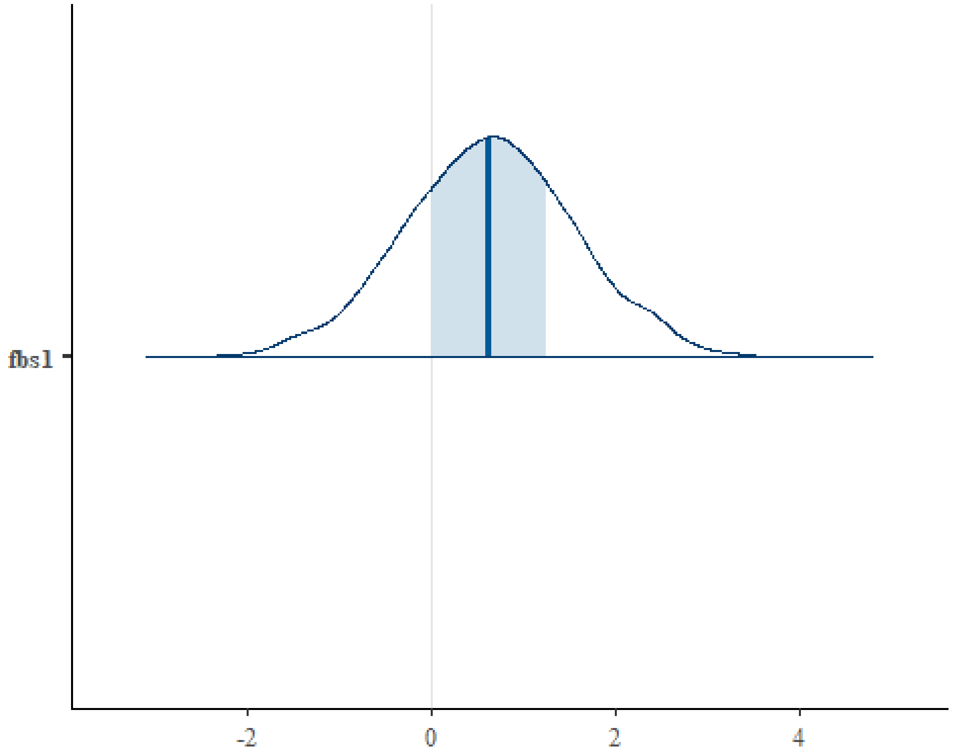
```r
mcmc_areas(as.array(post1), pars = c('chol'))
```



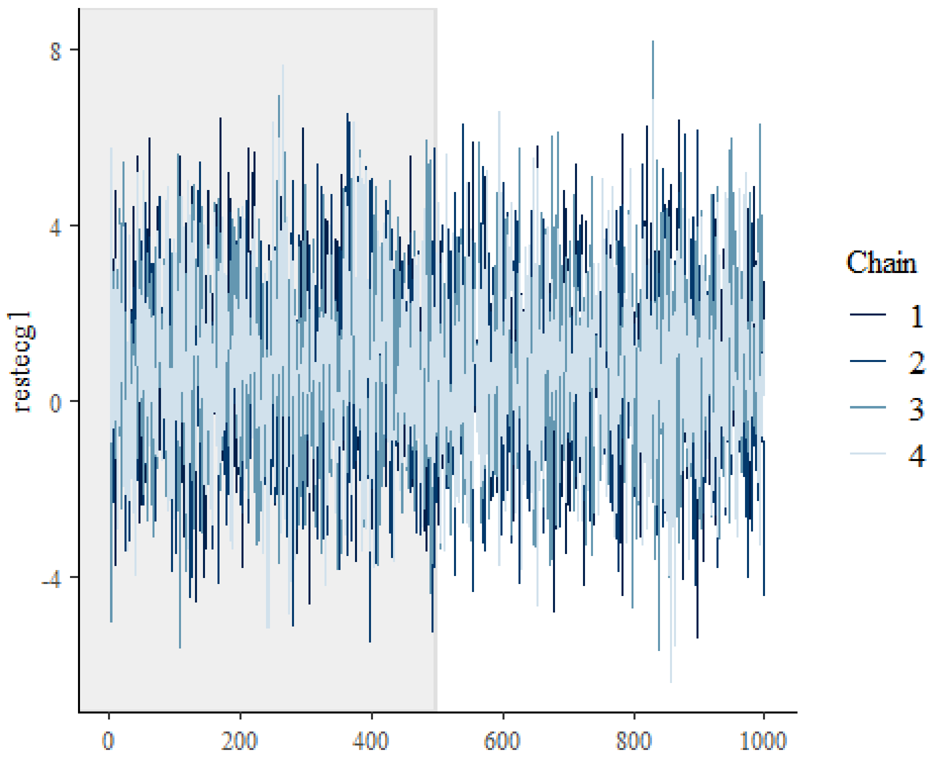
```r
mcmc_trace(as.array(post1),  pars = c('fbs1'), n_warmup = 500)
```



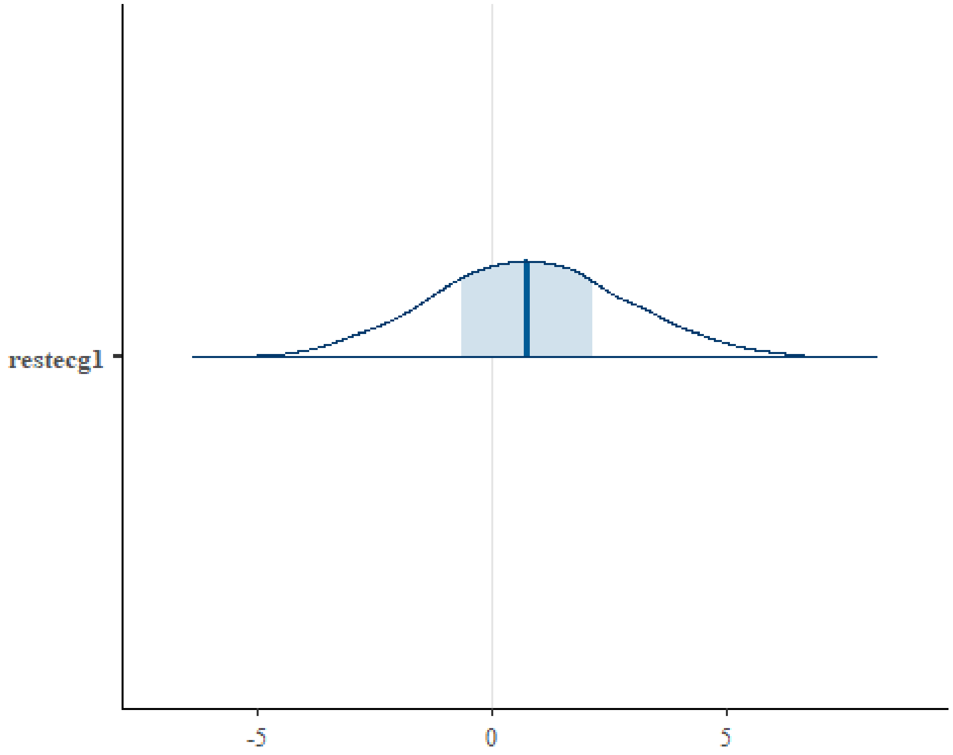
```
mcmc_areas(as.array(post1), pars = c('fbs1'))
```



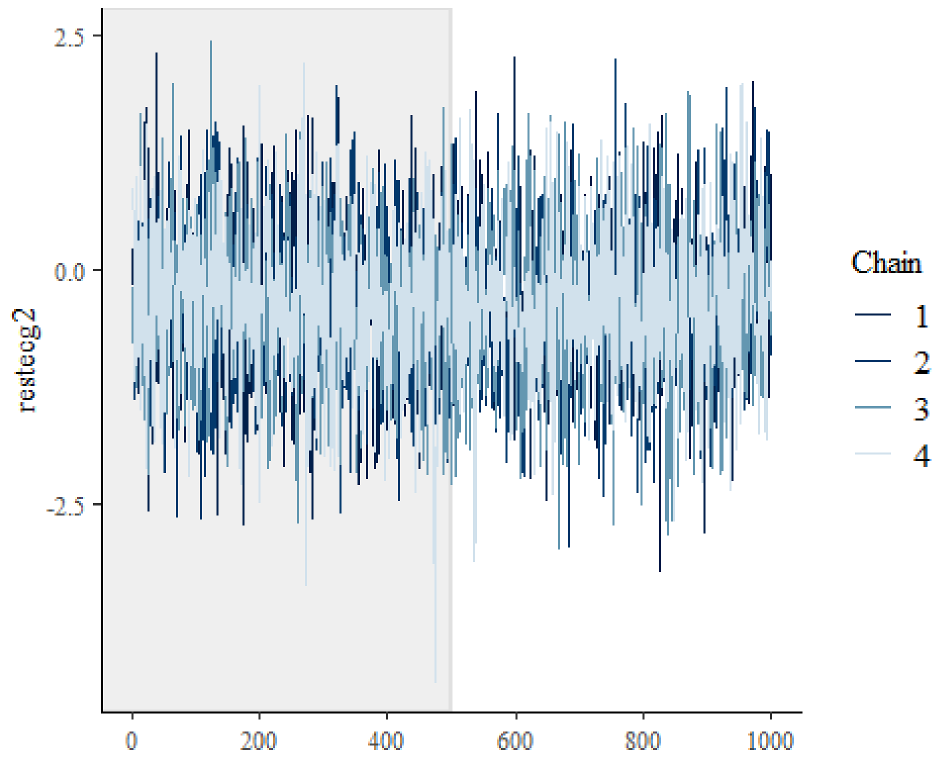
```
mcmc_trace(as.array(post1),  pars = c('restecg1'), n_warmup = 500)
```



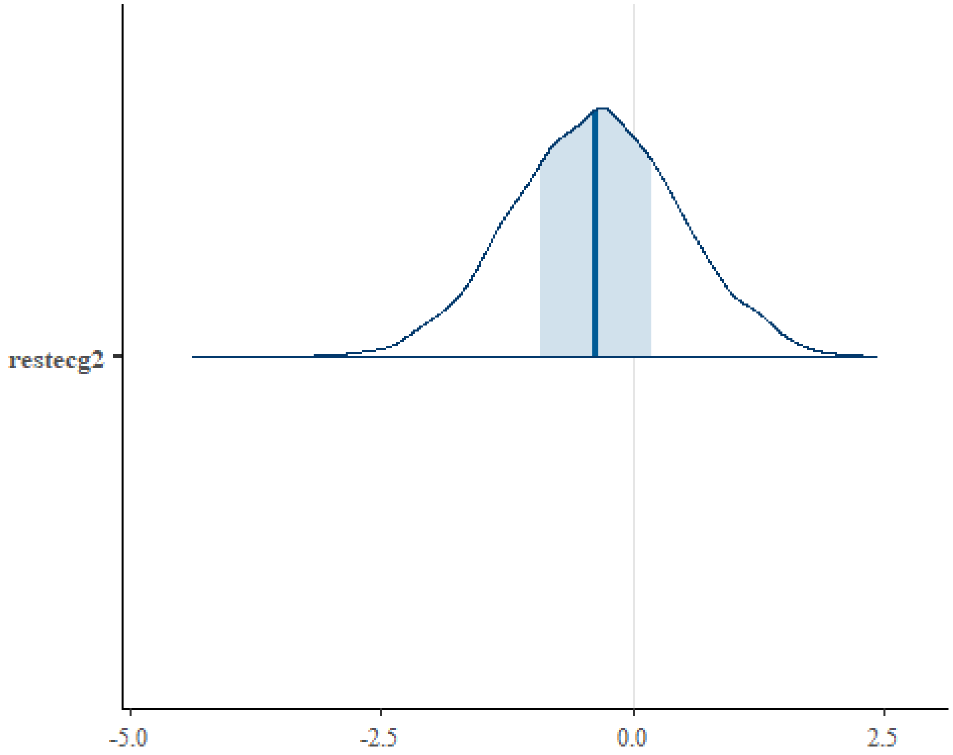
```
mcmc_areas(as.array(post1), pars = c('restecg1'))
```



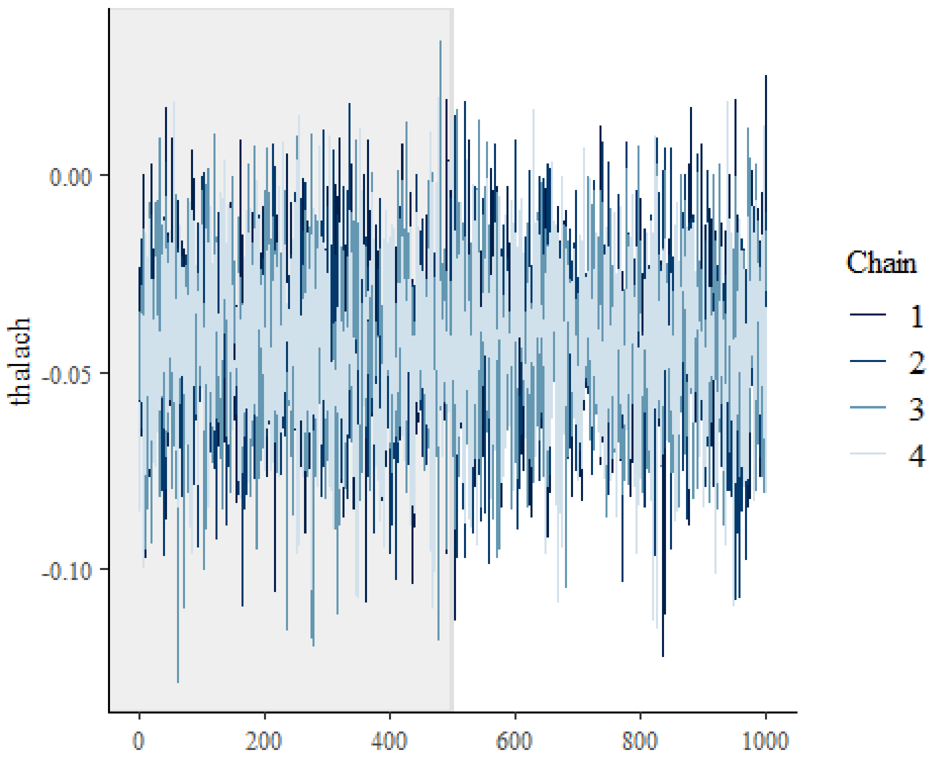
```
mcmc_trace(as.array(post1),  pars = c('restecg2'), n_warmup = 500)
```



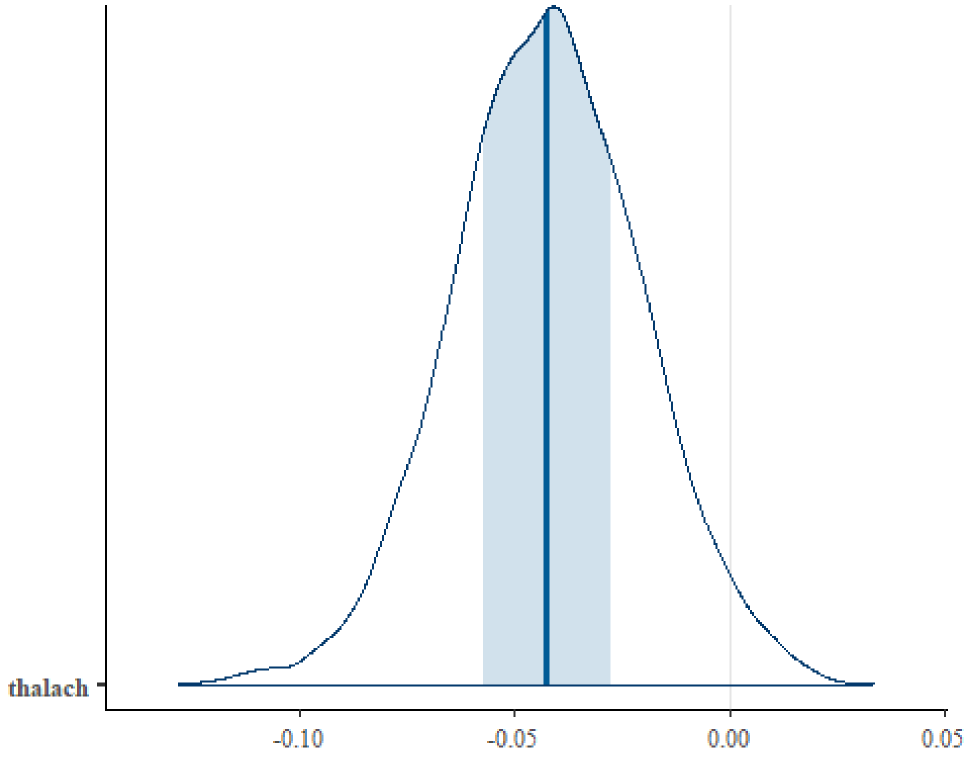
```r
mcmc_areas(as.array(post1), pars = c('restecg2'))
```



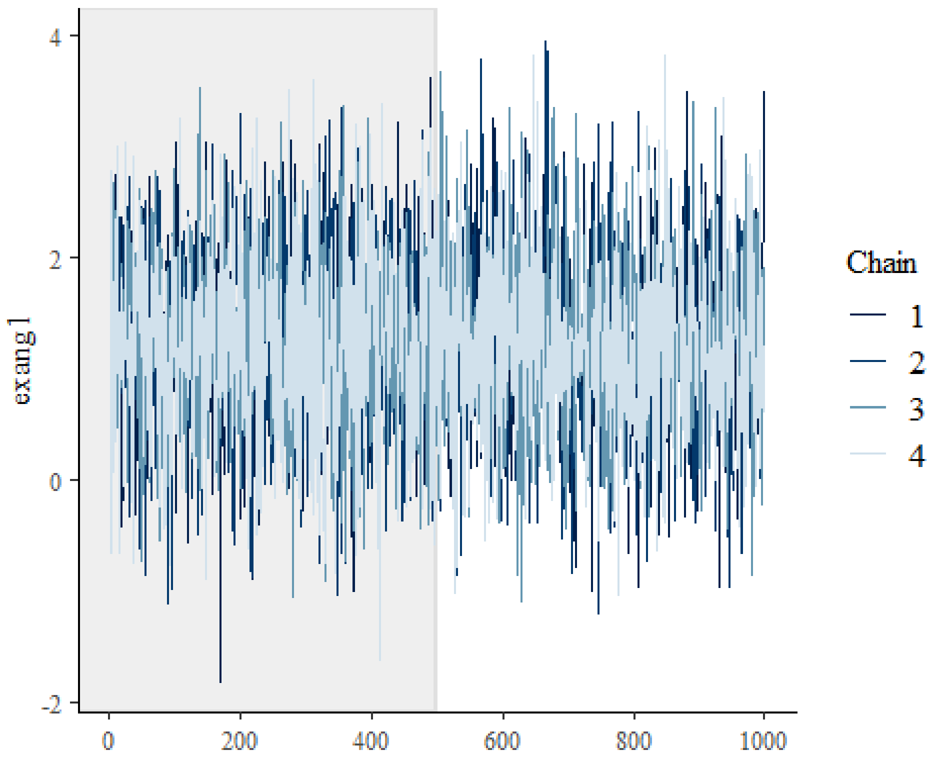
```r
mcmc_trace(as.array(post1),  pars = c('thalach'), n_warmup = 500)
```



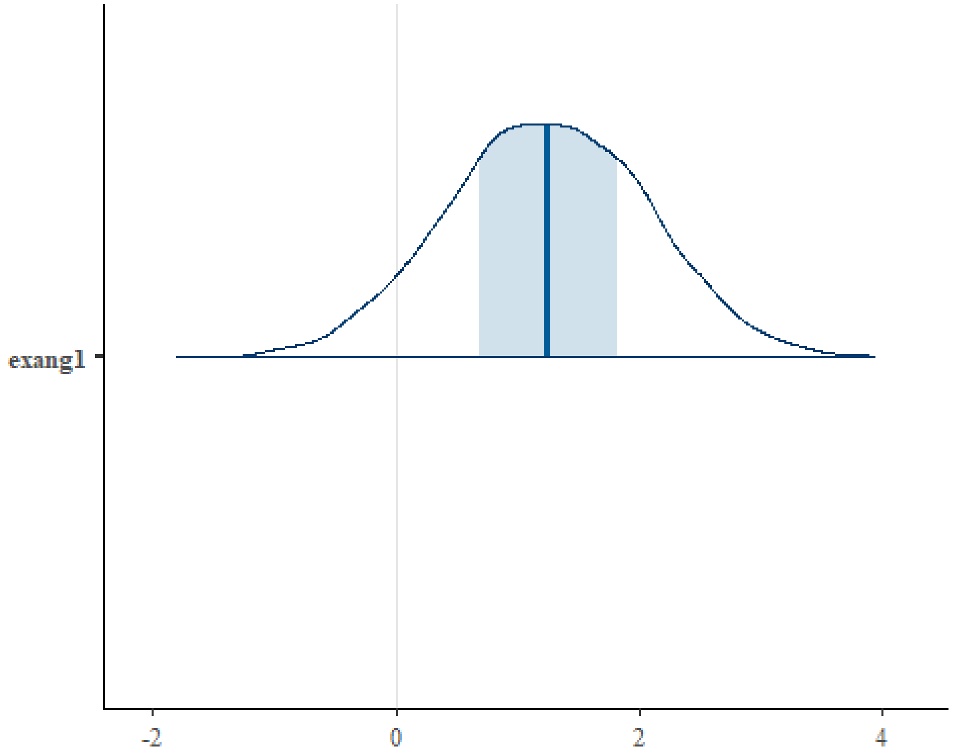
```
mcmc_areas(as.array(post1), pars = c('thalach'))
```



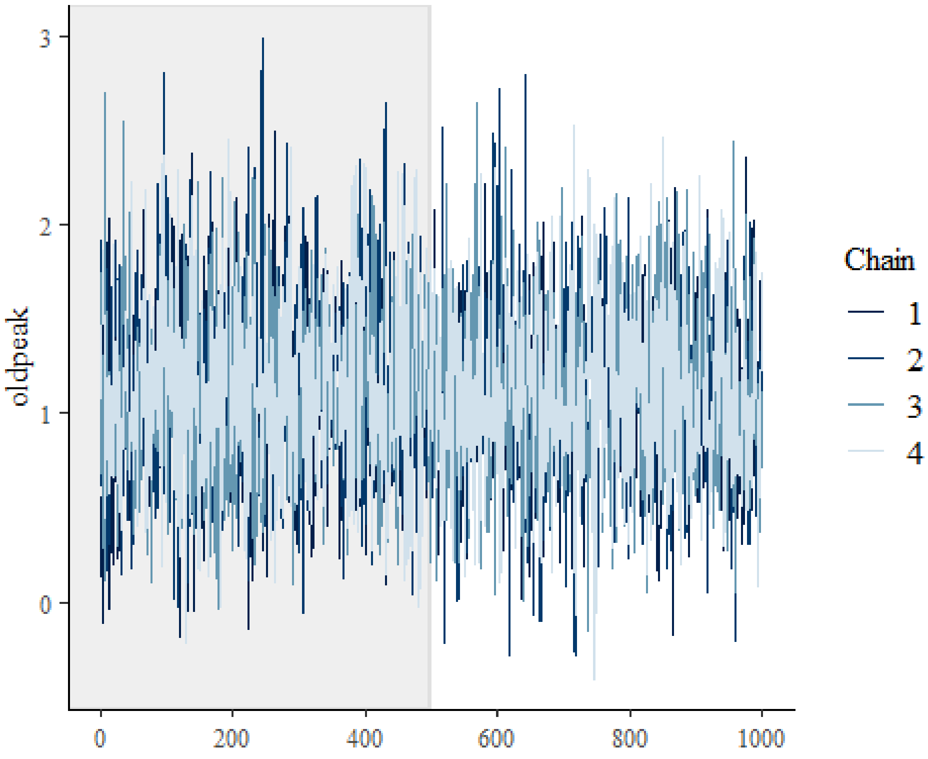
```
mcmc_trace(as.array(post1),  pars = c('exang1'), n_warmup = 500)
```



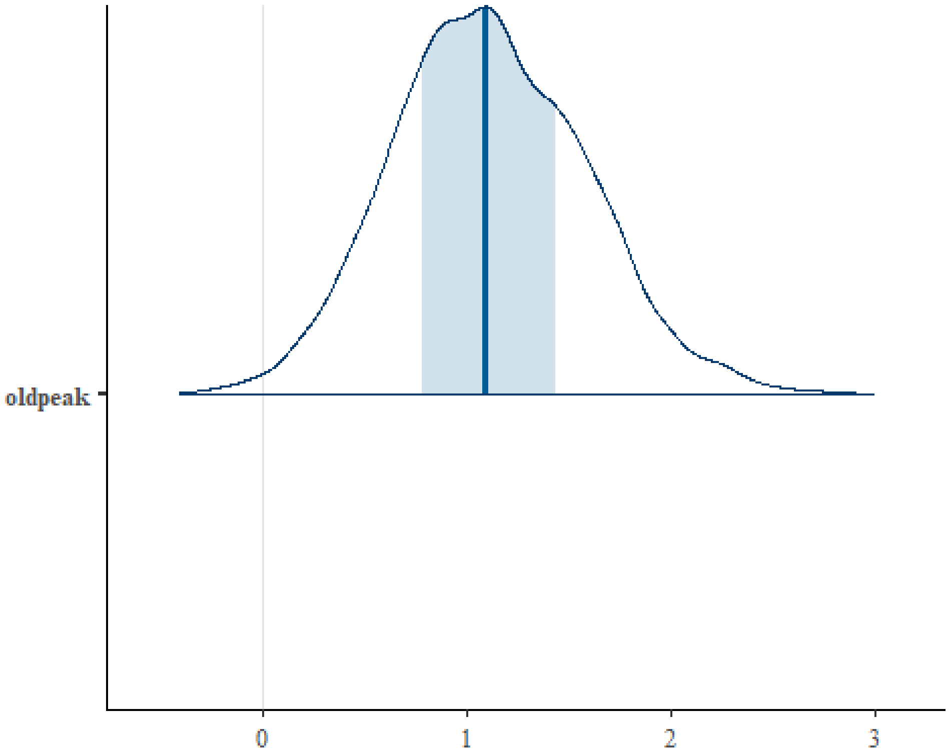
```r
mcmc_areas(as.array(post1), pars = c('exang1'))
```



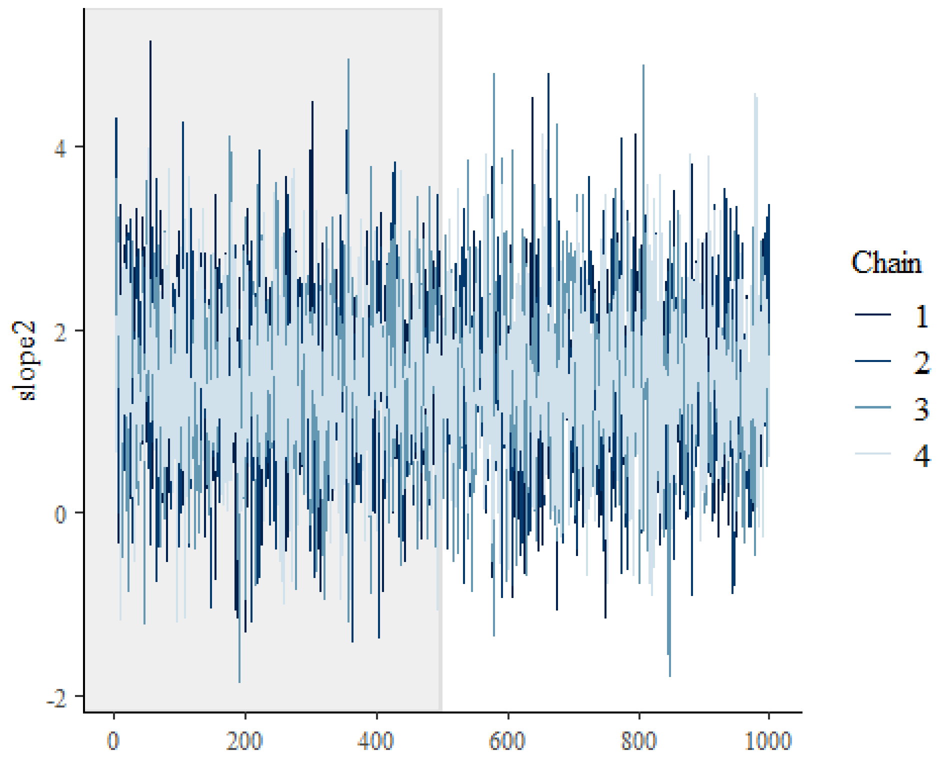
```r
mcmc_trace(as.array(post1),  pars = c('oldpeak'), n_warmup = 500)
```



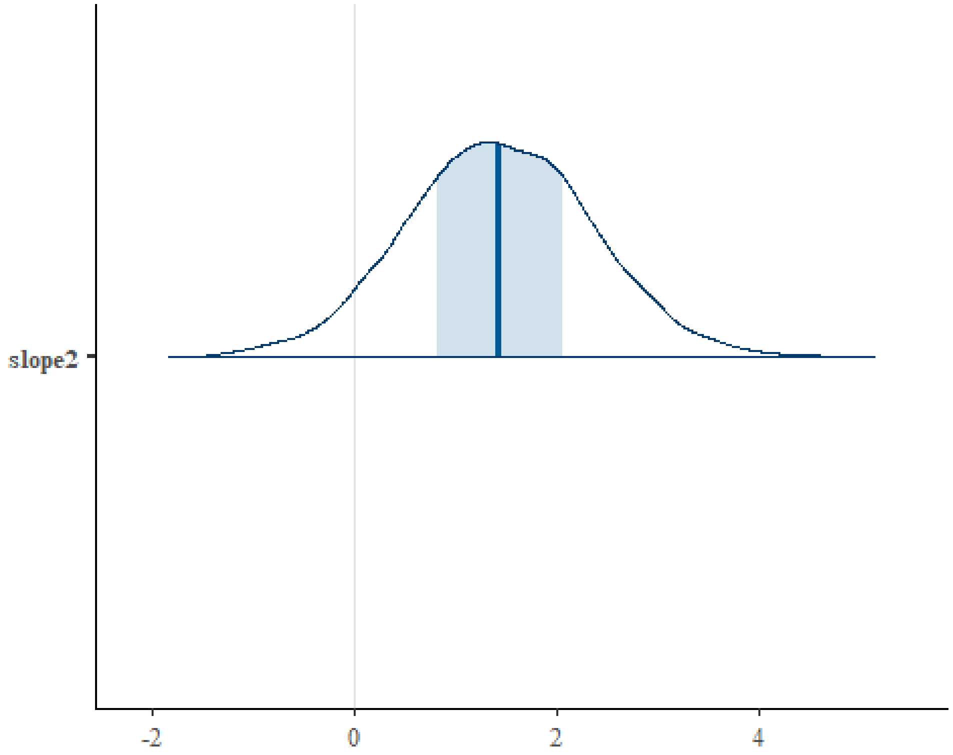
```
mcmc_areas(as.array(post1), pars = c('oldpeak'))
```



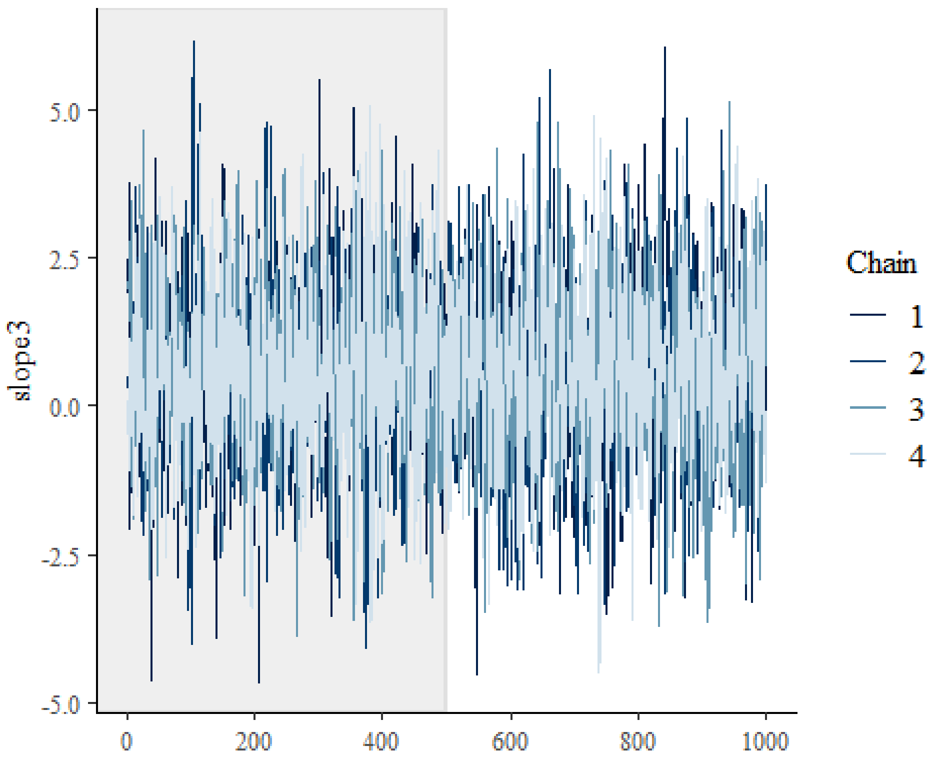
```
mcmc_trace(as.array(post1),  pars = c('slope2'), n_warmup = 500)
```



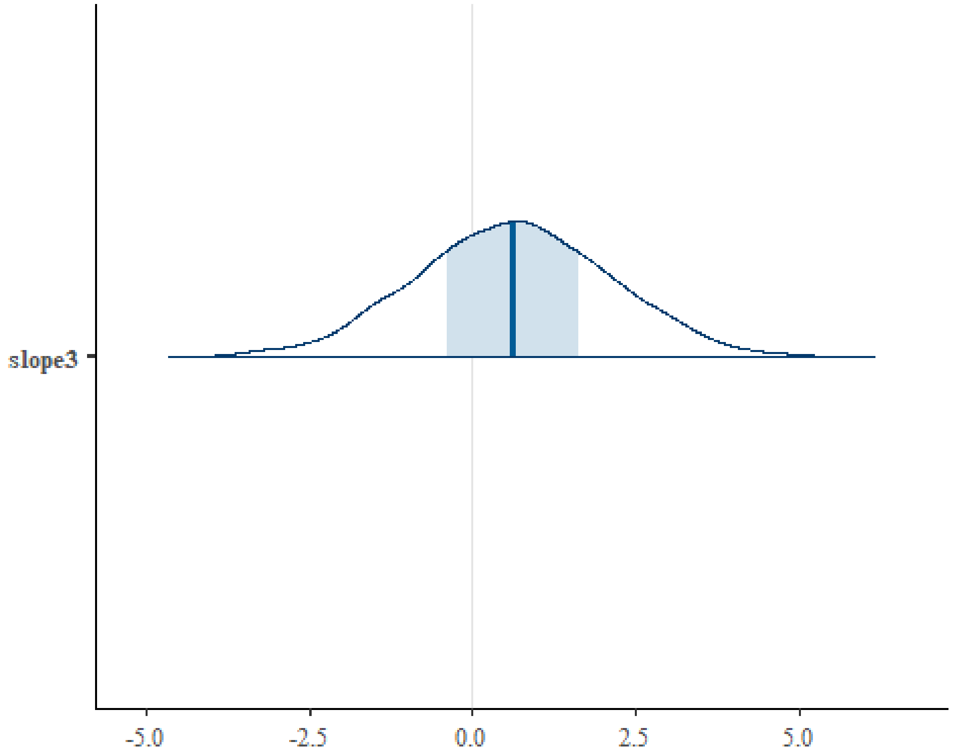
```
mcmc_areas(as.array(post1), pars = c('slope2'))
```



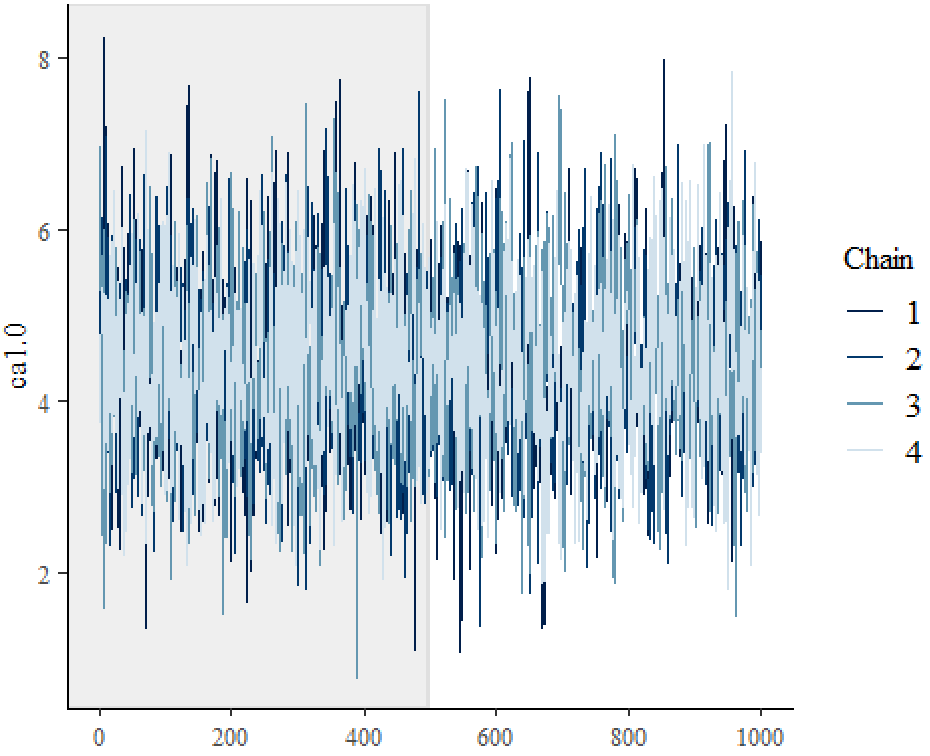
```
mcmc_trace(as.array(post1),  pars = c('slope3'), n_warmup = 500)
```



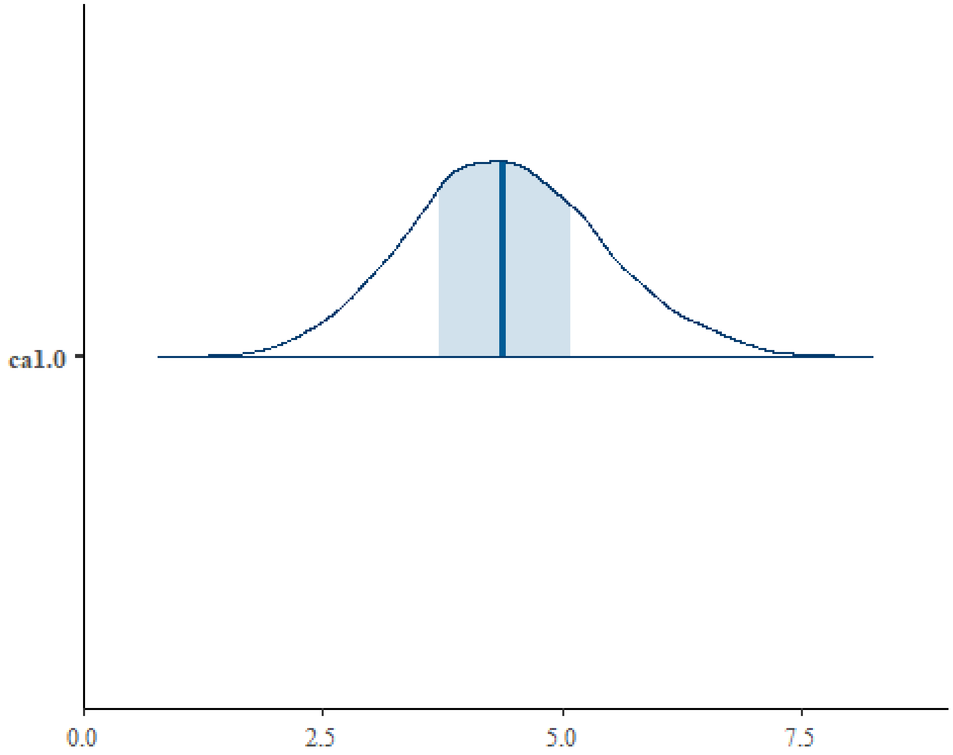
```r
mcmc_areas(as.array(post1), pars = c('slope3'))
```



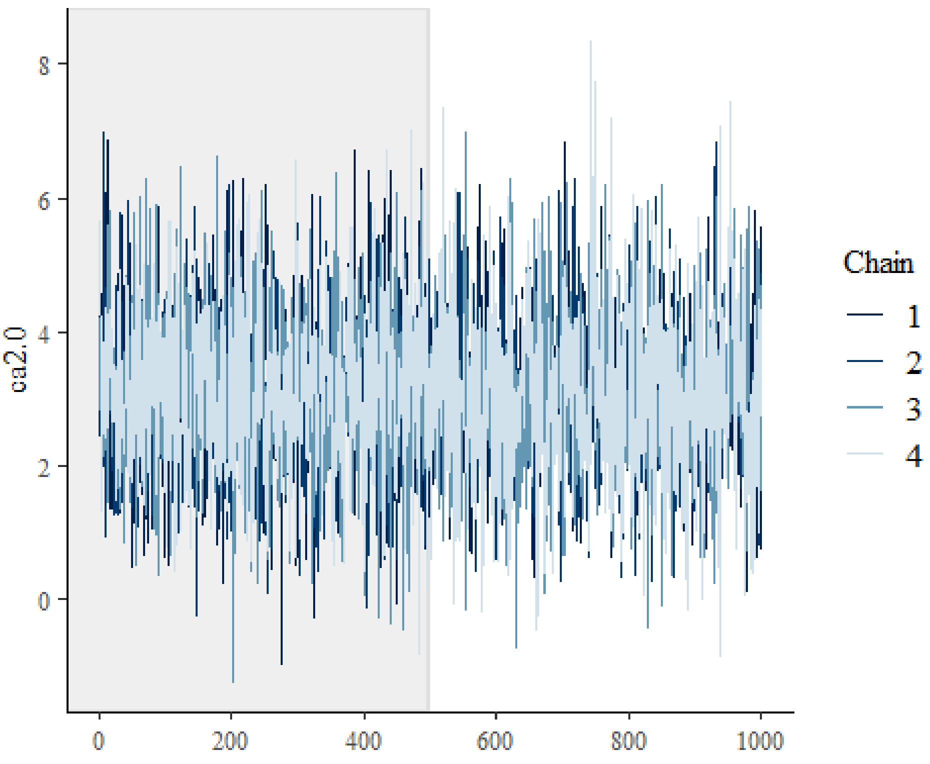
```r
mcmc_trace(as.array(post1),  pars = c('ca1.0'), n_warmup = 500)
```



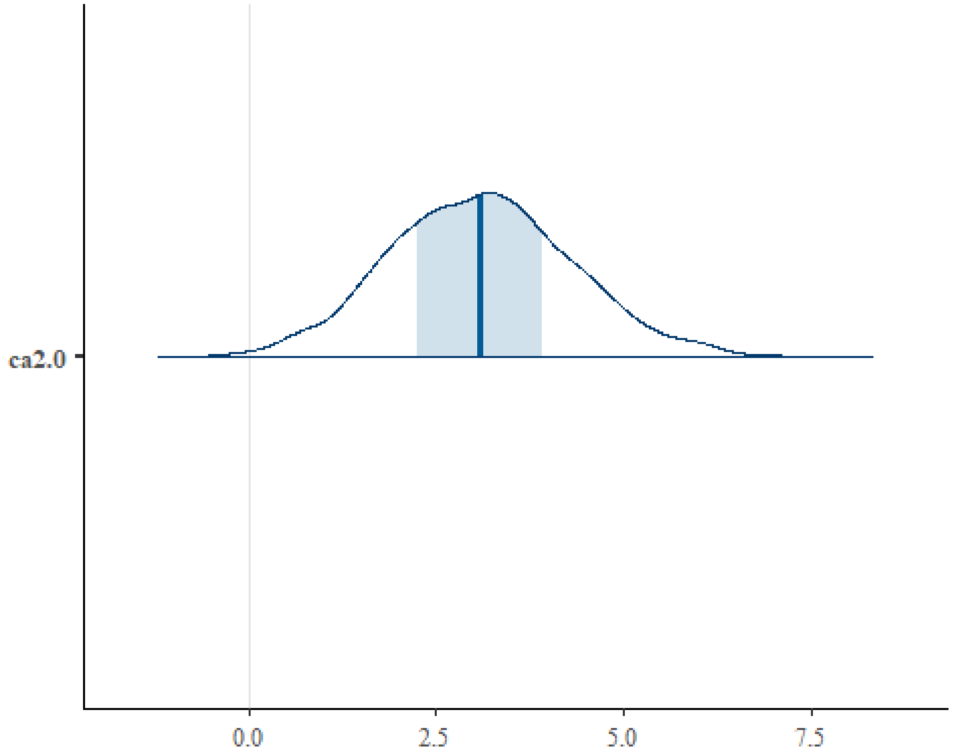
```
mcmc_areas(as.array(post1), pars = c('ca1.0'))
```



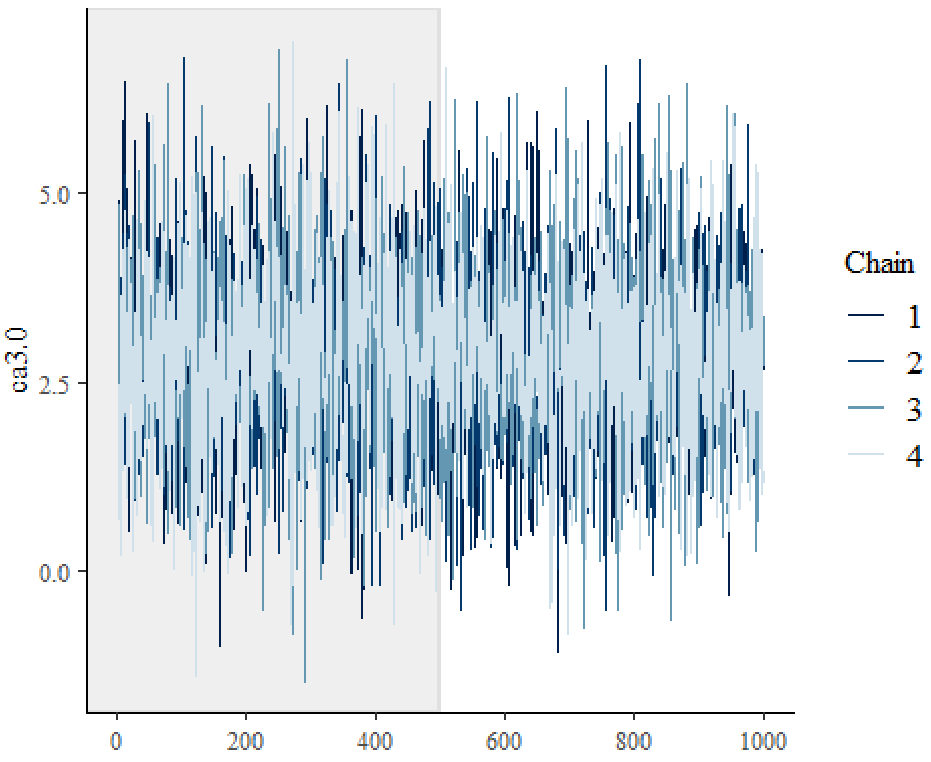
```
mcmc_trace(as.array(post1),  pars = c('ca2.0'), n_warmup = 500)
```



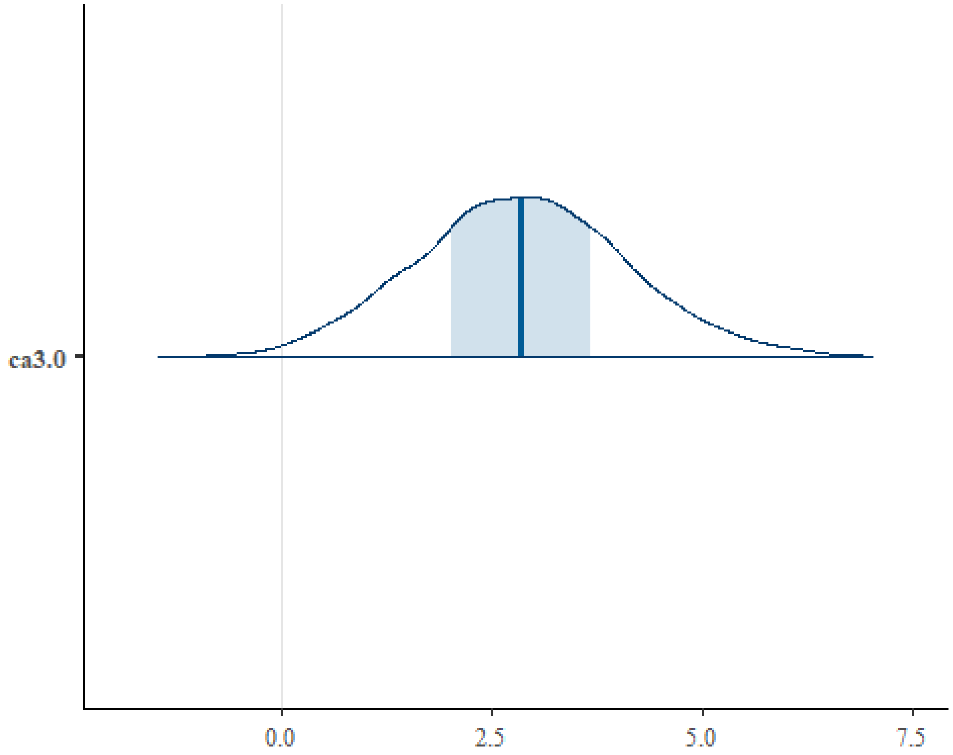
```
mcmc_areas(as.array(post1), pars = c('ca2.0'))
```



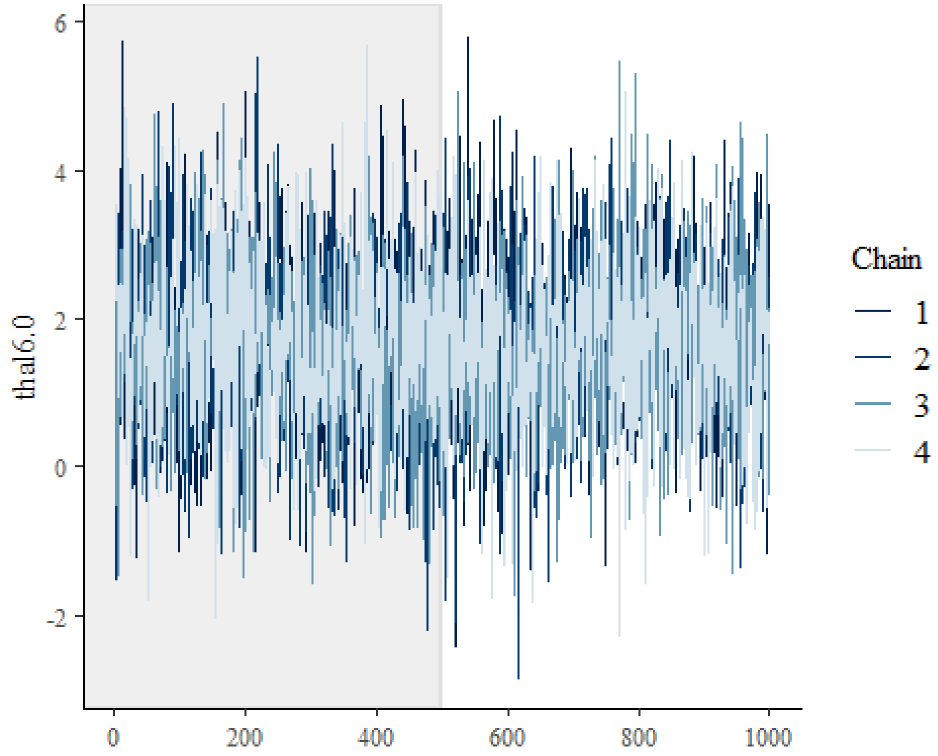
```
mcmc_trace(as.array(post1),  pars = c('ca3.0'), n_warmup = 500)
```



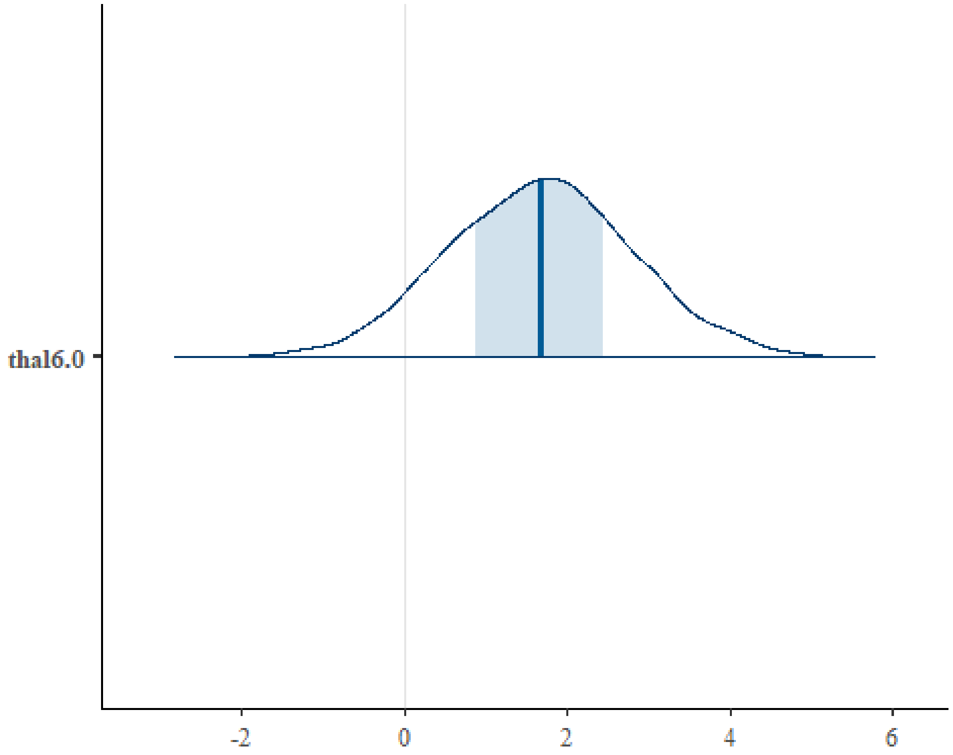
```
mcmc_areas(as.array(post1), pars = c('ca3.0'))
```



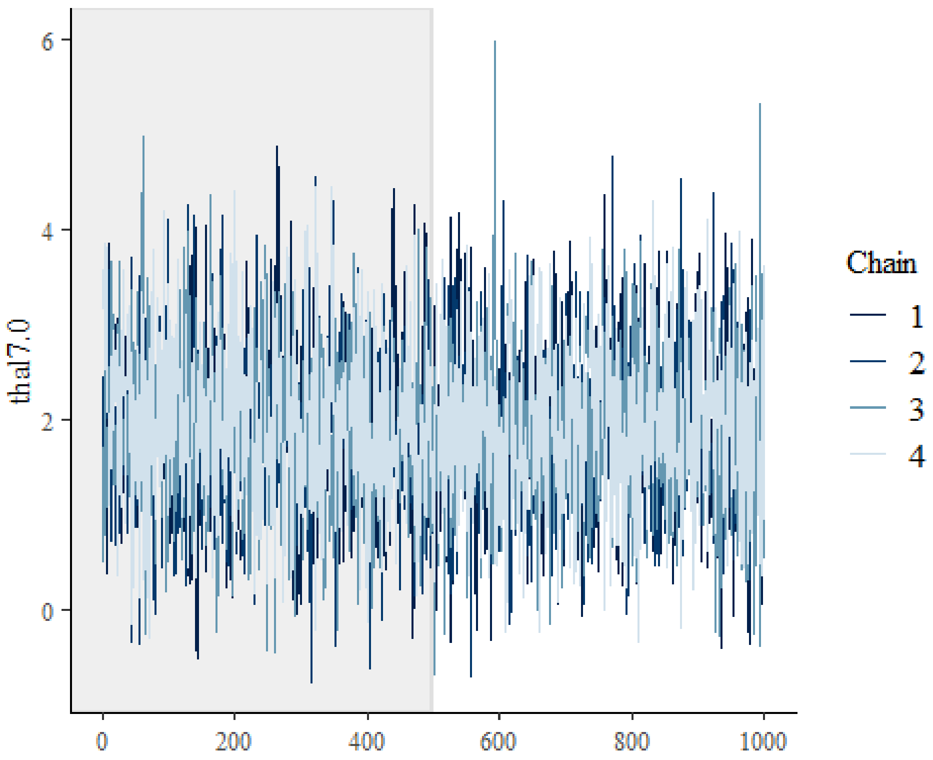
```
mcmc_trace(as.array(post1),  pars = c('thal6.0'), n_warmup = 500)
```



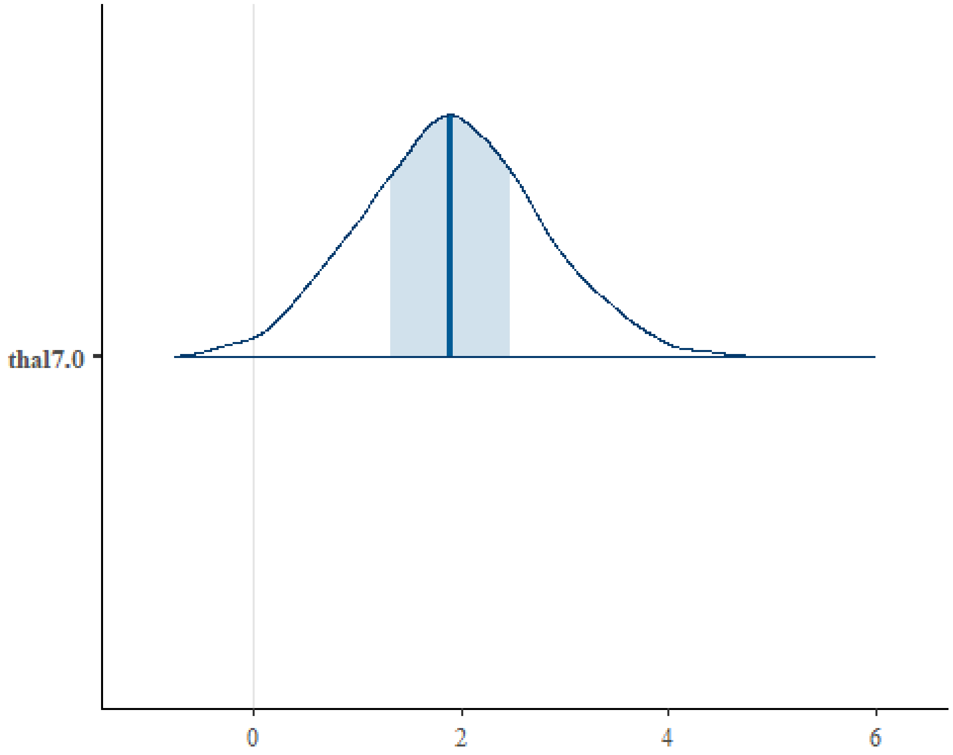
```
mcmc_areas(as.array(post1), pars = c('thal6.0'))
```



```
mcmc_trace(as.array(post1),  pars = c('thal7.0'), n_warmup = 500)
```



```r
mcmc_areas(as.array(post1), pars = c('thal7.0'))
```



Analyze with No Cardiovascular disease and Cardiovascular disease with severity = 3

```
# Taken the class with 0 and 1 only
data6 <- data2[data2['class'] == 0 | data2['class'] == 3,]
post4 <- stan_glm(class ~ ., data = data6,
                  family = binomial(link = "logit"),
                  prior = normal_prior, prior_intercept = normal_prior)

##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.535 seconds (Warm-up)
## Chain 1:                3.924 seconds (Sampling)
## Chain 1:                6.459 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 3.077 seconds (Warm-up)
## Chain 2:                2.72 seconds (Sampling)
## Chain 2:                5.797 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.705 seconds (Warm-up)
## Chain 3:                2.614 seconds (Sampling)
## Chain 3:                5.319 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 3.146 seconds (Warm-up)
## Chain 4:                2.538 seconds (Sampling)
## Chain 4:                5.684 seconds (Total)
## Chain 4:
```

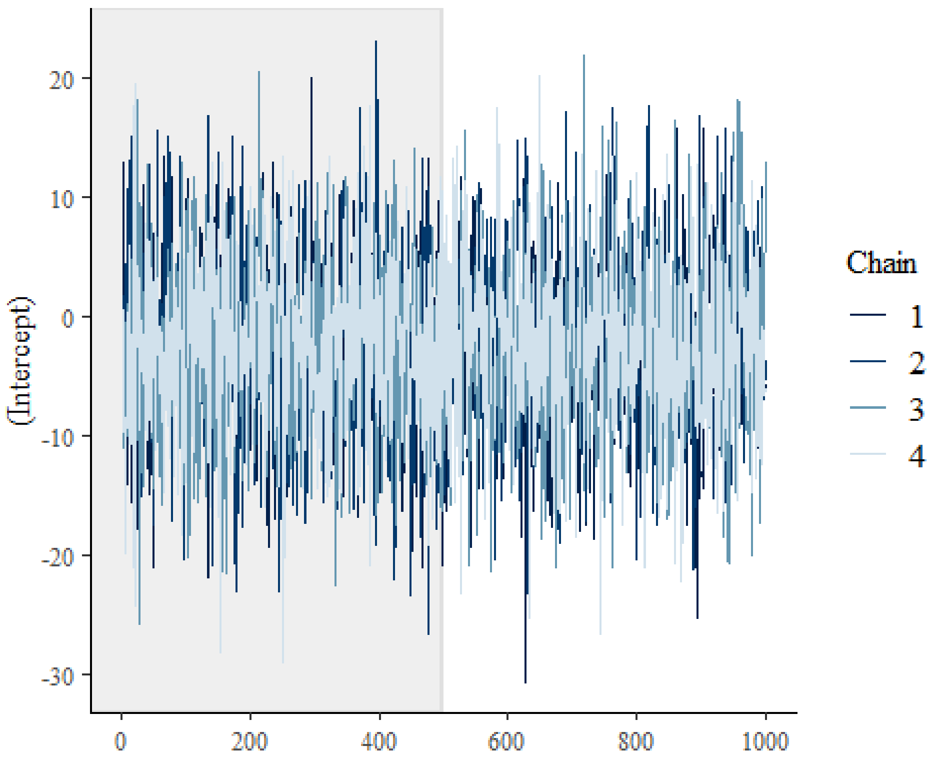
```
round(coef(post4), 2)
```

```
## (Intercept)         age        sex1         cp2         cp3         cp4 
##       -2.29       -0.11        0.95        2.47        0.77        3.22 
##    trestbps        chol        fbs1    restecg1    restecg2     thalach 
##        0.02        0.00        1.94        0.79        2.33       -0.05 
##      exang1     oldpeak      slope2      slope3       ca1.0       ca2.0 
##        0.99        1.18        2.00        0.48        2.28        5.91 
##       ca3.0     thal6.0     thal7.0 
##        1.88       -0.63        3.97
```

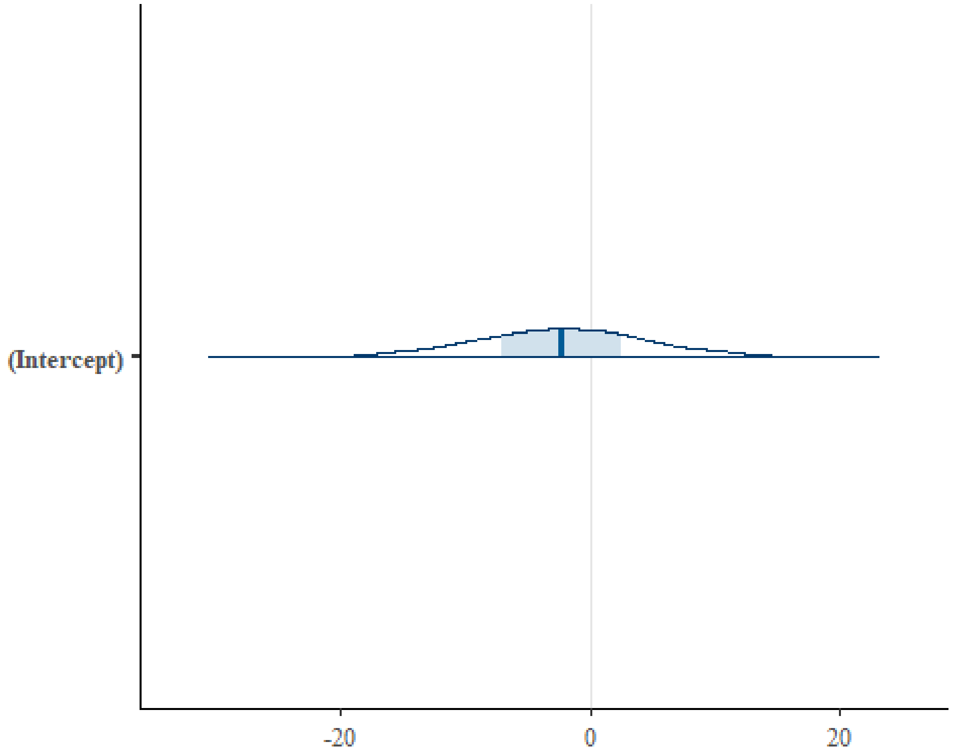
```
round(posterior_interval(post4, prob = 0.95), 2)
```

```
##               2.5% 97.5%
## (Intercept) -16.57 11.59
## age          -0.25  0.03
## sex1         -1.41  3.47
## cp2          -0.81  5.52
## cp3          -1.80  3.47
## cp4           0.71  5.86
## trestbps     -0.03  0.08
## chol         -0.02  0.02
## fbs1         -0.31  4.45
## restecg1     -3.29  5.10
## restecg2      0.44  4.43
## thalach      -0.11  0.01
## exang1       -1.19  3.14
## oldpeak       0.31  2.16
## slope2        0.12  3.99
## slope3       -2.85  3.56
## ca1.0        -0.13  4.66
## ca2.0         3.47  8.51
## ca3.0        -0.73  4.61
## thal6.0      -4.85  3.19
## thal7.0       1.89  6.34
```

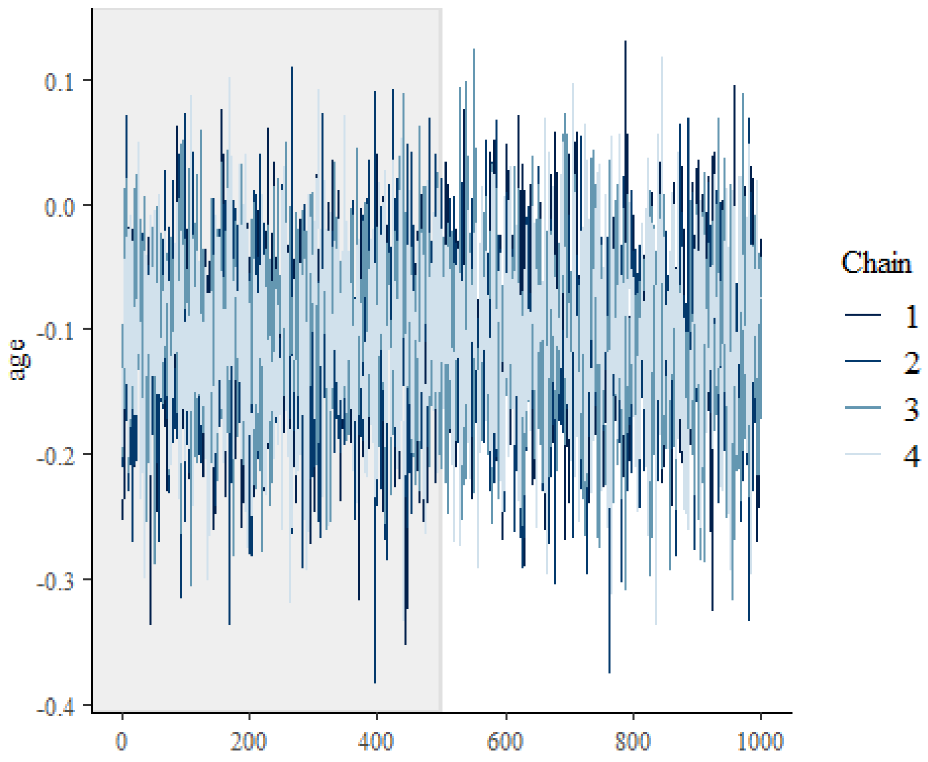
```
post1 <- post4  # to make reuse of the code of plotting

mcmc_trace(as.array(post1),  pars = c('(Intercept)'), n_warmup = 500)
```



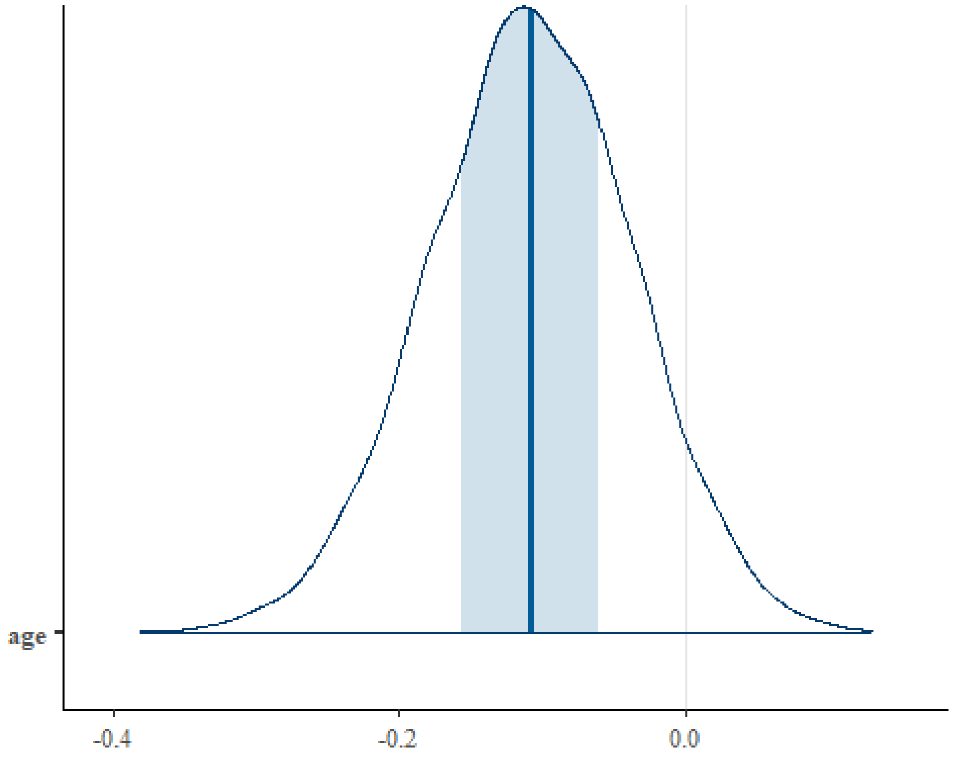
```
mcmc_areas(as.array(post1), pars = c('(Intercept)'))
```



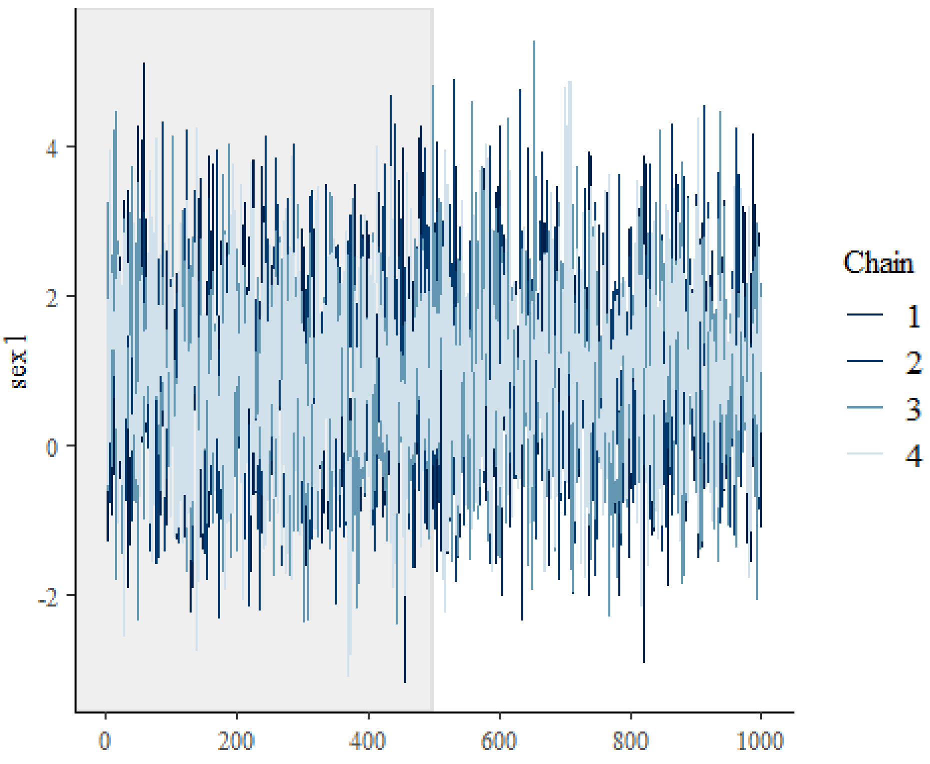
```
mcmc_trace(as.array(post1),  pars = c('age'), n_warmup = 500)
```



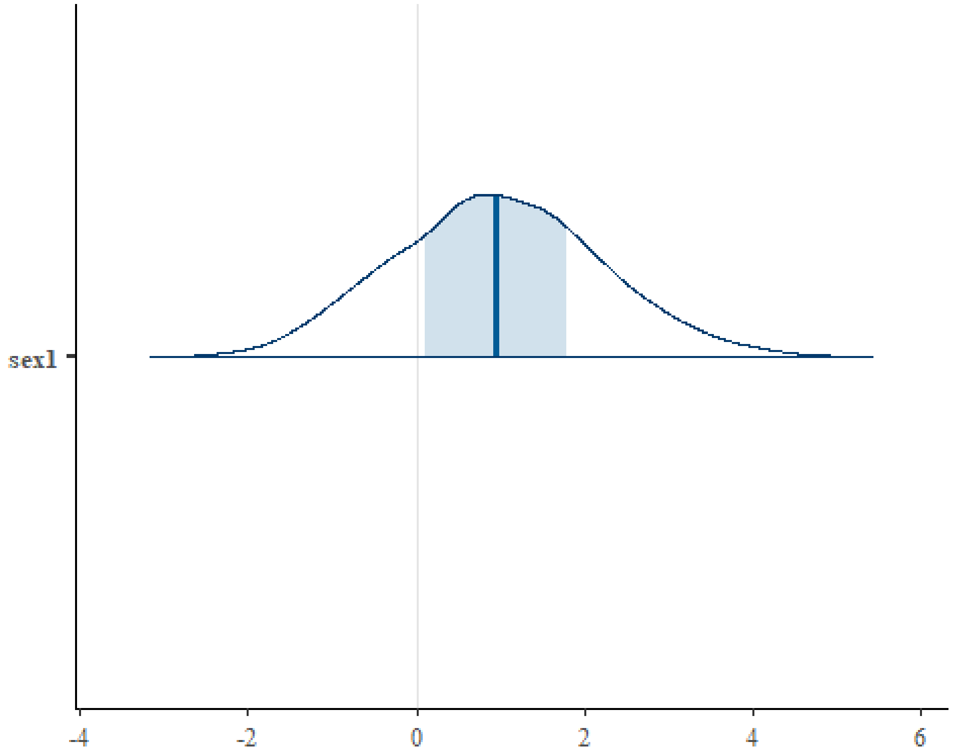
```
mcmc_areas(as.array(post1), pars = c('age'))
```



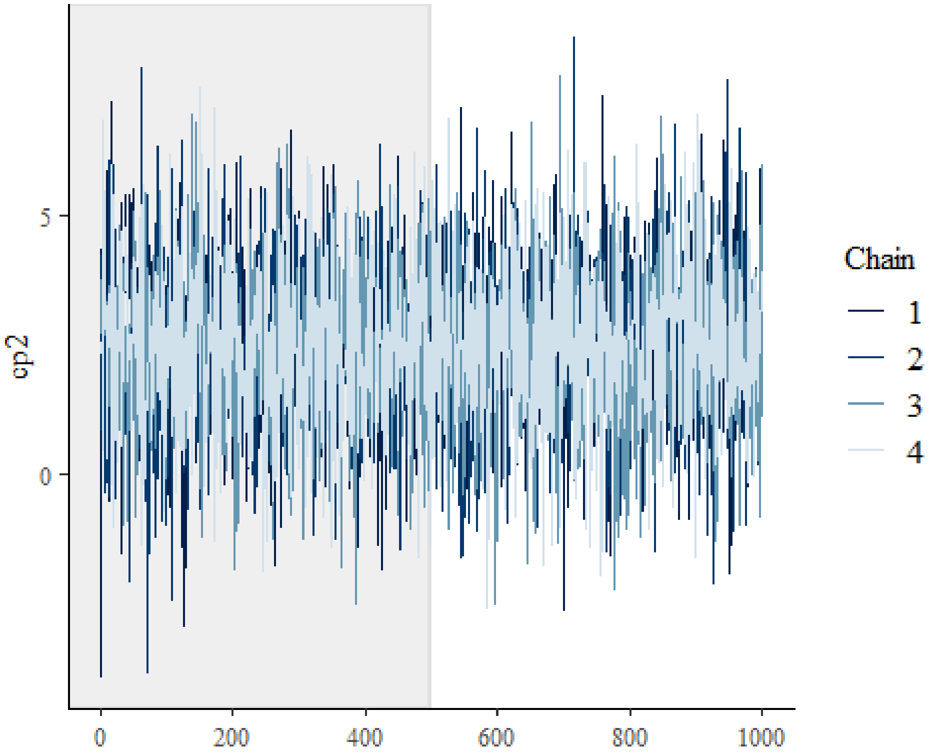
```
mcmc_trace(as.array(post1),  pars = c('sex1'), n_warmup = 500)
```



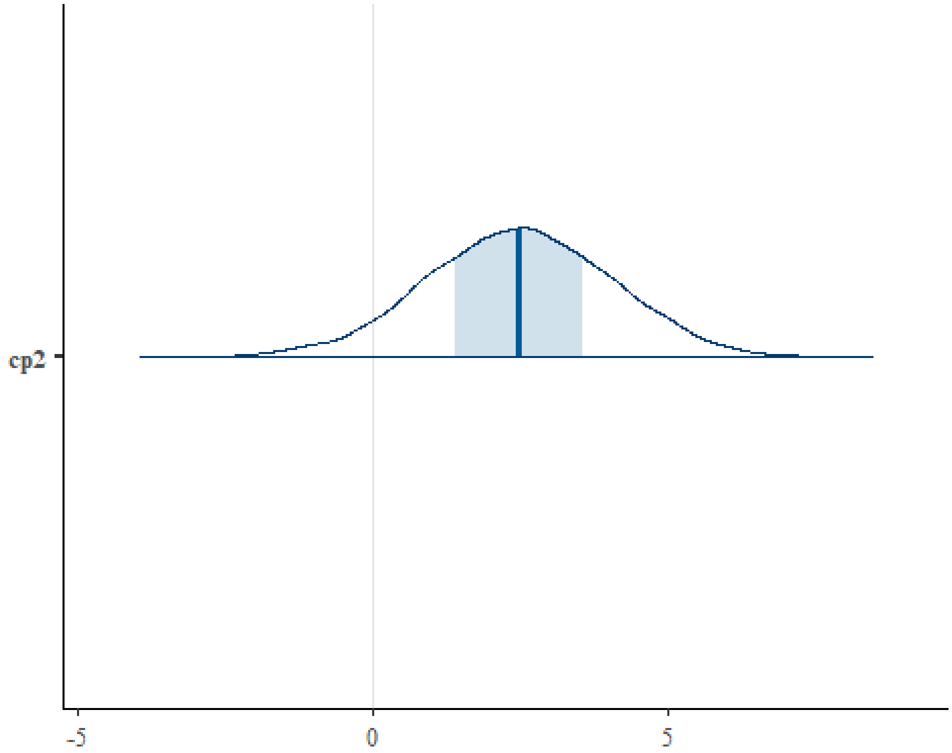
```
mcmc_areas(as.array(post1), pars = c('sex1'))
```



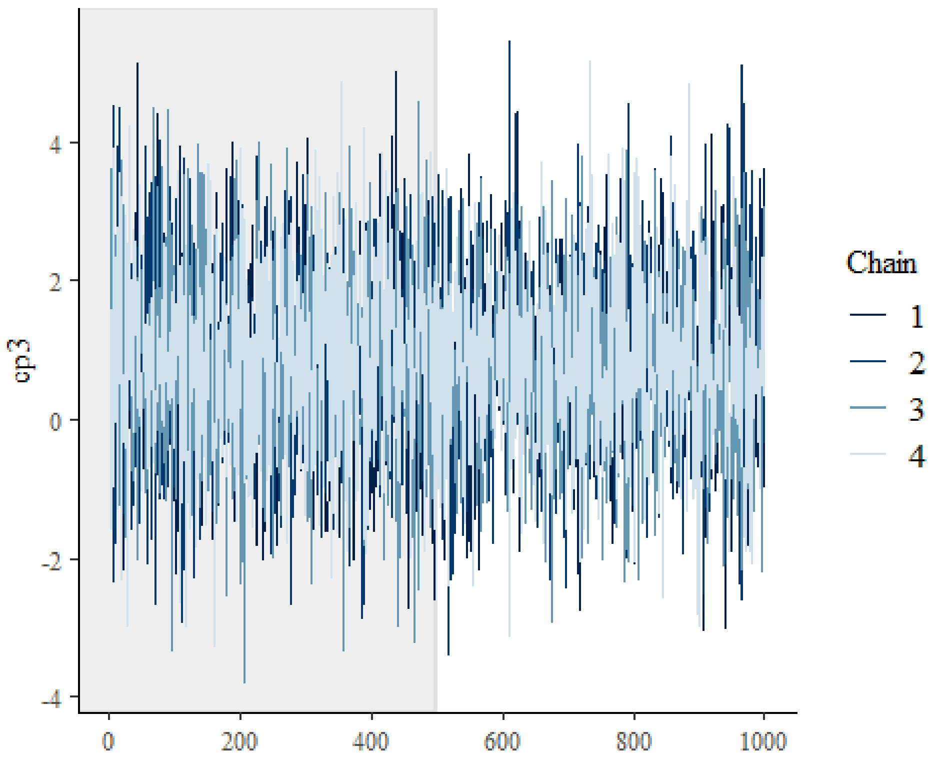
```
mcmc_trace(as.array(post1),  pars = c('cp2'), n_warmup = 500)
```



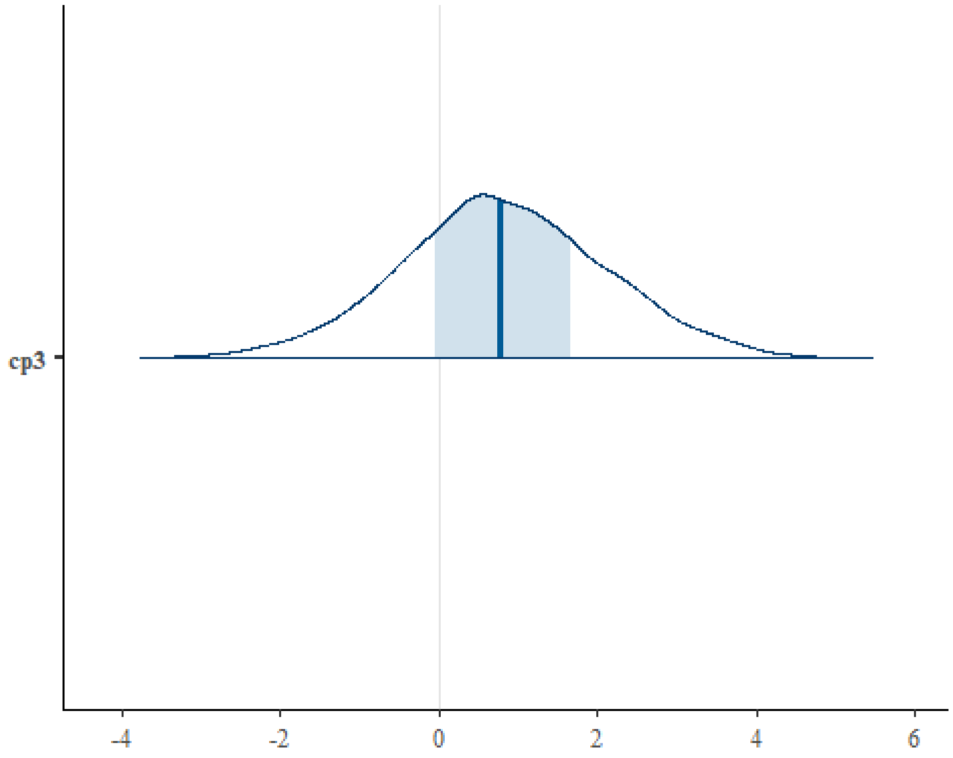
```r
mcmc_areas(as.array(post1), pars = c('cp2'))
```



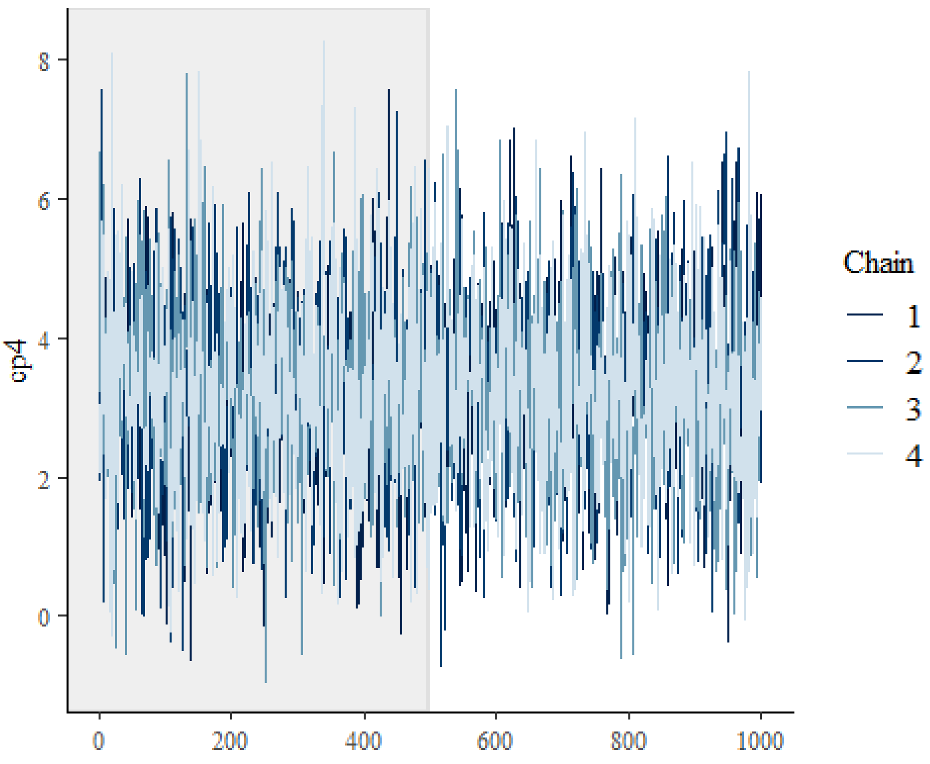
```r
mcmc_trace(as.array(post1),  pars = c('cp3'), n_warmup = 500)
```



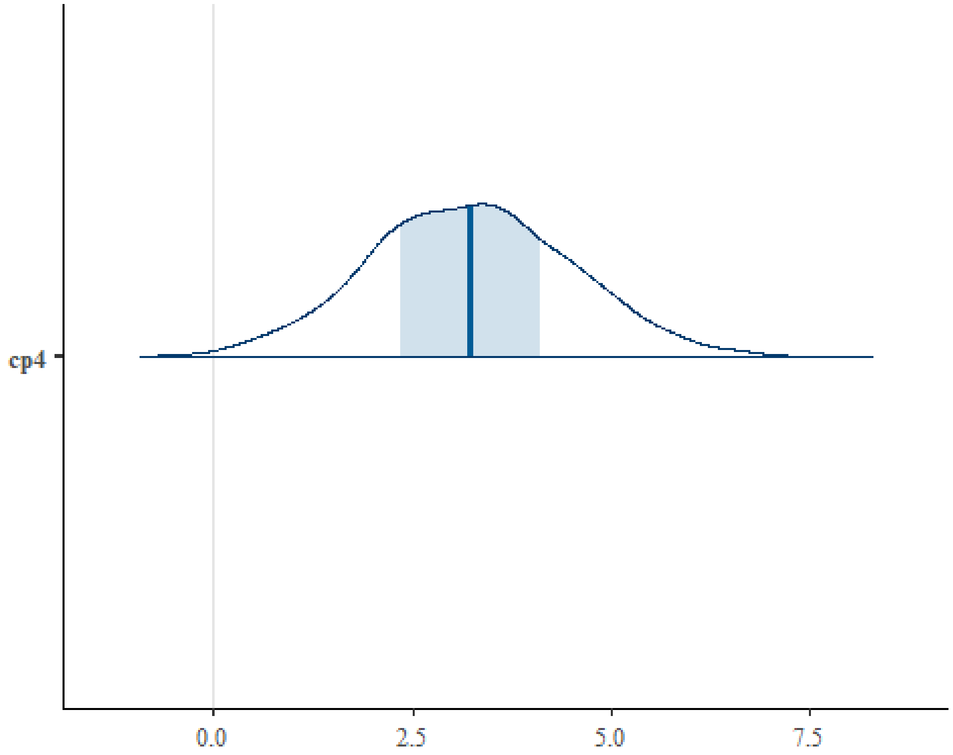
```r
mcmc_areas(as.array(post1), pars = c('cp3'))
```



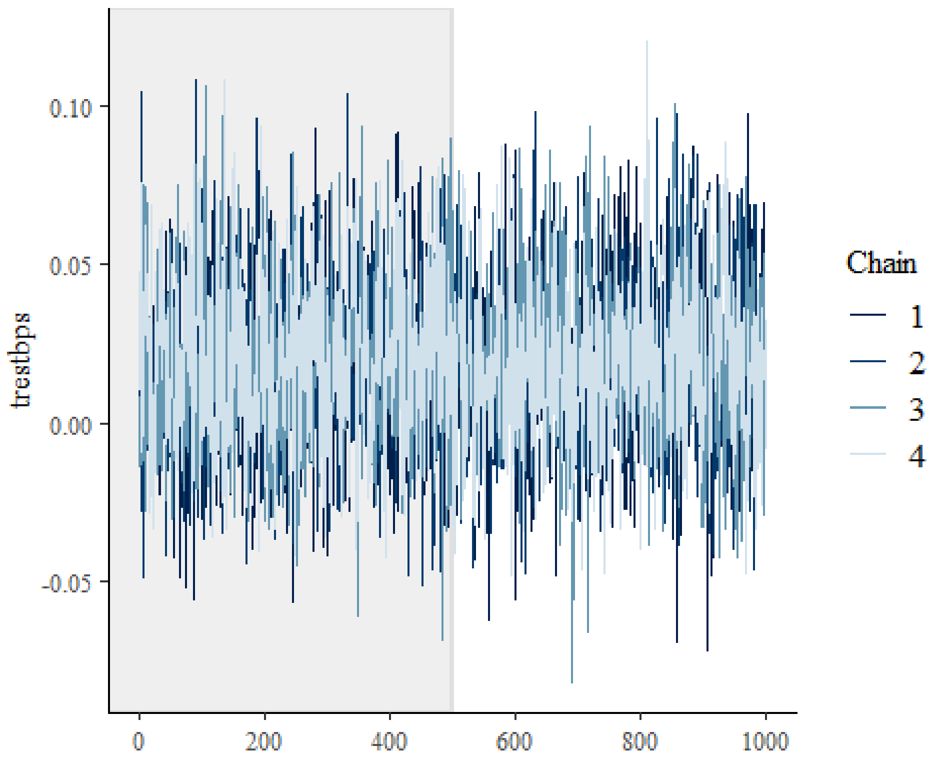
```r
mcmc_trace(as.array(post1),  pars = c('cp4'), n_warmup = 500)
```



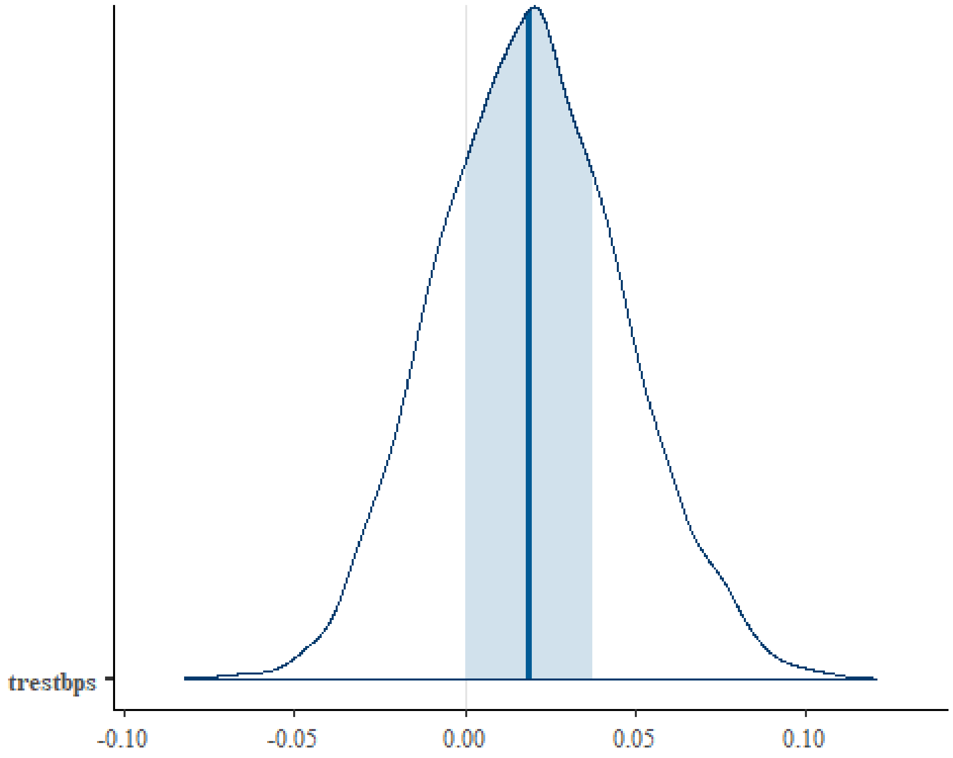
```r
mcmc_areas(as.array(post1), pars = c('cp4'))
```



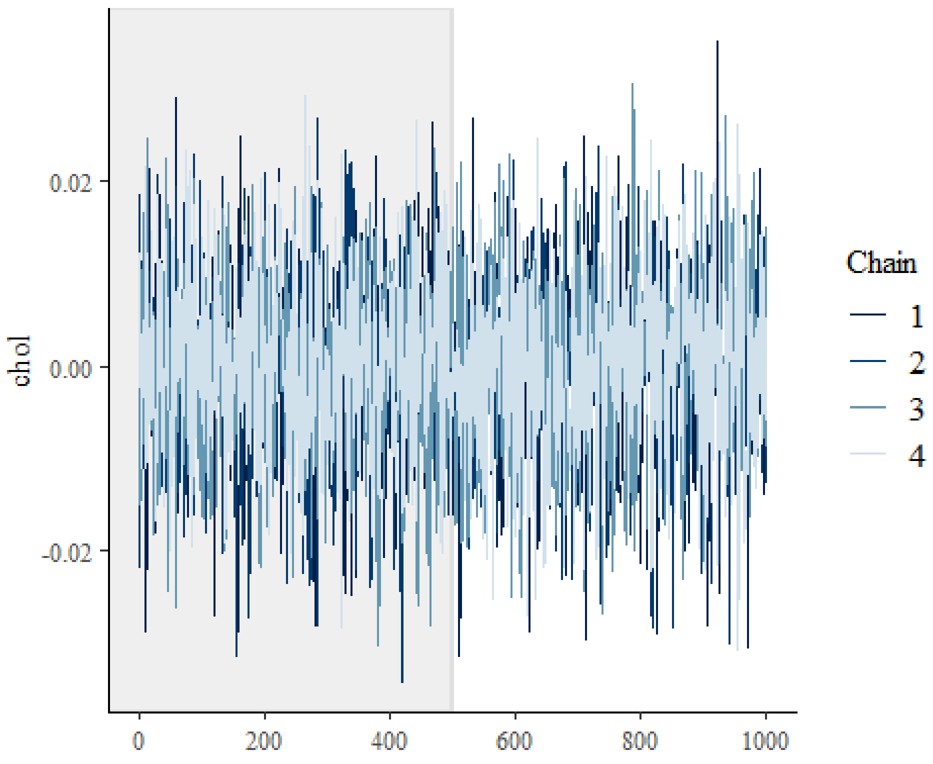
```r
mcmc_trace(as.array(post1),  pars = c('trestbps'), n_warmup = 500)
```



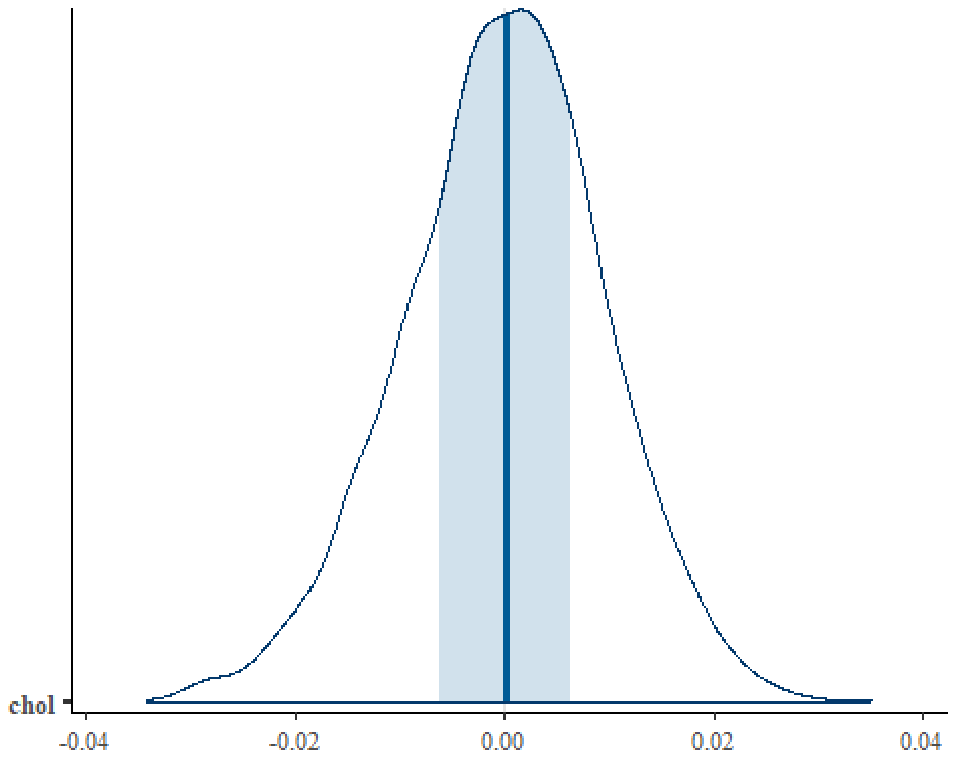
```
mcmc_areas(as.array(post1), pars = c('trestbps'))
```



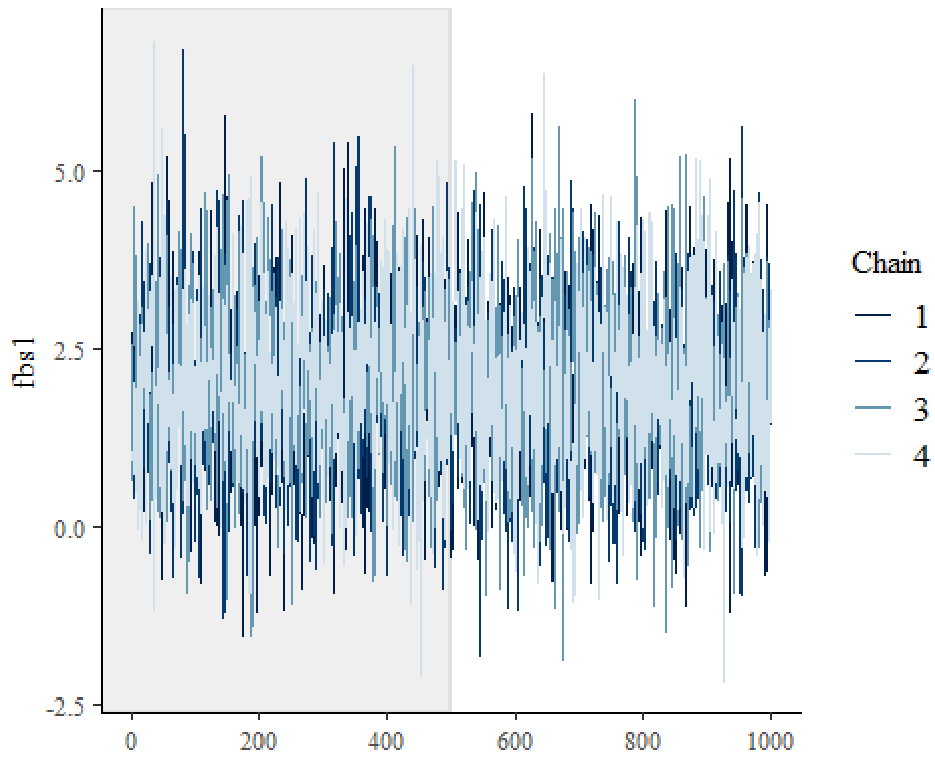
```
mcmc_trace(as.array(post1),  pars = c('chol'), n_warmup = 500)
```



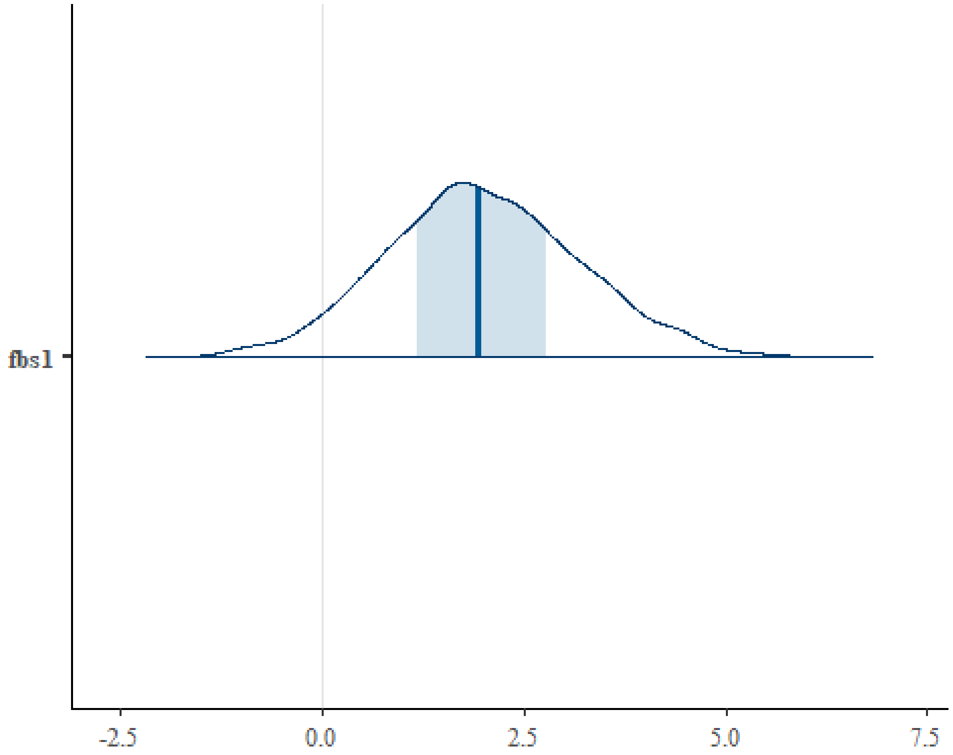
```r
mcmc_areas(as.array(post1), pars = c('chol'))
```



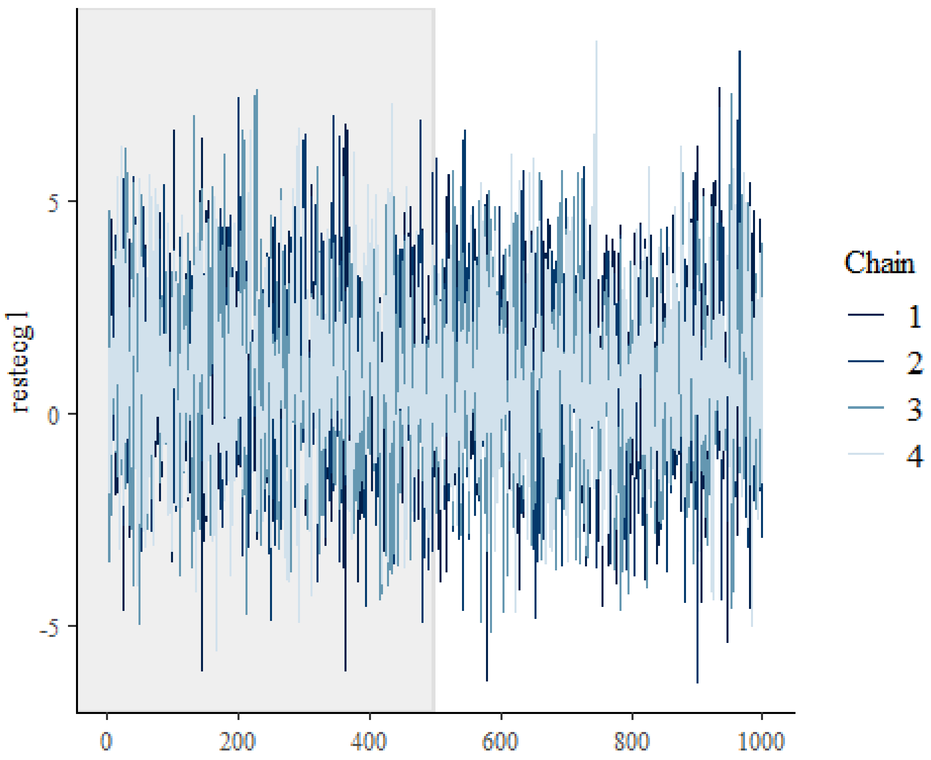
```r
mcmc_trace(as.array(post1),  pars = c('fbs1'), n_warmup = 500)
```



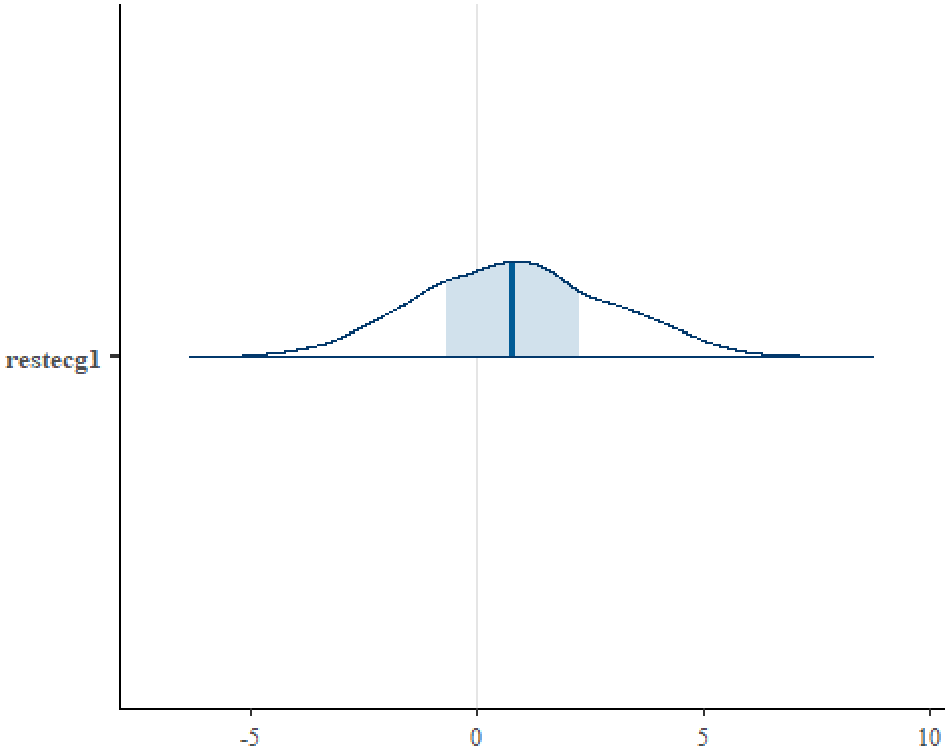
```
mcmc_areas(as.array(post1), pars = c('fbs1'))
```



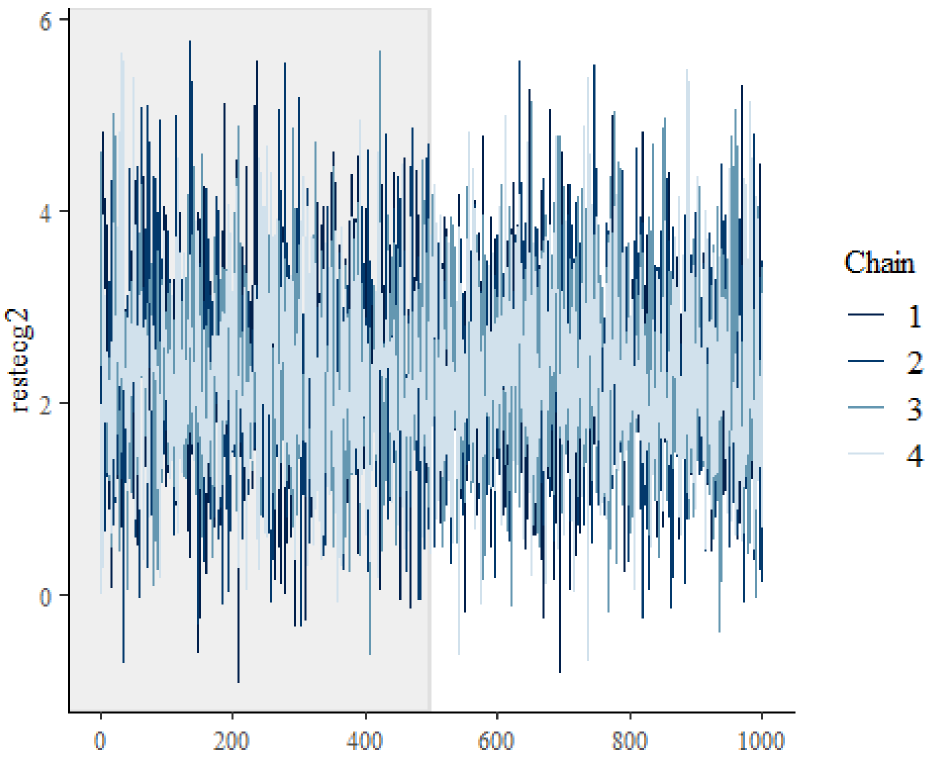
```
mcmc_trace(as.array(post1),  pars = c('restecg1'), n_warmup = 500)
```



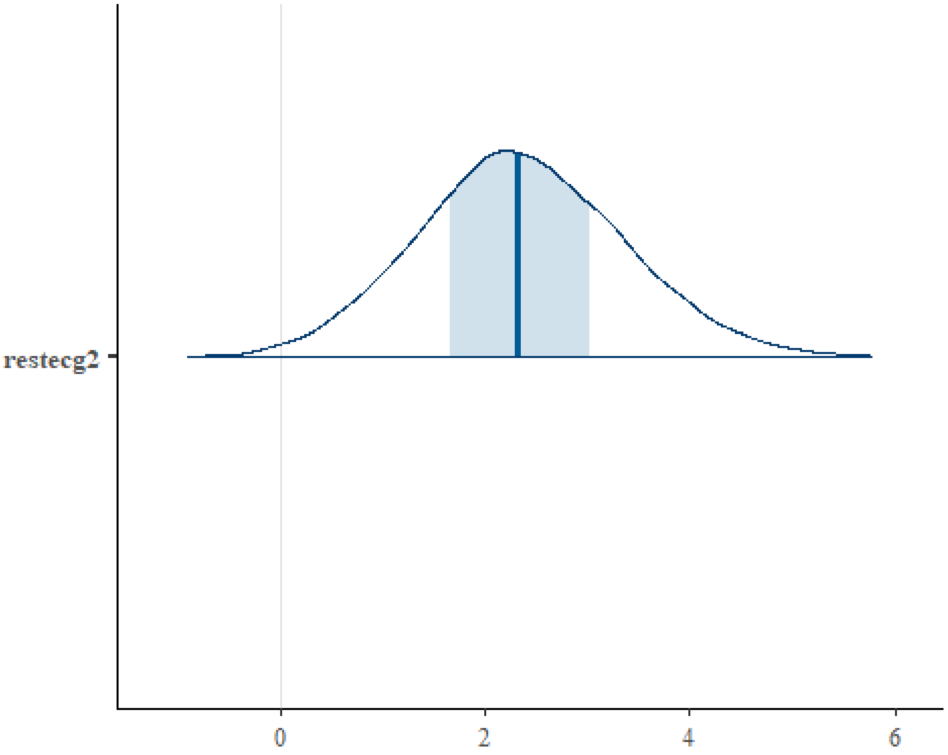
```
mcmc_areas(as.array(post1), pars = c('restecg1'))
```



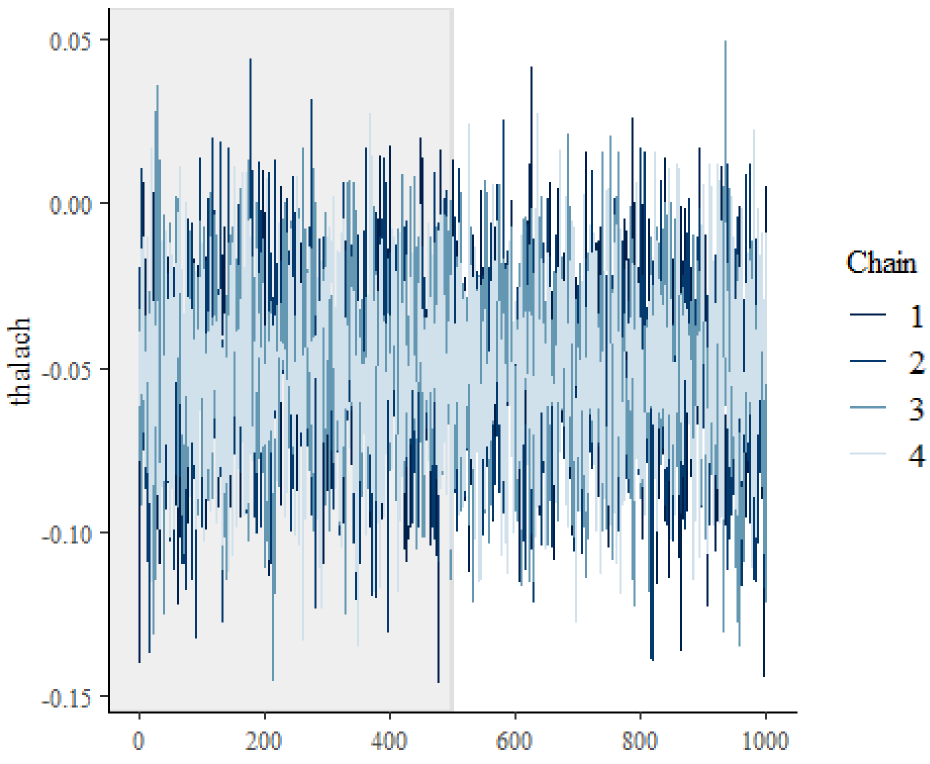
```
mcmc_trace(as.array(post1),  pars = c('restecg2'), n_warmup = 500)
```



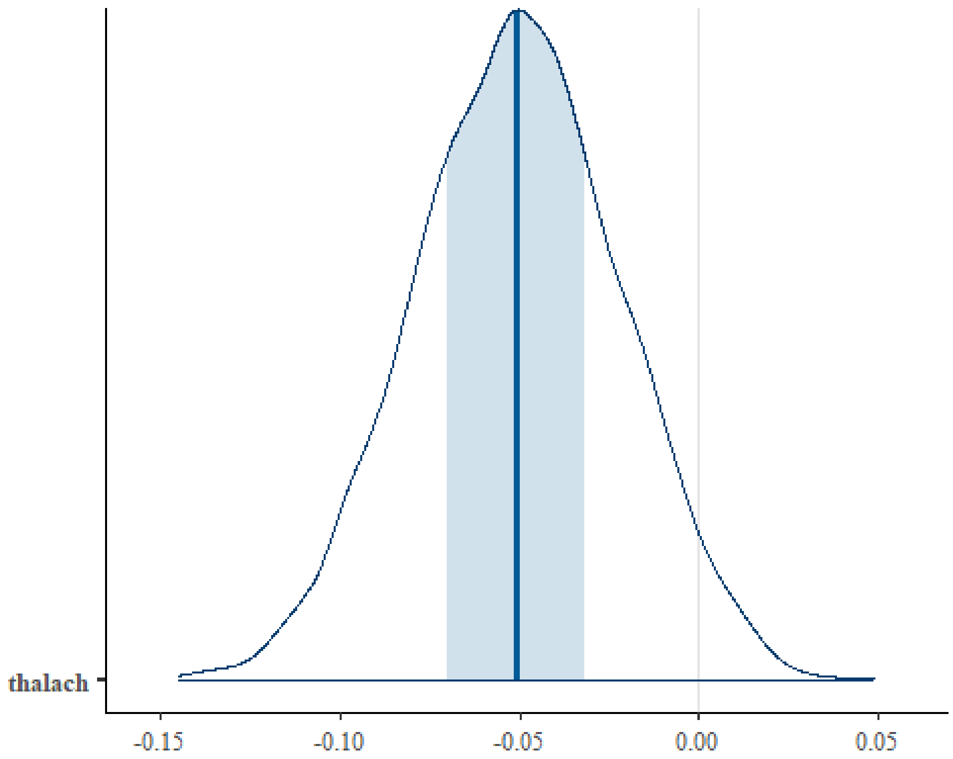
```
mcmc_areas(as.array(post1), pars = c('restecg2'))
```



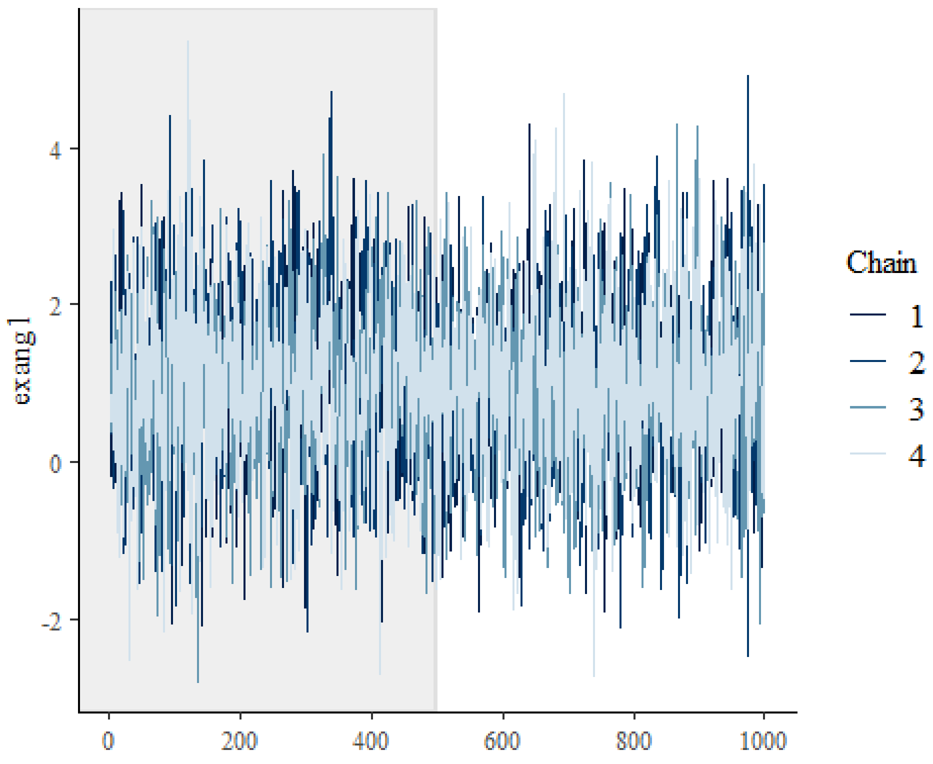
```
mcmc_trace(as.array(post1),  pars = c('thalach'), n_warmup = 500)
```



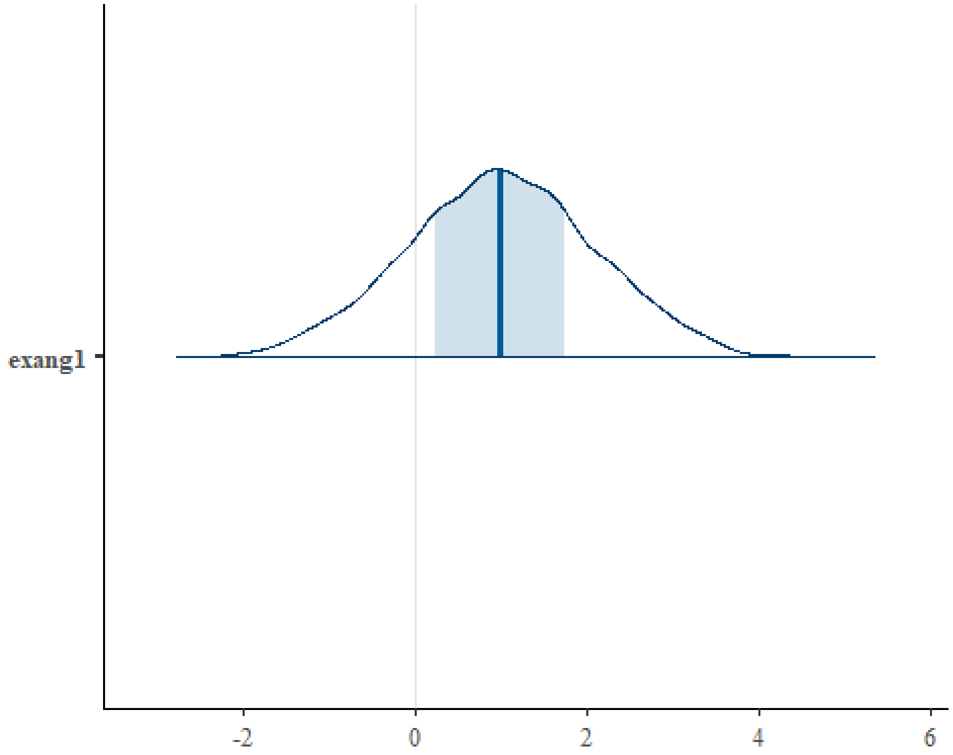
```
mcmc_areas(as.array(post1), pars = c('thalach'))
```



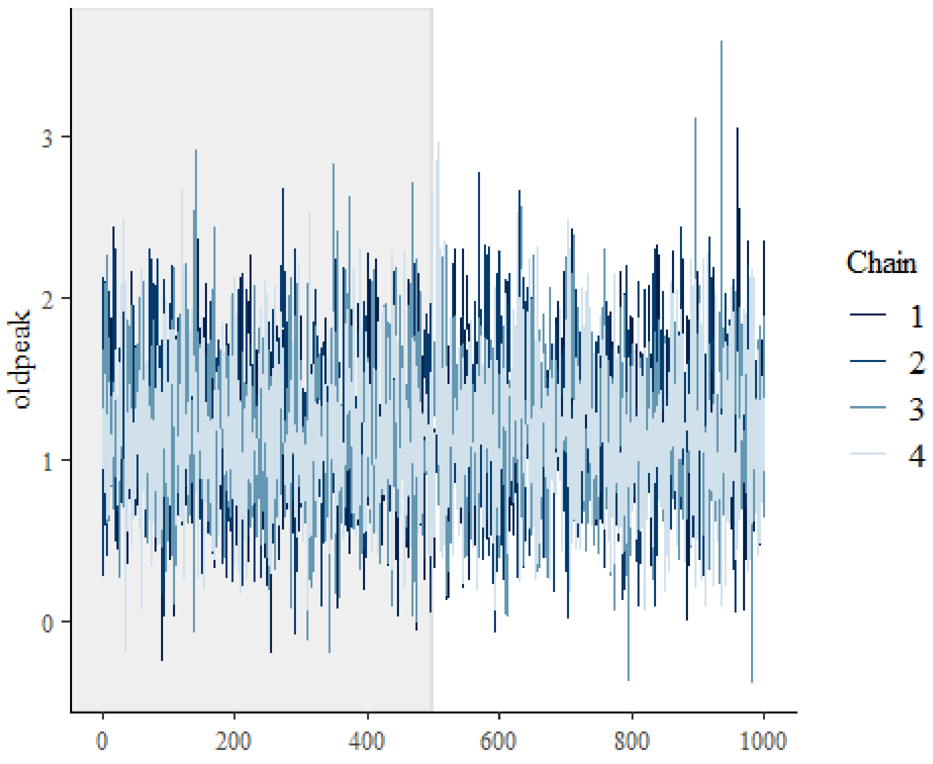
```
mcmc_trace(as.array(post1),  pars = c('exang1'), n_warmup = 500)
```



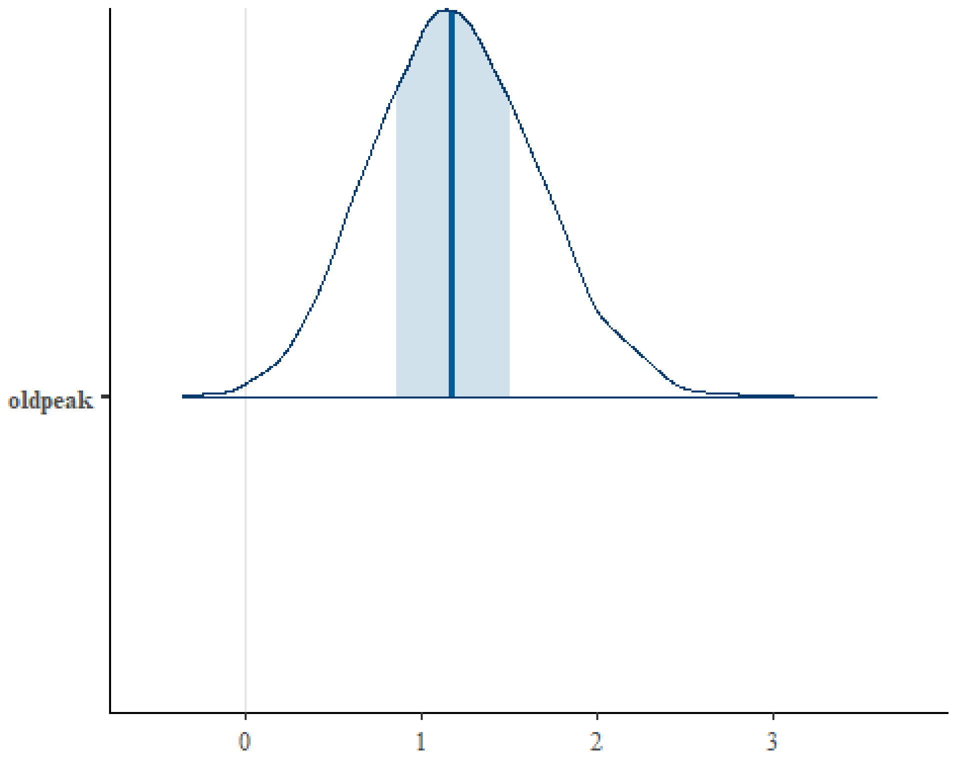
```
mcmc_areas(as.array(post1), pars = c('exang1'))
```



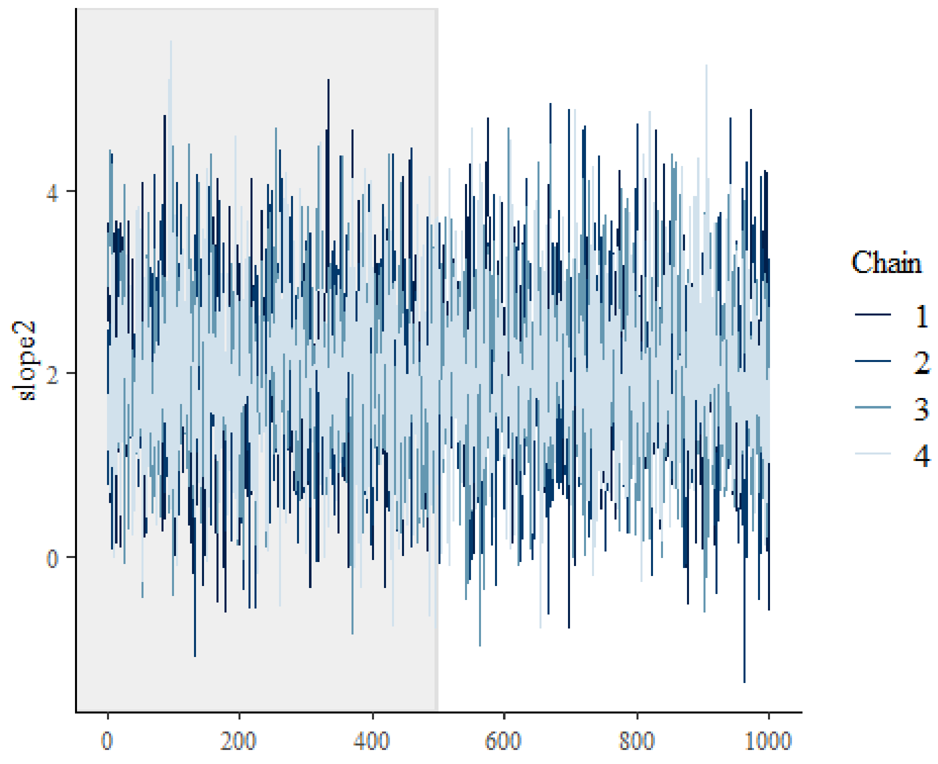
```
mcmc_trace(as.array(post1),  pars = c('oldpeak'), n_warmup = 500)
```



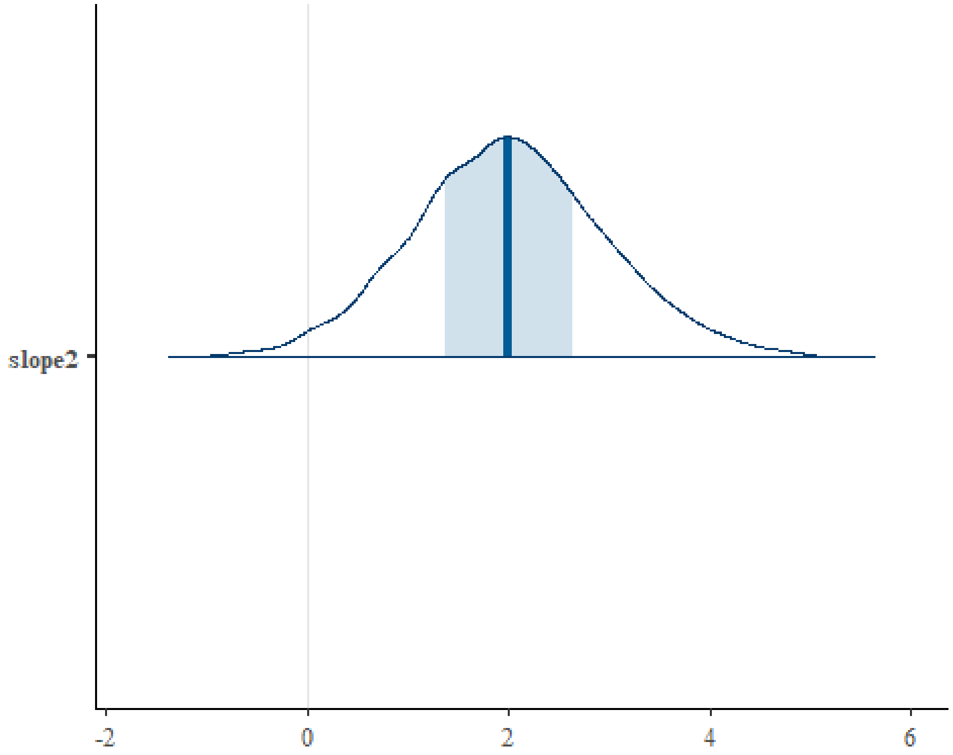
```
mcmc_areas(as.array(post1), pars = c('oldpeak'))
```



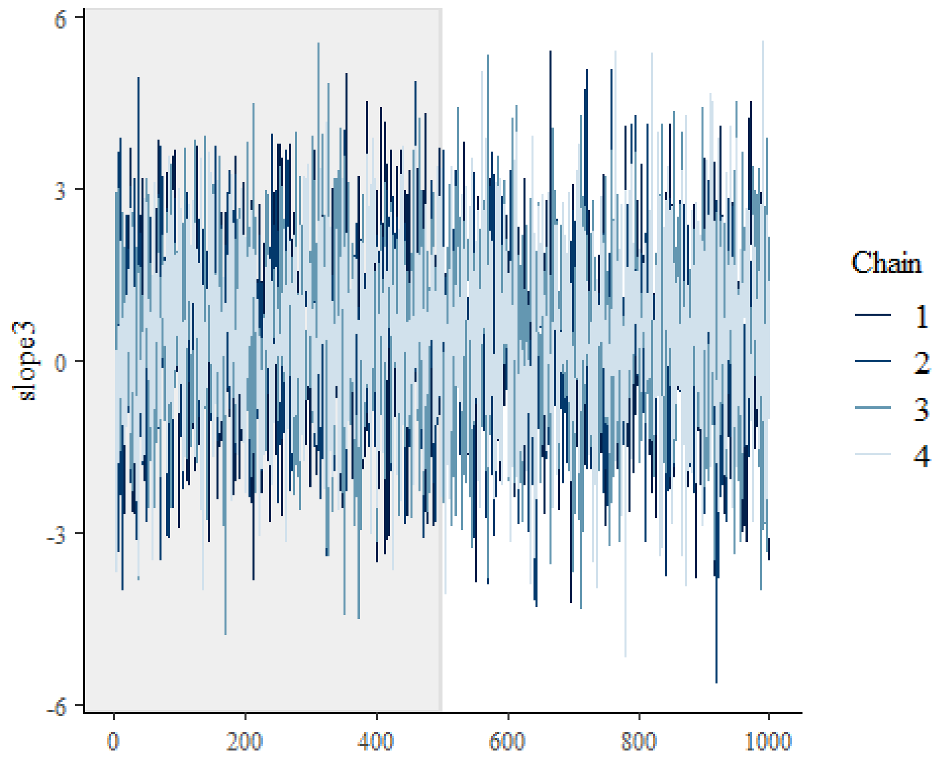
```
mcmc_trace(as.array(post1),  pars = c('slope2'), n_warmup = 500)
```



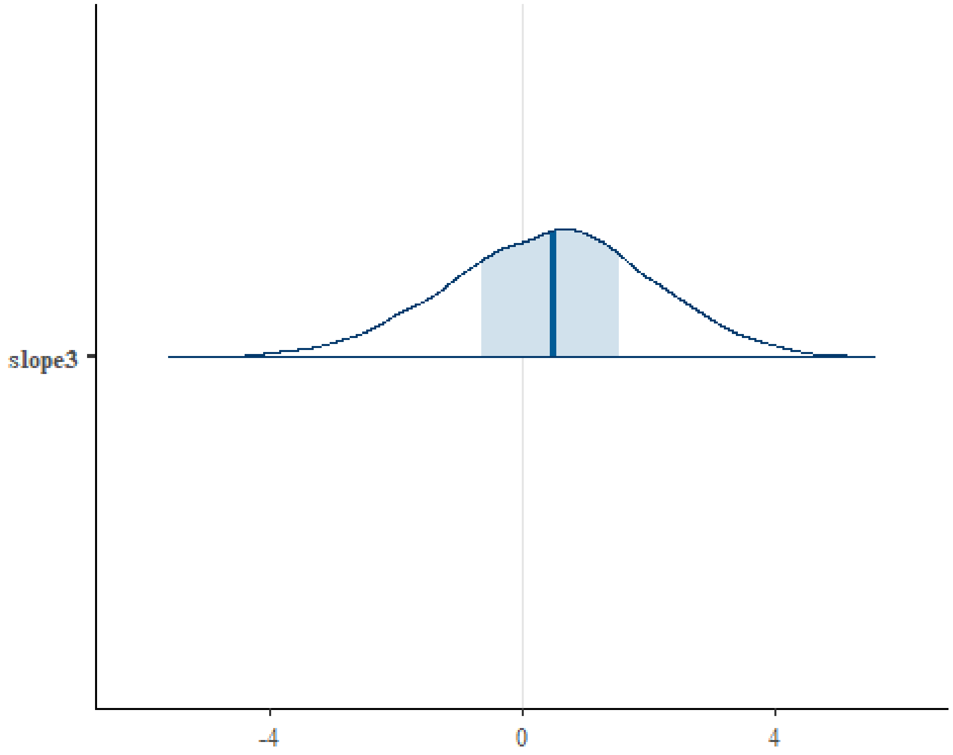
```r
mcmc_areas(as.array(post1), pars = c('slope2'))
```



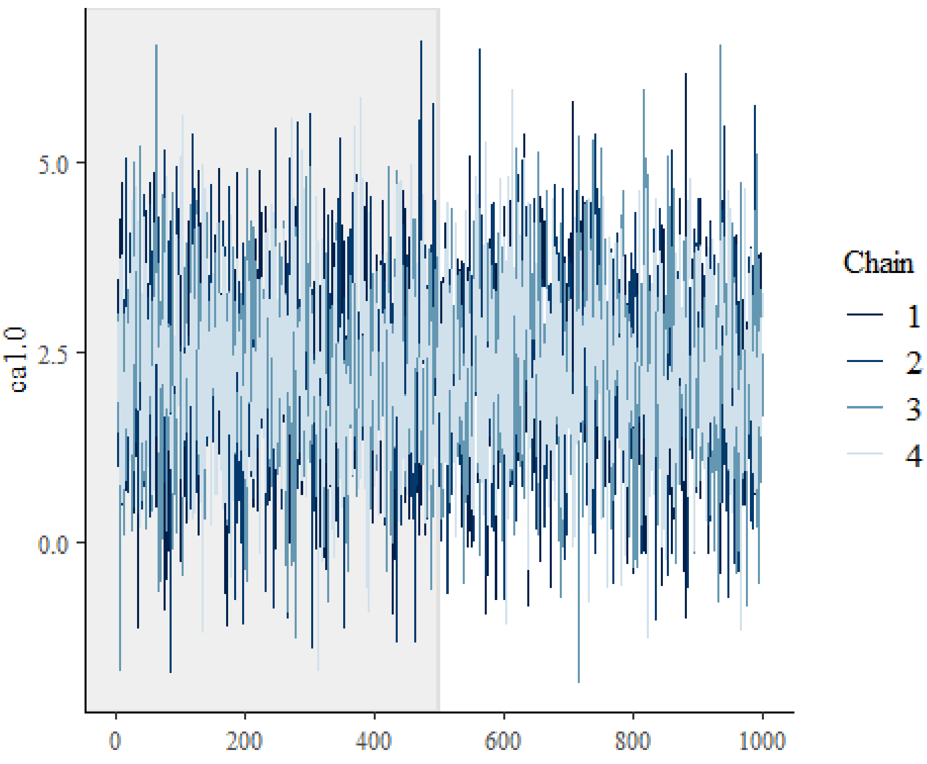
```r
mcmc_trace(as.array(post1),  pars = c('slope3'), n_warmup = 500)
```



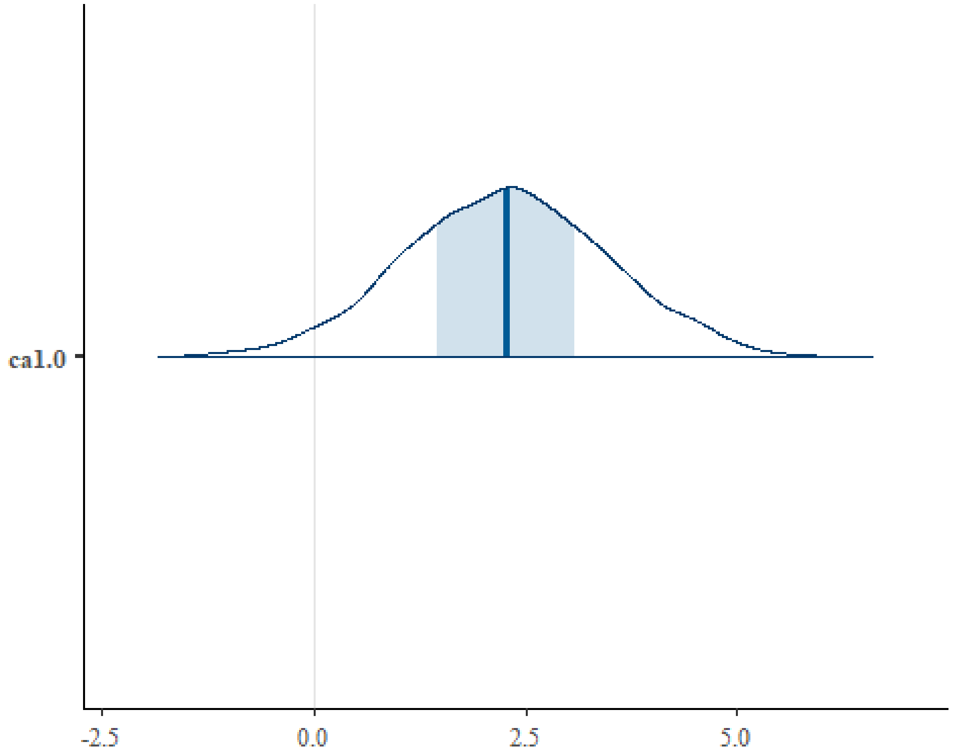
```
mcmc_areas(as.array(post1), pars = c('slope3'))
```



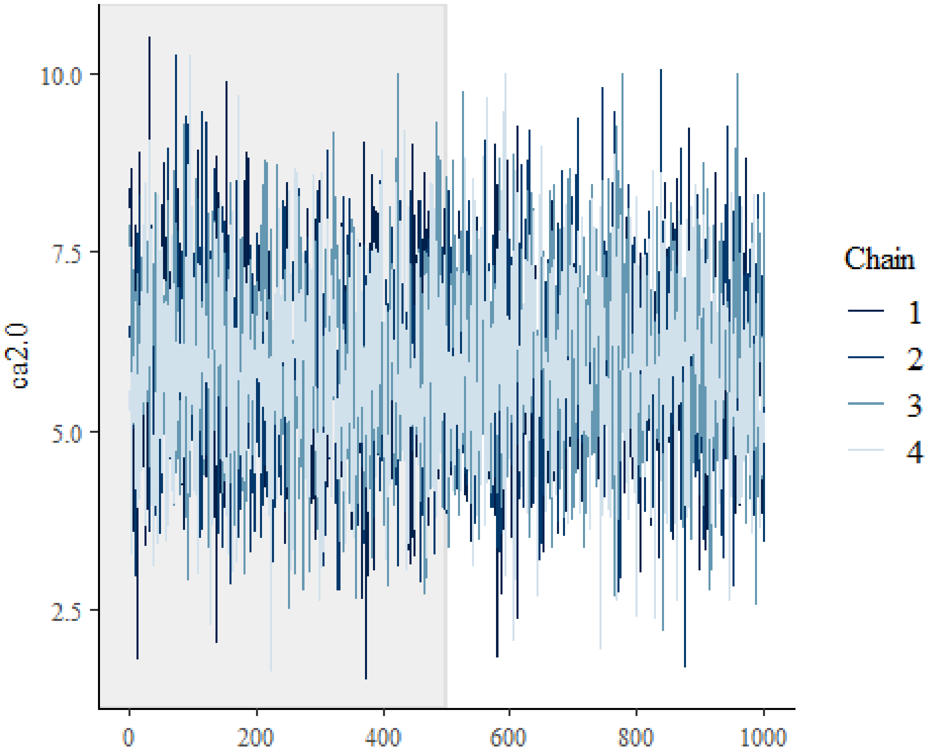
```
mcmc_trace(as.array(post1),  pars = c('ca1.0'), n_warmup = 500)
```



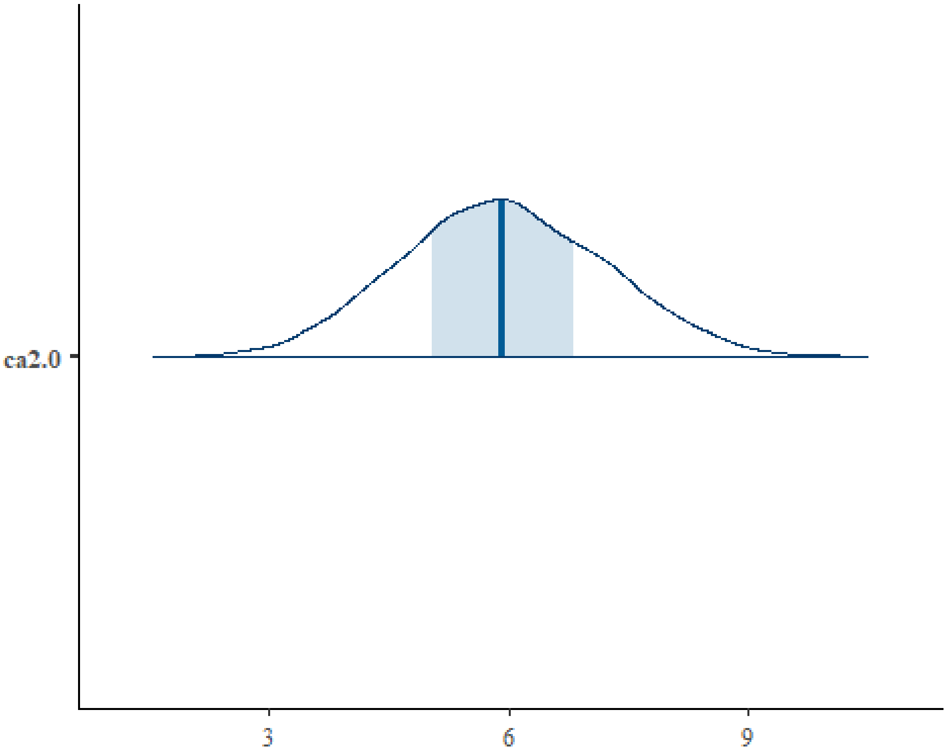
```
mcmc_areas(as.array(post1), pars = c('ca1.0'))
```



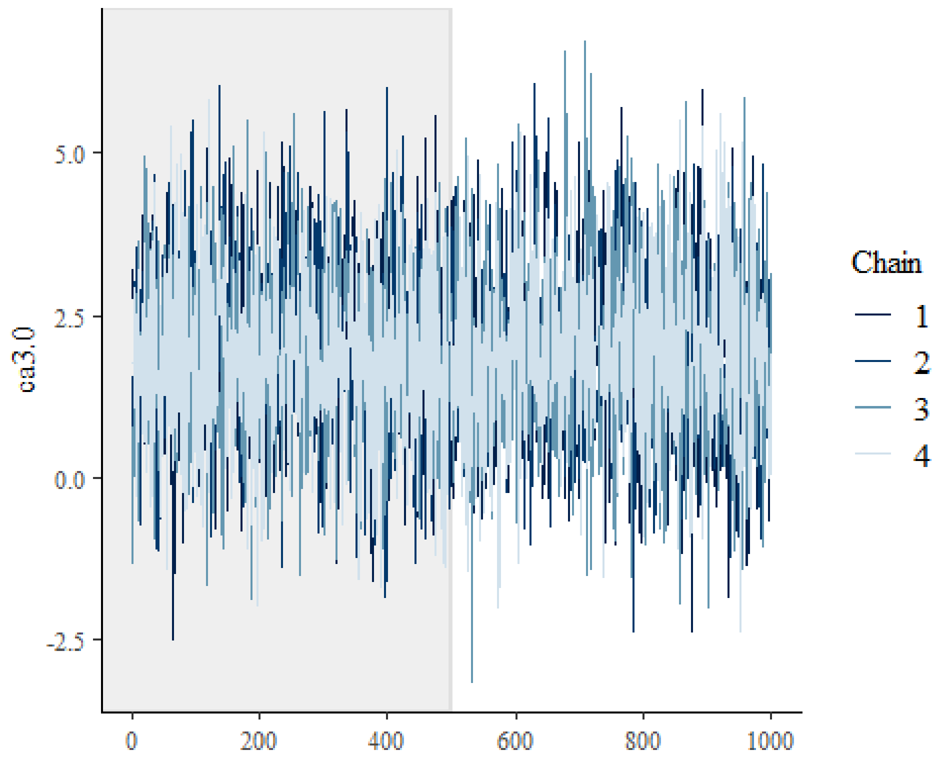
```
mcmc_trace(as.array(post1),  pars = c('ca2.0'), n_warmup = 500)
```



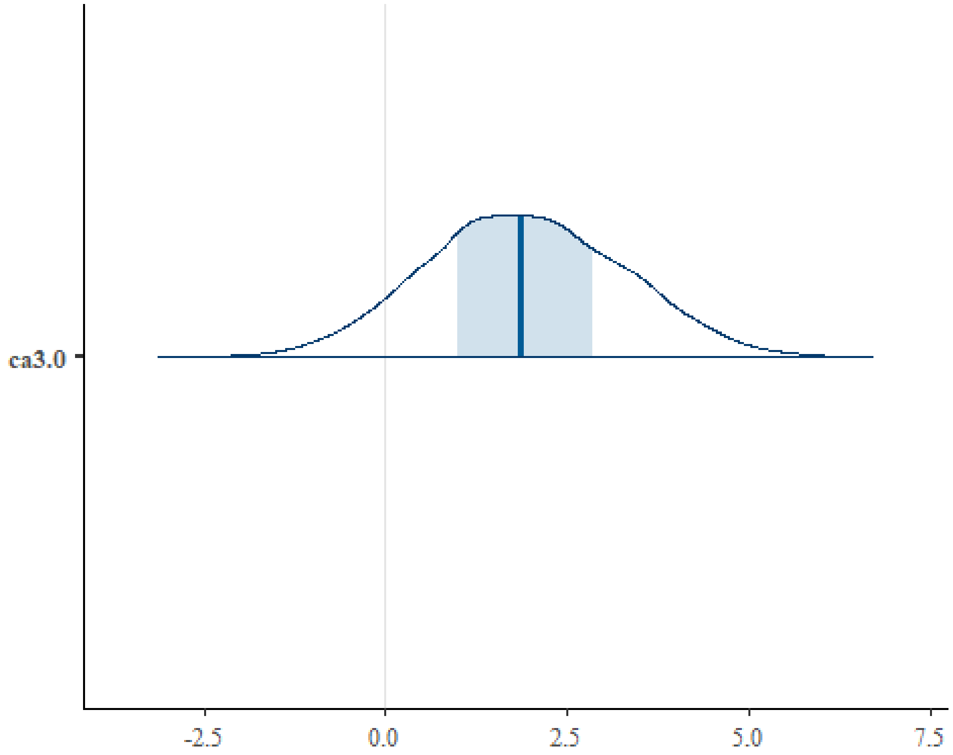
```
mcmc_areas(as.array(post1), pars = c('ca2.0'))
```



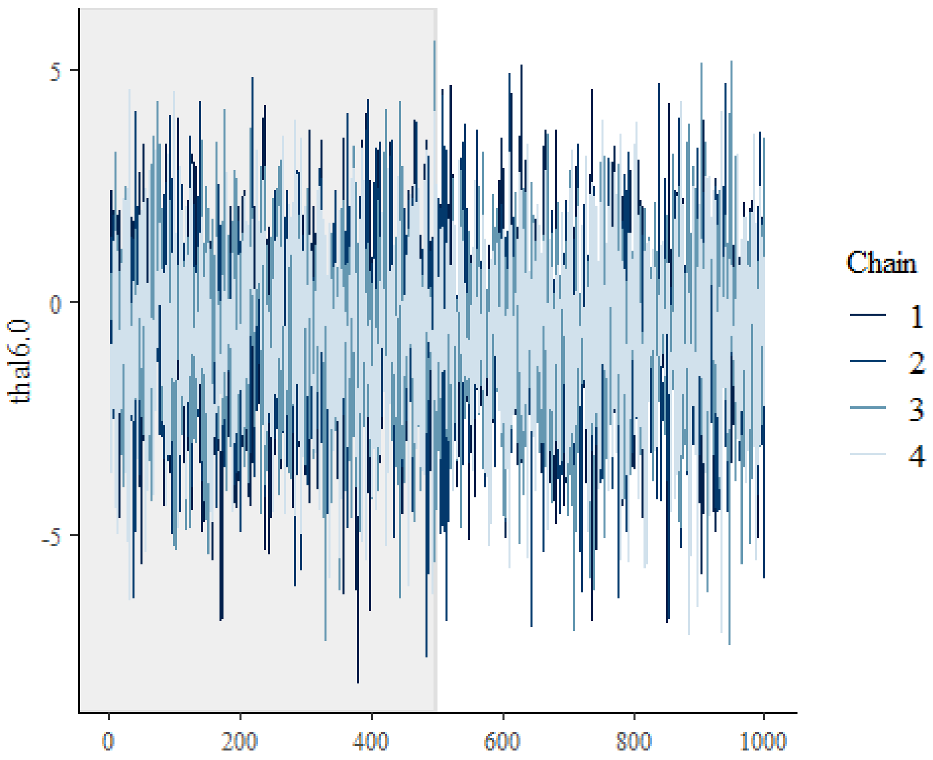
```
mcmc_trace(as.array(post1),  pars = c('ca3.0'), n_warmup = 500)
```



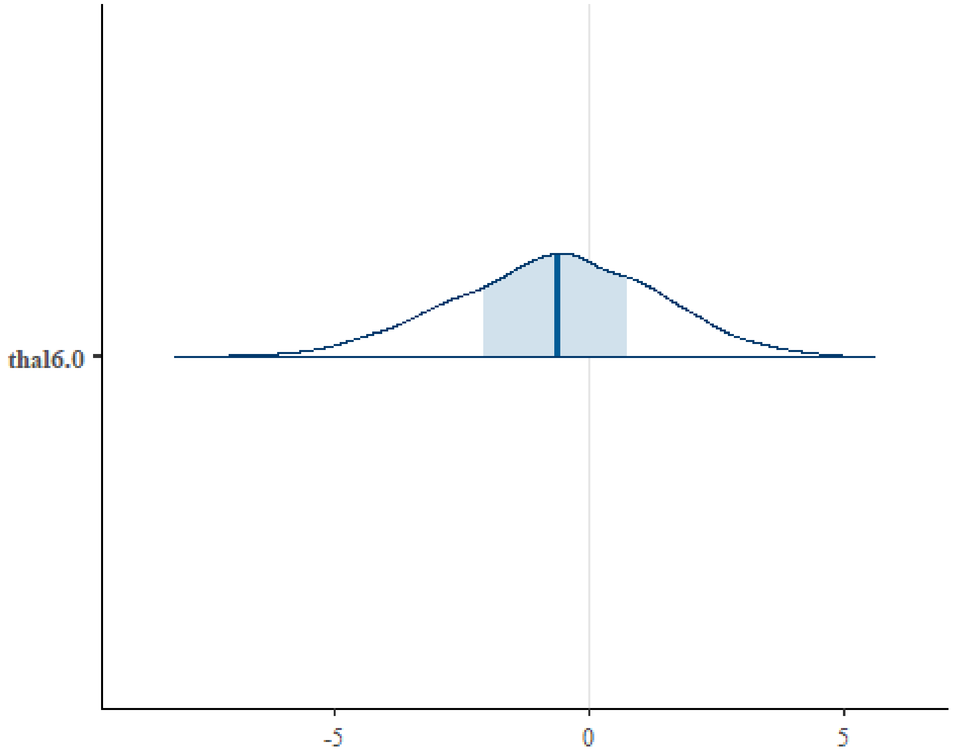
```
mcmc_areas(as.array(post1), pars = c('ca3.0'))
```



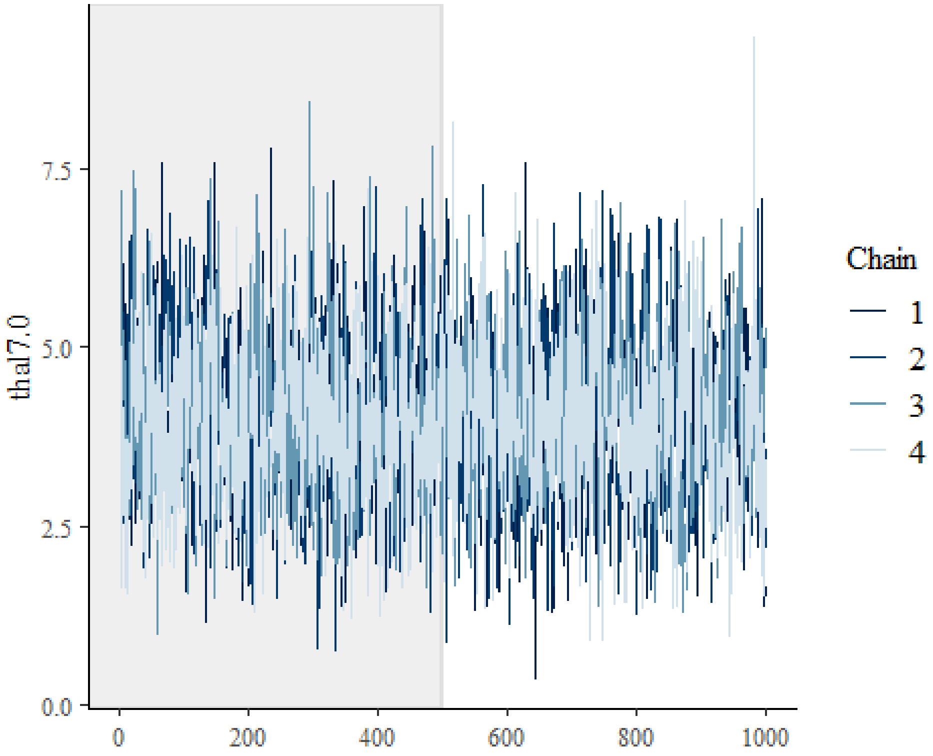
```
mcmc_trace(as.array(post1),  pars = c('thal6.0'), n_warmup = 500)
```



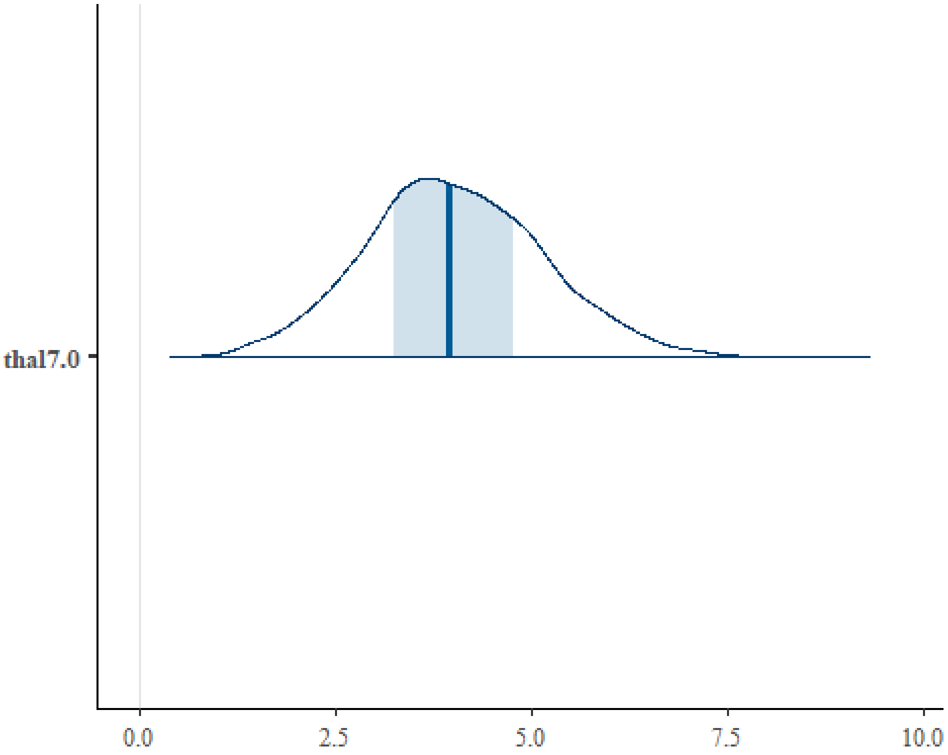
```
mcmc_areas(as.array(post1), pars = c('thal6.0'))
```



```
mcmc_trace(as.array(post1),  pars = c('thal7.0'), n_warmup = 500)
```



```r
mcmc_areas(as.array(post1), pars = c('thal7.0'))
```



Analyze with No Cardiovascular disease and Cardiovascular disease with severity = 4

```
# Taken the class with 0 and 1 only
data7 <- data2[data2['class'] == 0 | data2['class'] == 4,]
post5 <- stan_glm(class ~ ., data = data7,
                  family = binomial(link = "logit"),
                  prior = normal_prior, prior_intercept = normal_prior)

##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.944 seconds (Warm-up)
## Chain 1:                2.23 seconds (Sampling)
## Chain 1:                5.174 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 10 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: 2.913 seconds (Warm-up)
## Chain 2:                2.753 seconds (Sampling)
## Chain 2:                5.666 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.47 seconds (Warm-up)
## Chain 3:                2.128 seconds (Sampling)
## Chain 3:                4.598 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would tak
e 0 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: 2.398 seconds (Warm-up)
## Chain 4:                2.076 seconds (Sampling)
## Chain 4:                4.474 seconds (Total)
## Chain 4:
```

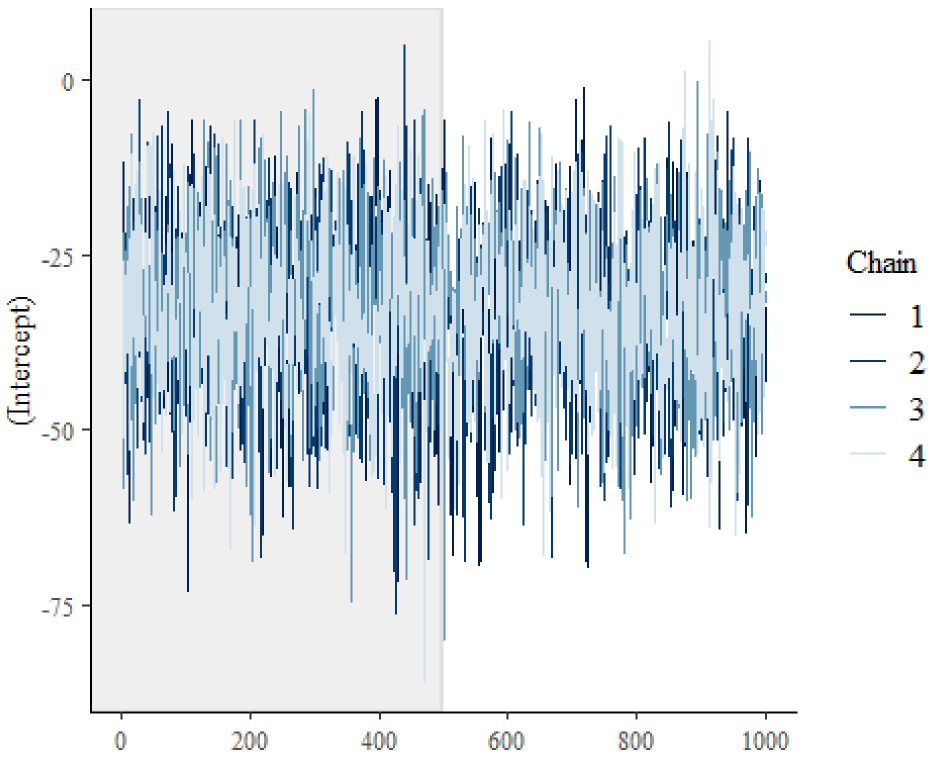
```
round(coef(post5), 2)
```

```
## (Intercept)         age        sex1         cp2         cp3         cp4 
##      -32.35        0.14        0.96       -0.21       -0.72        3.90 
##    trestbps        chol        fbs1    restecg1    restecg2     thalach 
##        0.02       -0.01        0.09        0.93        1.80        0.05 
##      exang1     oldpeak      slope2      slope3       ca1.0       ca2.0 
##        2.06        2.28        2.09       -0.60        2.15        1.70 
##       ca3.0     thal6.0     thal7.0 
##        4.15        0.74        3.12
```

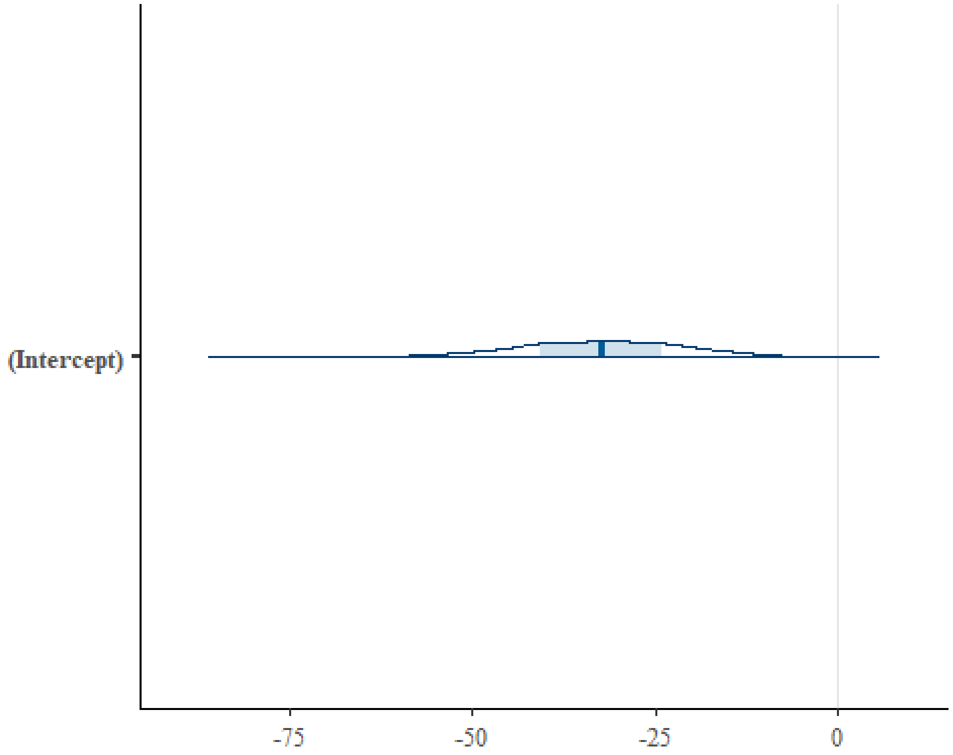
```
round(posterior_interval(post5, prob = 0.95), 2)
```

```
##               2.5%  97.5%
## (Intercept) -57.30 -10.80
## age          -0.02   0.34
## sex1         -2.17   4.33
## cp2          -4.92   4.22
## cp3          -4.17   2.51
## cp4           0.98   6.87
## trestbps     -0.05   0.09
## chol         -0.03   0.02
## fbs1         -3.41   3.46
## restecg1     -3.54   5.27
## restecg2     -0.58   4.26
## thalach      -0.03   0.13
## exang1       -0.40   4.73
## oldpeak       1.09   3.82
## slope2       -0.55   5.11
## slope3       -4.30   3.15
## ca1.0        -0.68   5.16
## ca2.0        -1.73   5.25
## ca3.0         1.40   7.09
## thal6.0      -2.73   4.12
## thal7.0       0.40   5.87
```

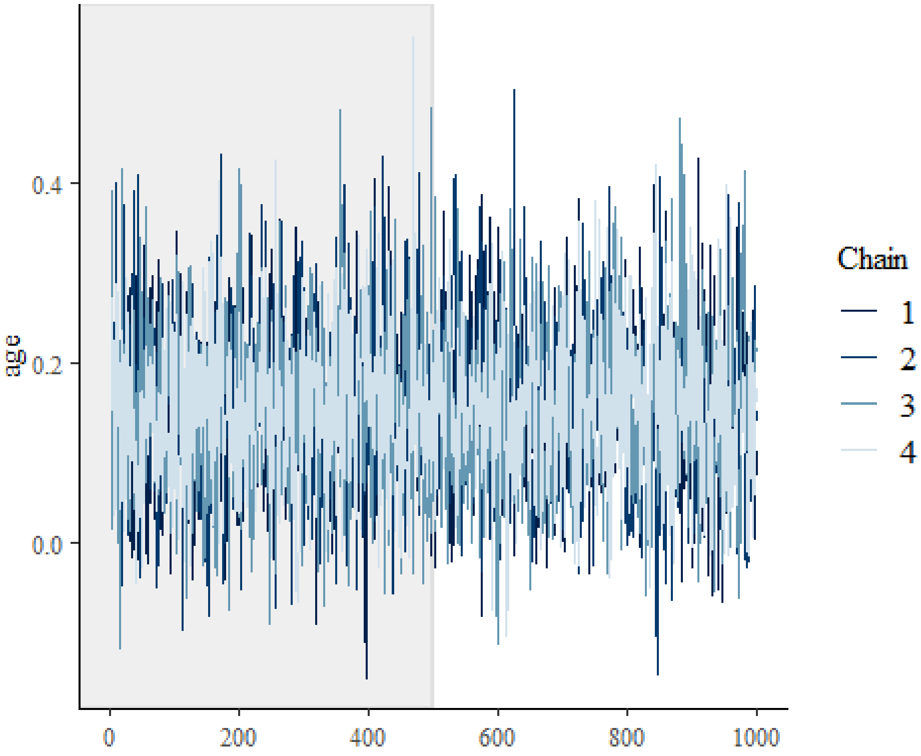
```
post1 <- post5  # to make reuse of the code of plotting

mcmc_trace(as.array(post1),  pars = c('(Intercept)'), n_warmup = 500)
```



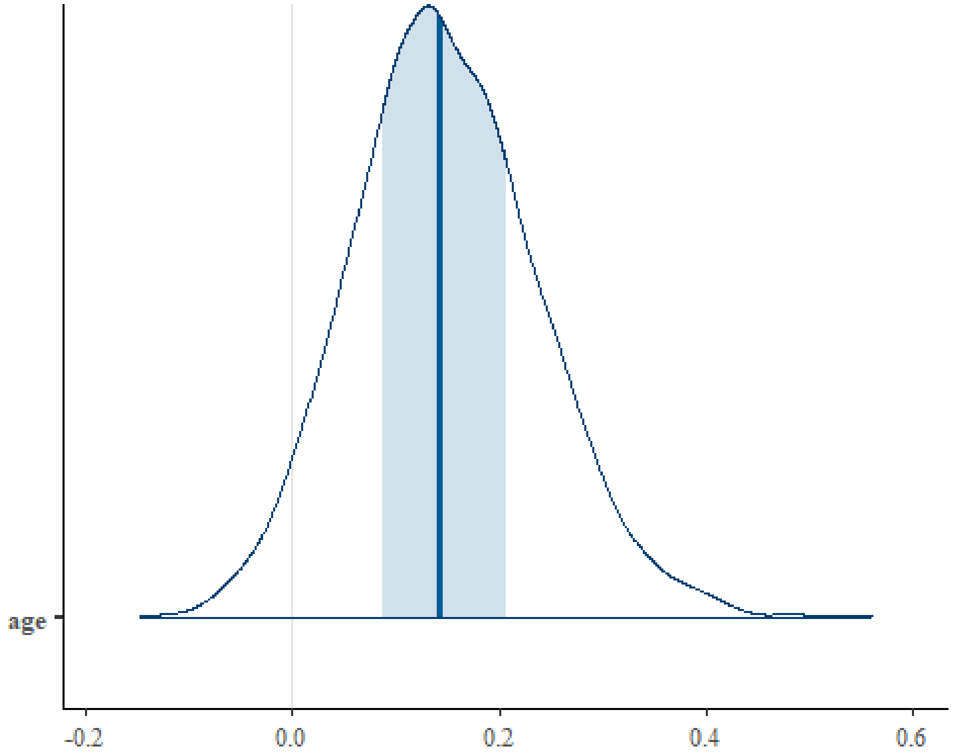
```r
mcmc_areas(as.array(post1), pars = c('(Intercept)'))
```



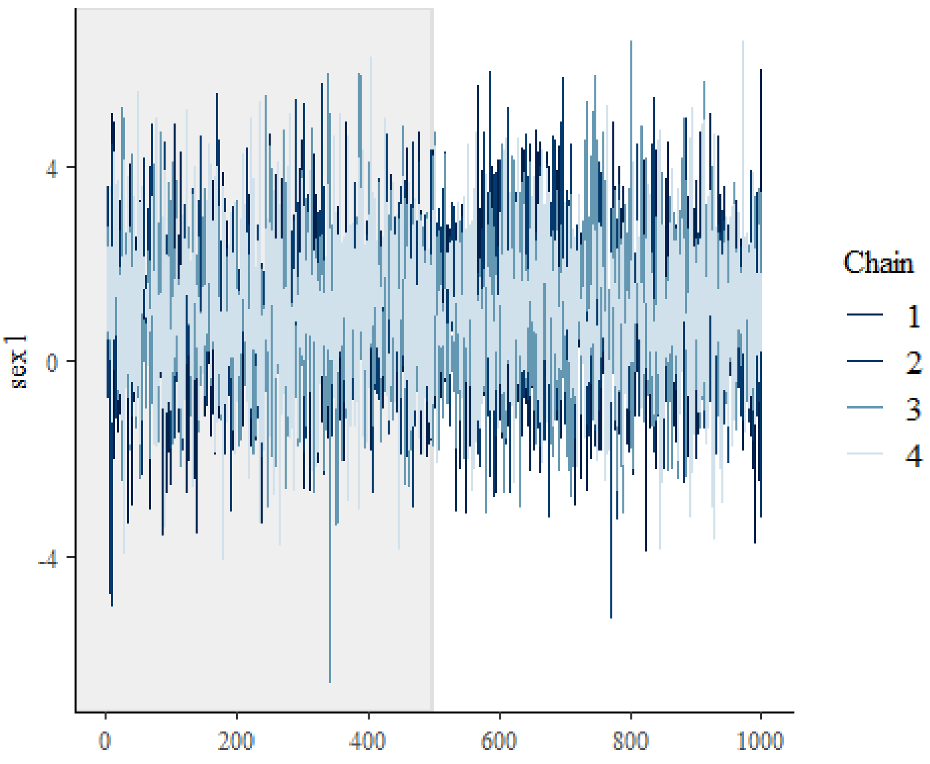
```r
mcmc_trace(as.array(post1),  pars = c('age'), n_warmup = 500)
```



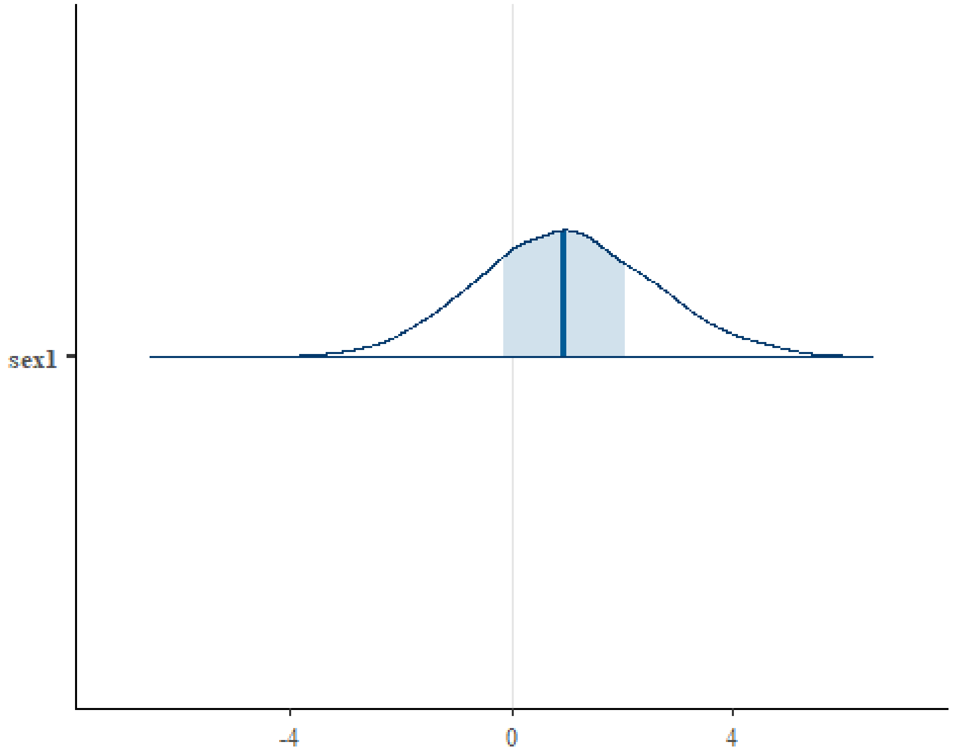
```
mcmc_areas(as.array(post1), pars = c('age'))
```



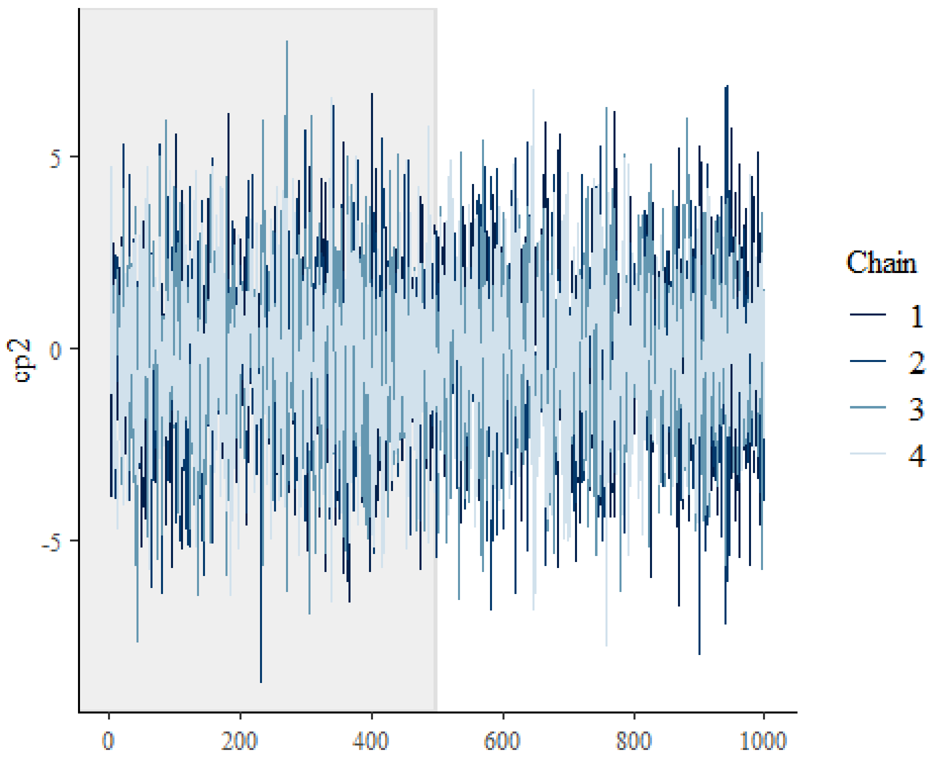
```
mcmc_trace(as.array(post1),  pars = c('sex1'), n_warmup = 500)
```



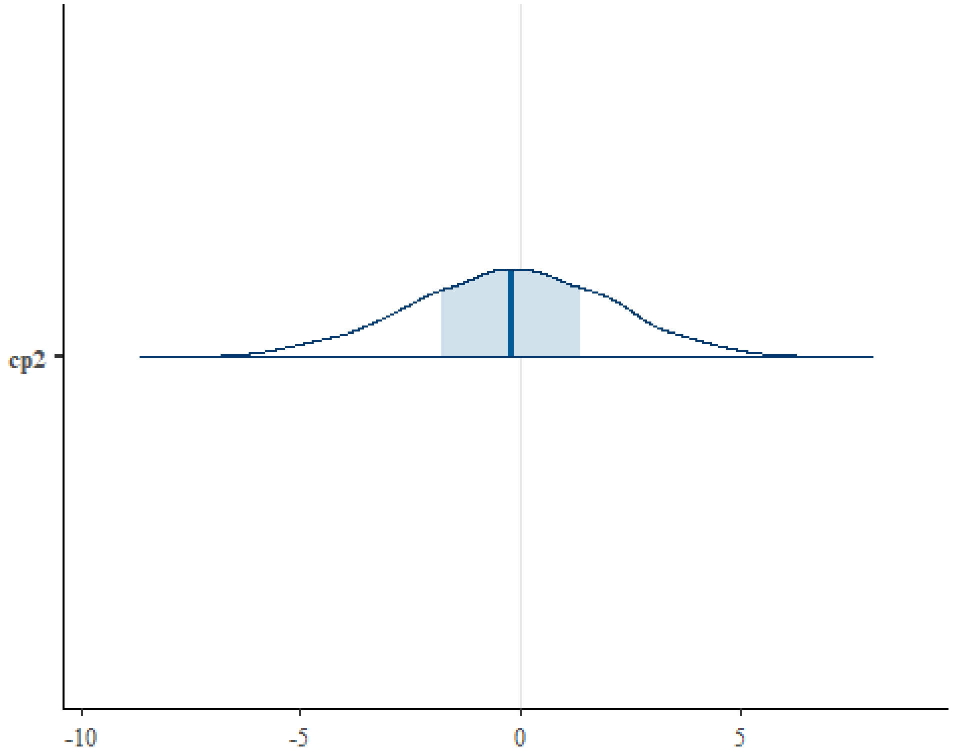
```
mcmc_areas(as.array(post1), pars = c('sex1'))
```



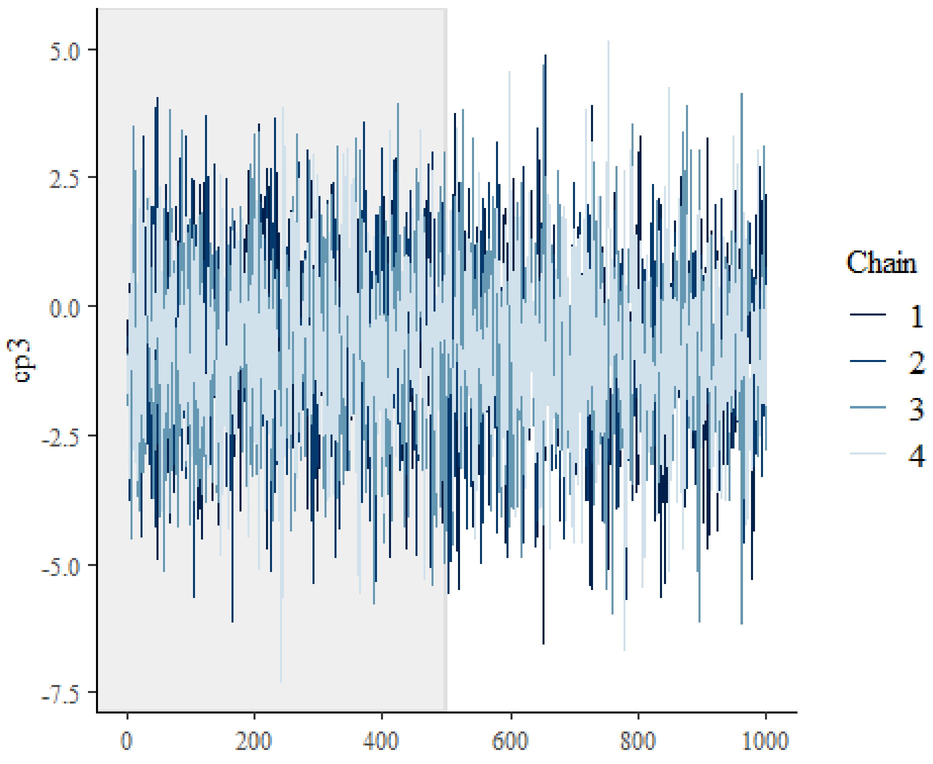
```
mcmc_trace(as.array(post1),  pars = c('cp2'), n_warmup = 500)
```



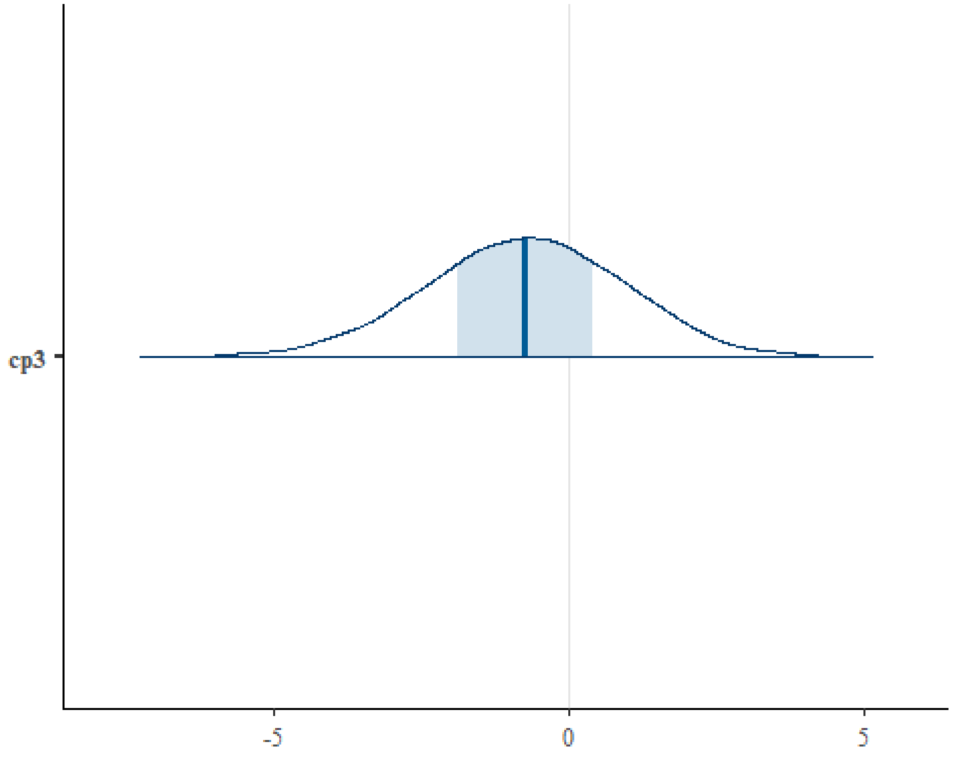
```
mcmc_areas(as.array(post1), pars = c('cp2'))
```



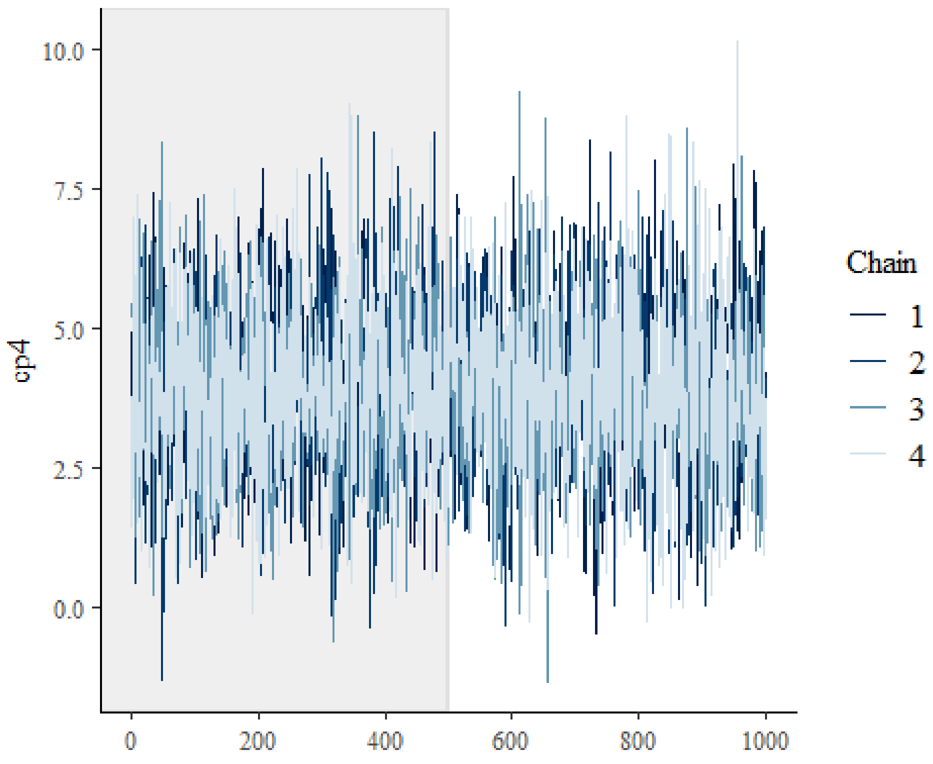
```
mcmc_trace(as.array(post1),  pars = c('cp3'), n_warmup = 500)
```



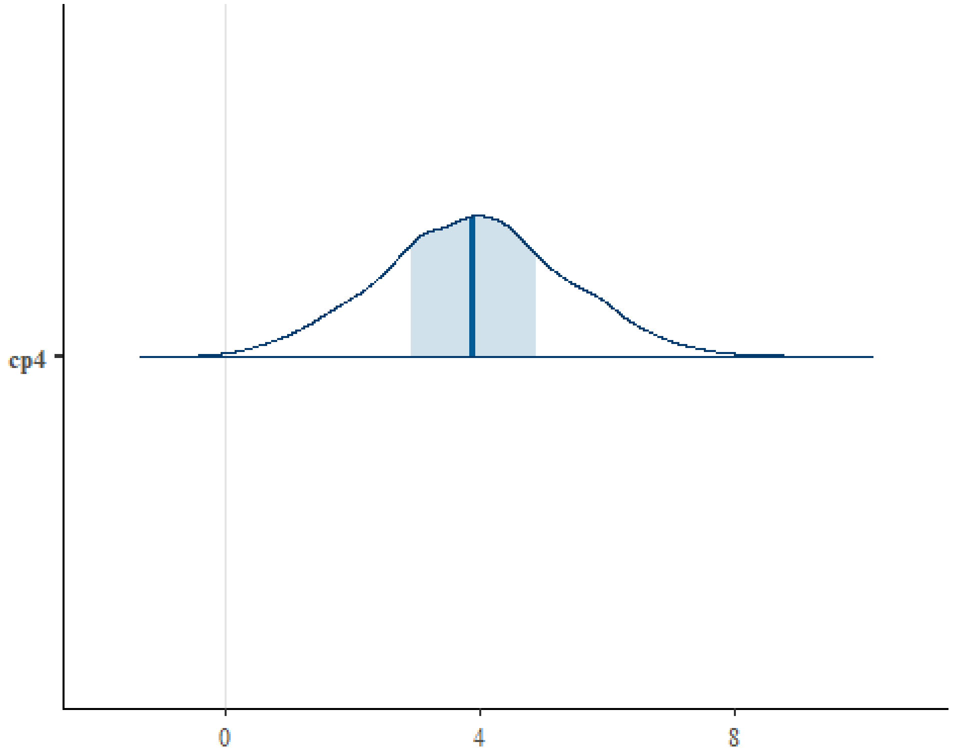
```r
mcmc_areas(as.array(post1), pars = c('cp3'))
```



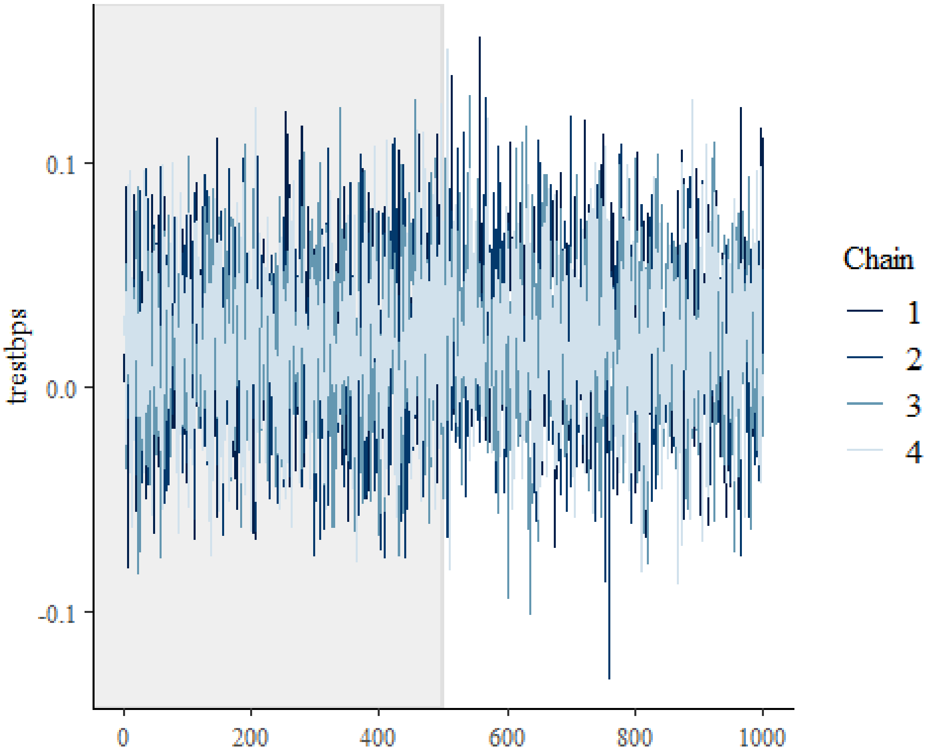
```r
mcmc_trace(as.array(post1),  pars = c('cp4'), n_warmup = 500)
```



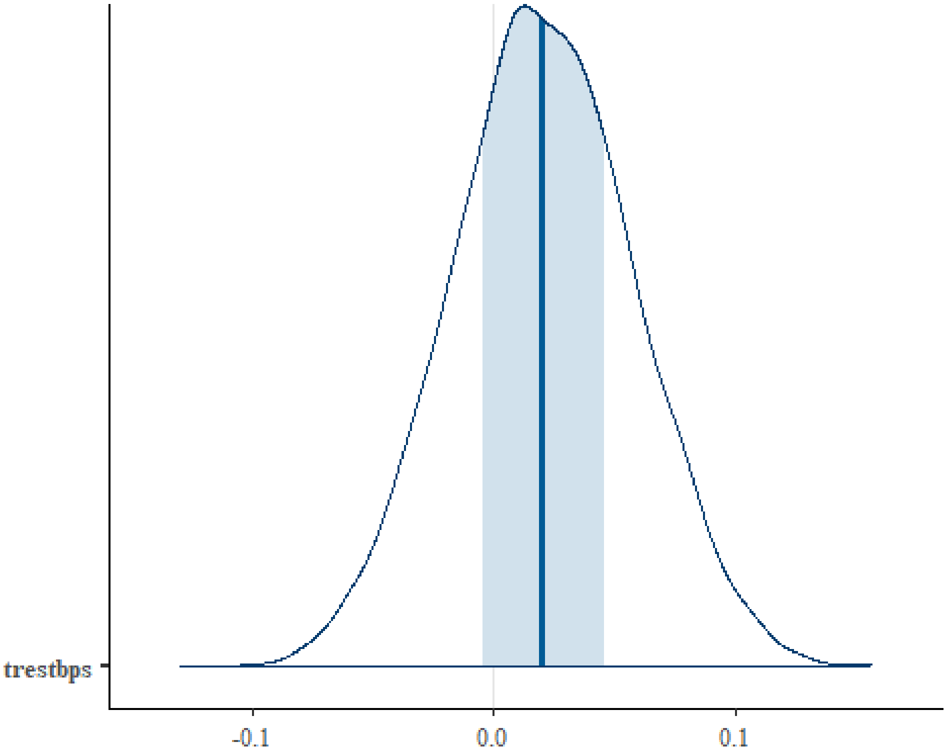
```
mcmc_areas(as.array(post1), pars = c('cp4'))
```



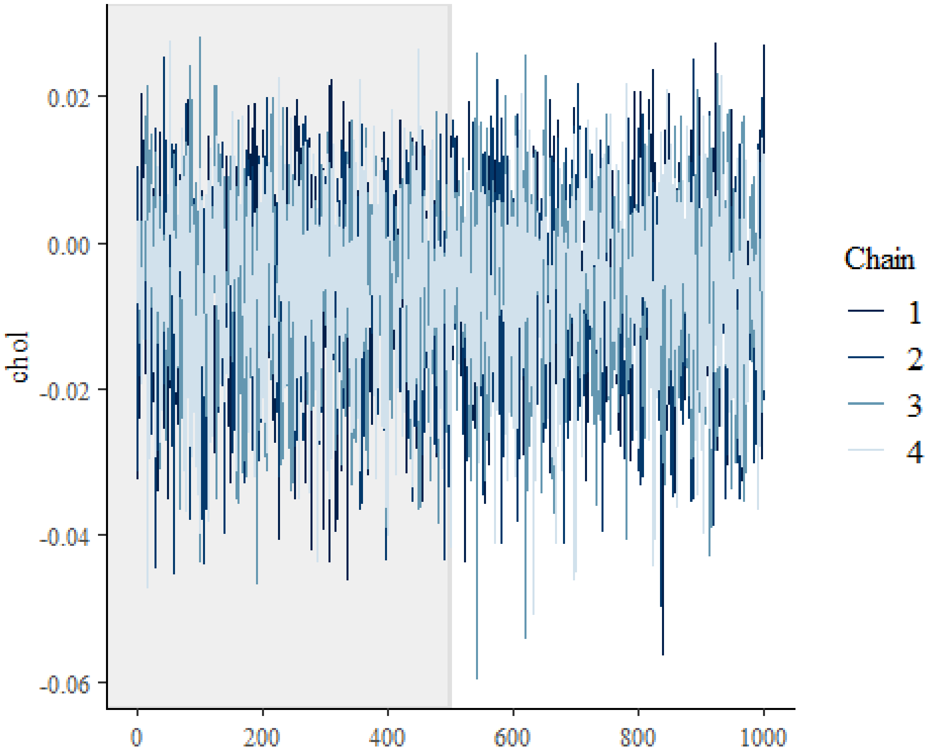
```
mcmc_trace(as.array(post1),  pars = c('trestbps'), n_warmup = 500)
```



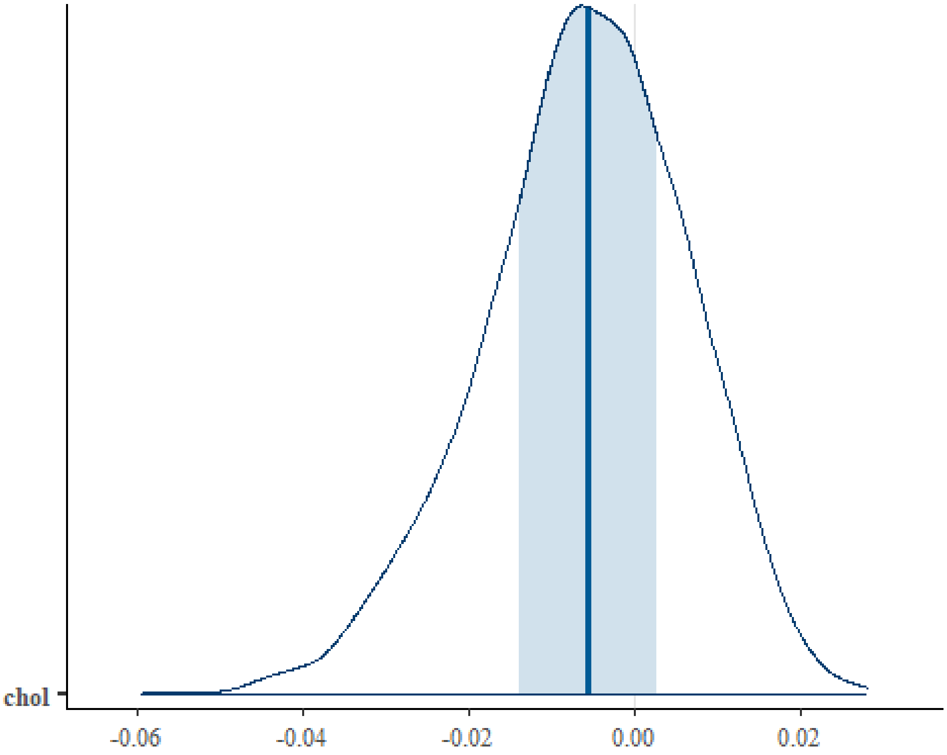
```
mcmc_areas(as.array(post1), pars = c('trestbps'))
```



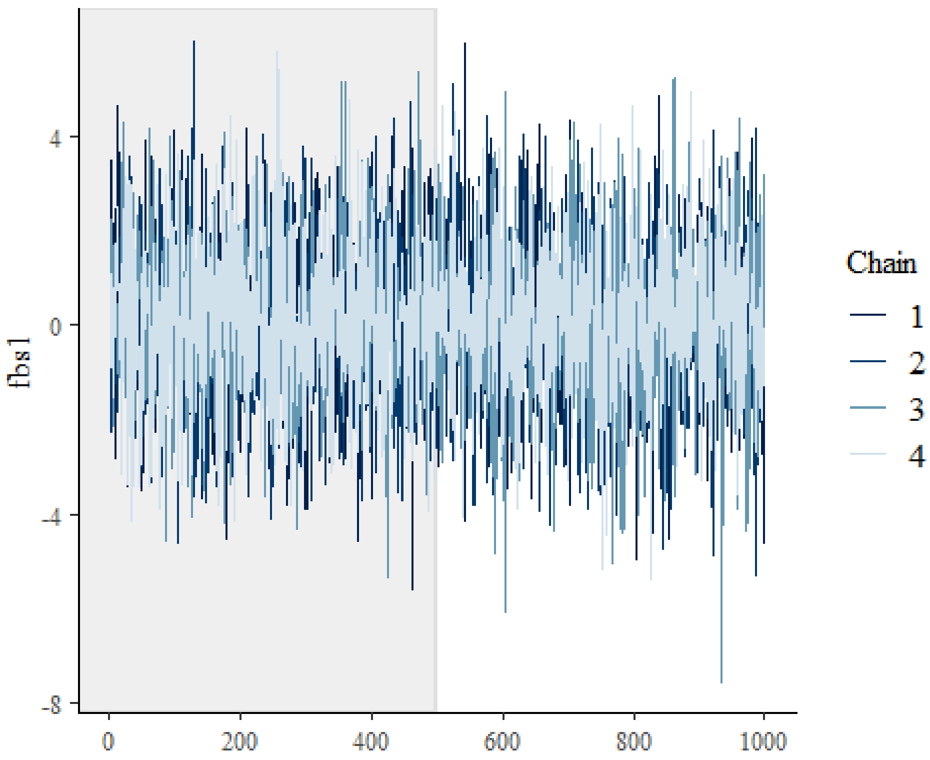
```
mcmc_trace(as.array(post1),  pars = c('chol'), n_warmup = 500)
```



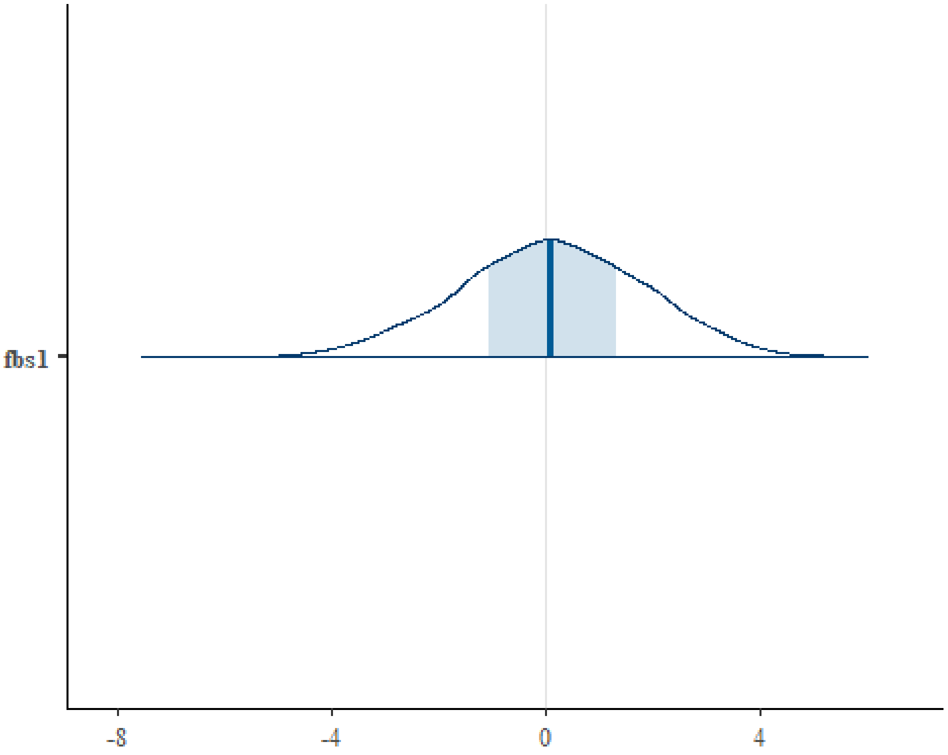
```
mcmc_areas(as.array(post1), pars = c('chol'))
```



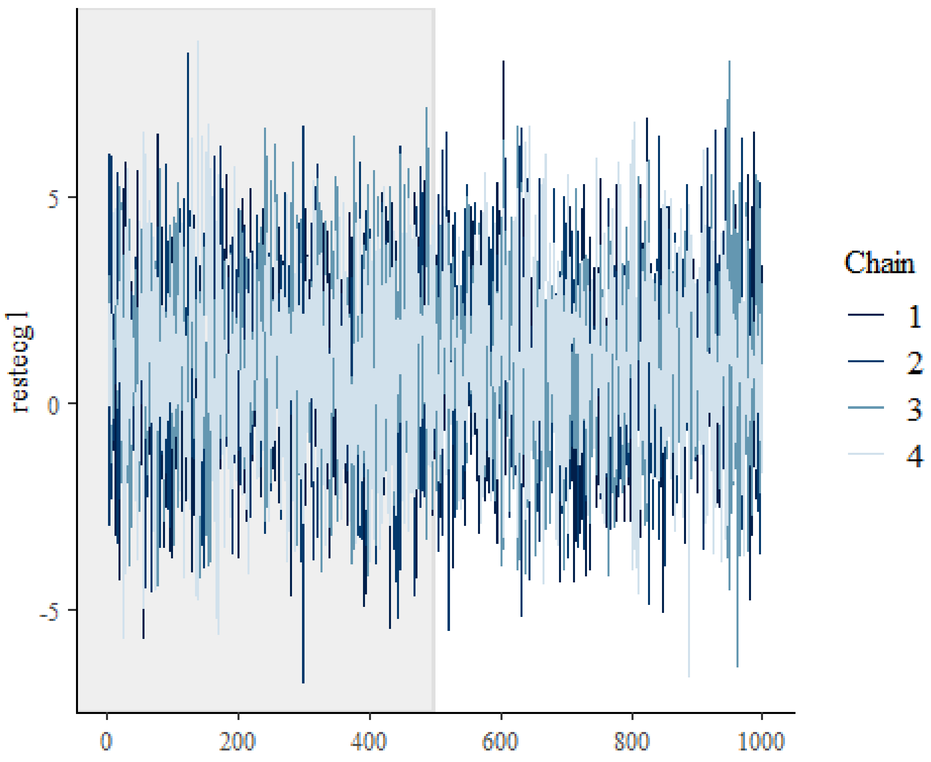
```
mcmc_trace(as.array(post1),  pars = c('fbs1'), n_warmup = 500)
```



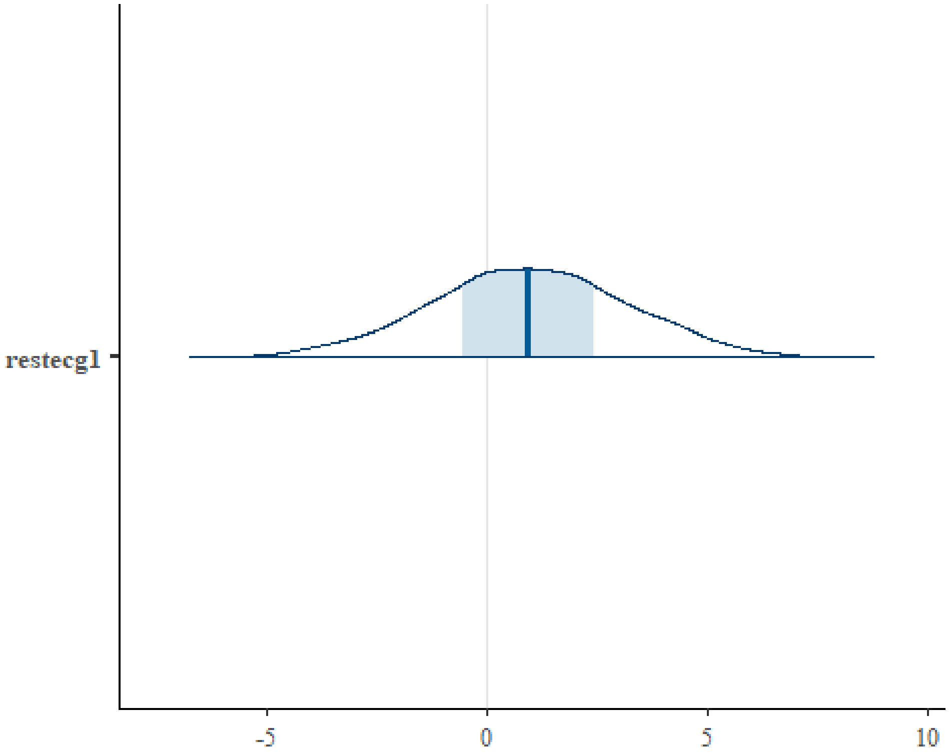
```
mcmc_areas(as.array(post1), pars = c('fbs1'))
```



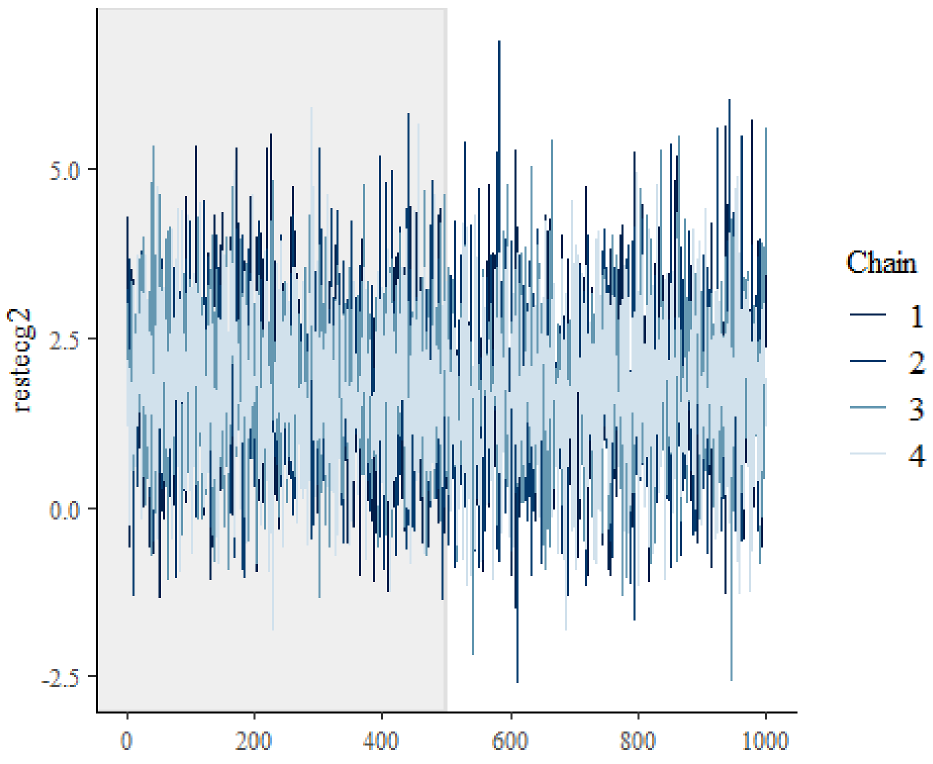
```
mcmc_trace(as.array(post1),  pars = c('restecg1'), n_warmup = 500)
```



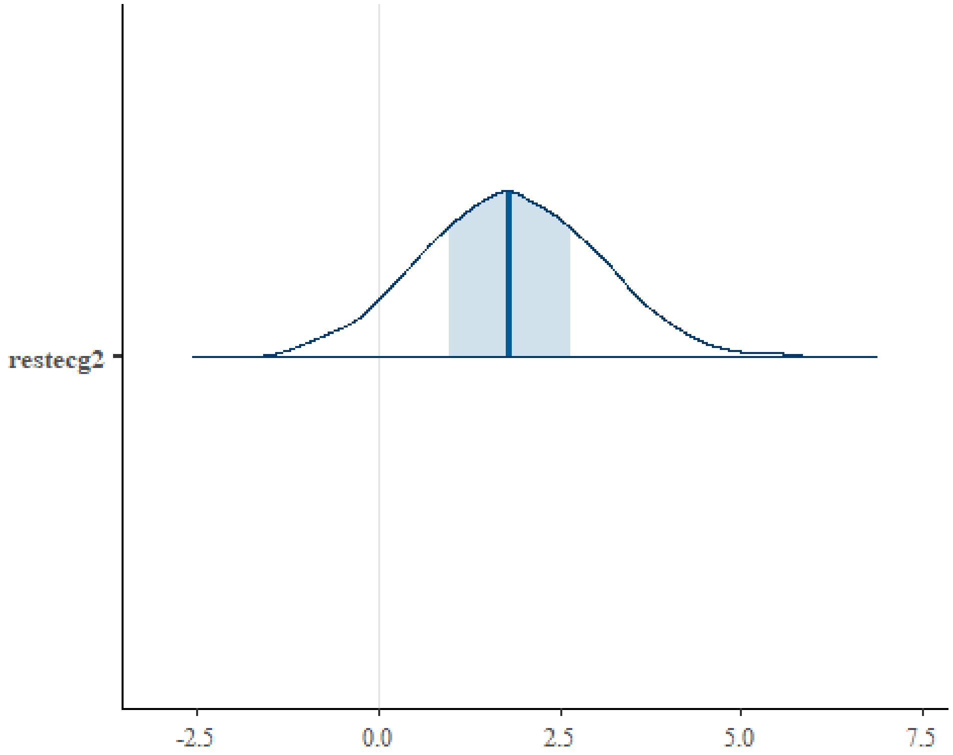
```r
mcmc_areas(as.array(post1), pars = c('restecg1'))
```



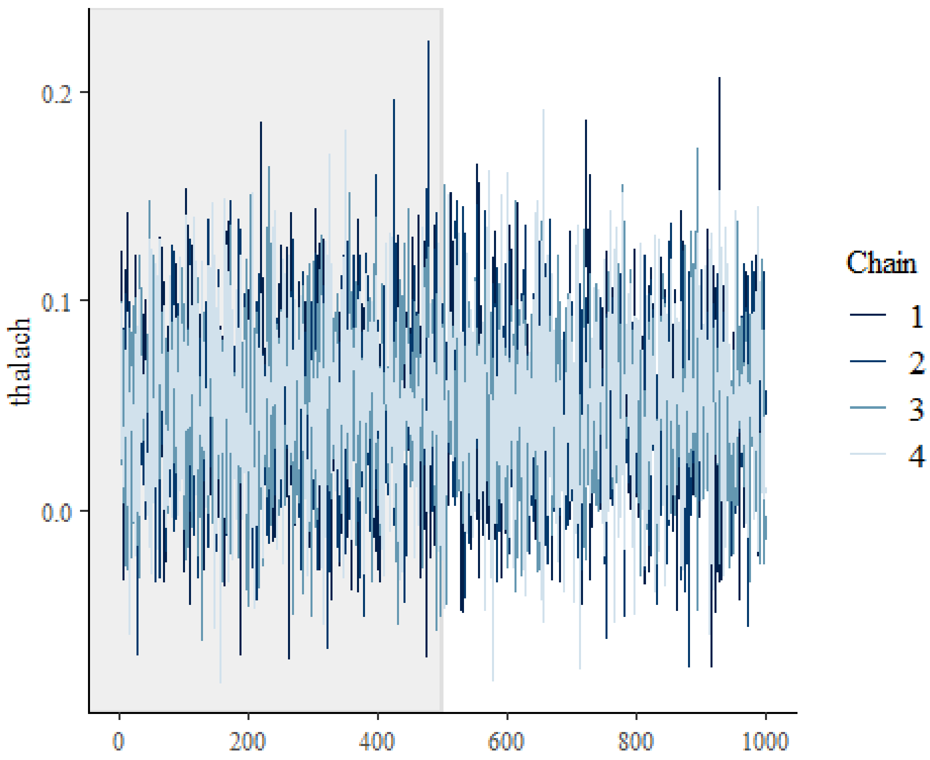
```r
mcmc_trace(as.array(post1),  pars = c('restecg2'), n_warmup = 500)
```



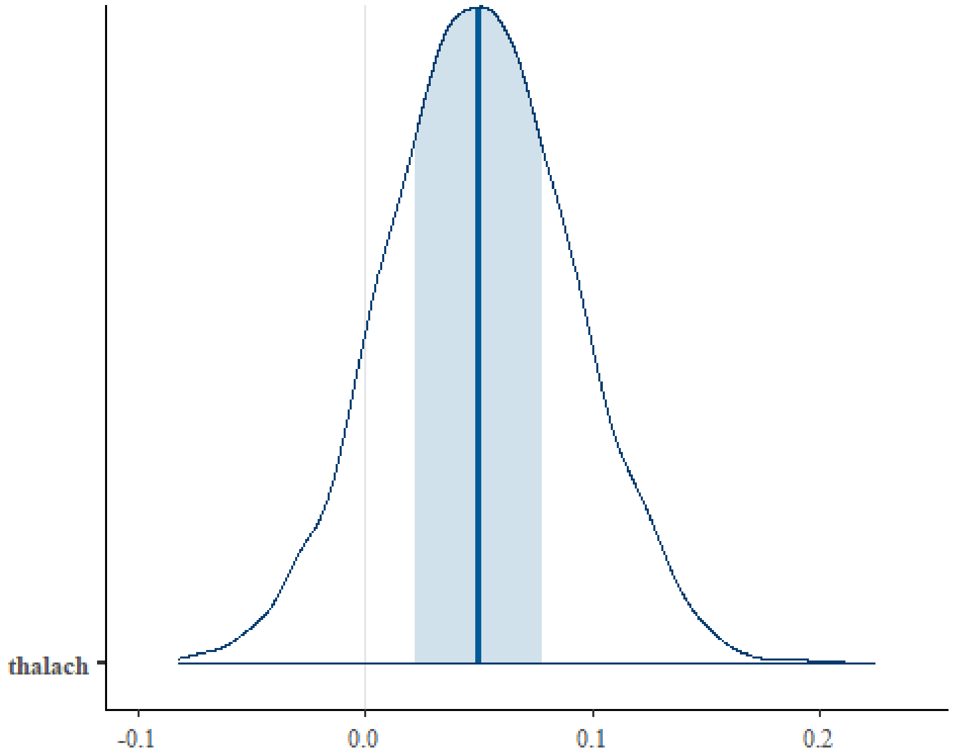
```
mcmc_areas(as.array(post1), pars = c('restecg2'))
```



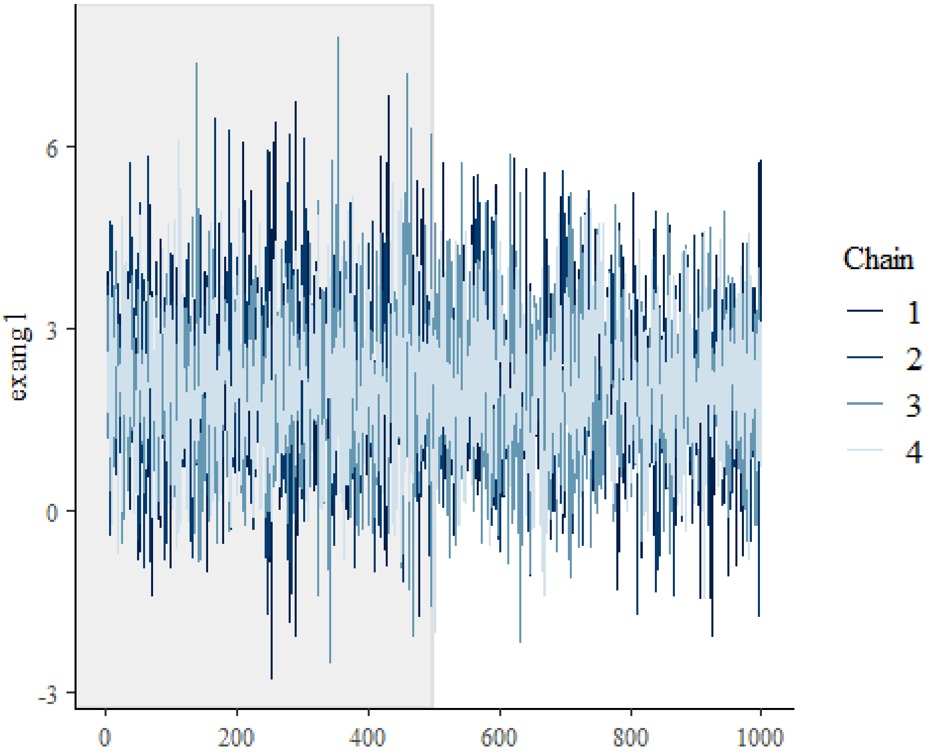
```
mcmc_trace(as.array(post1),  pars = c('thalach'), n_warmup = 500)
```



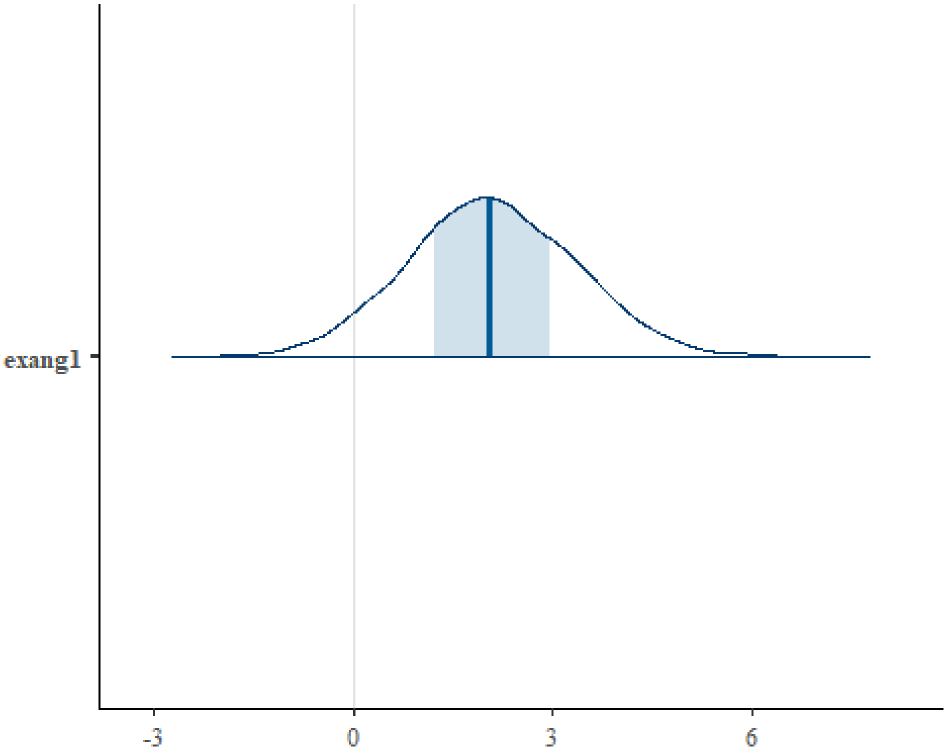
```r
mcmc_areas(as.array(post1), pars = c('thalach'))
```



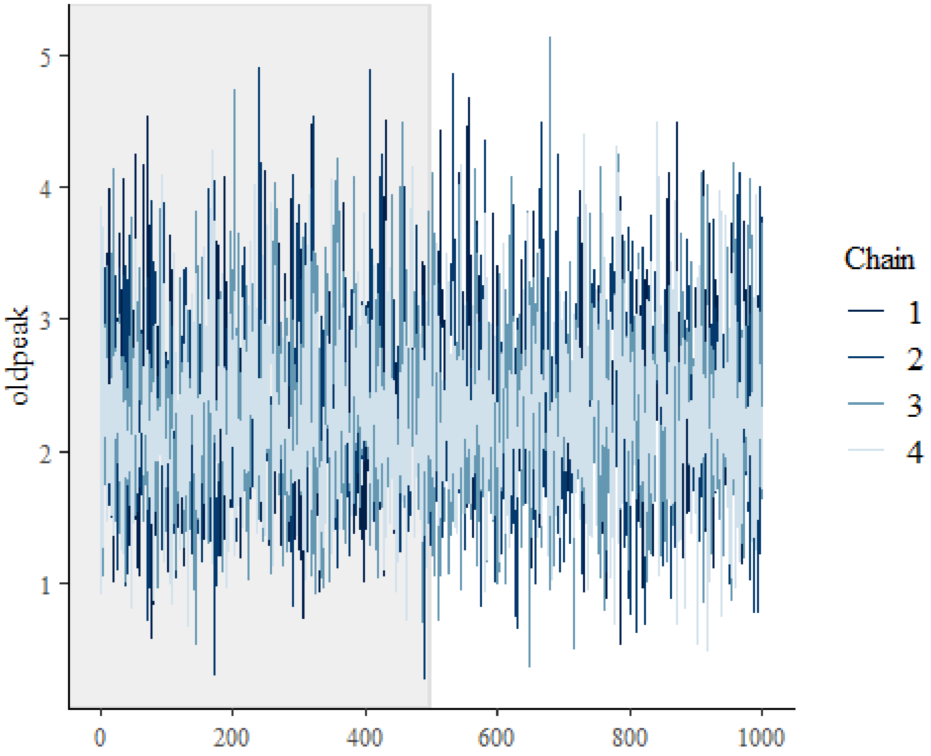
```r
mcmc_trace(as.array(post1),  pars = c('exang1'), n_warmup = 500)
```



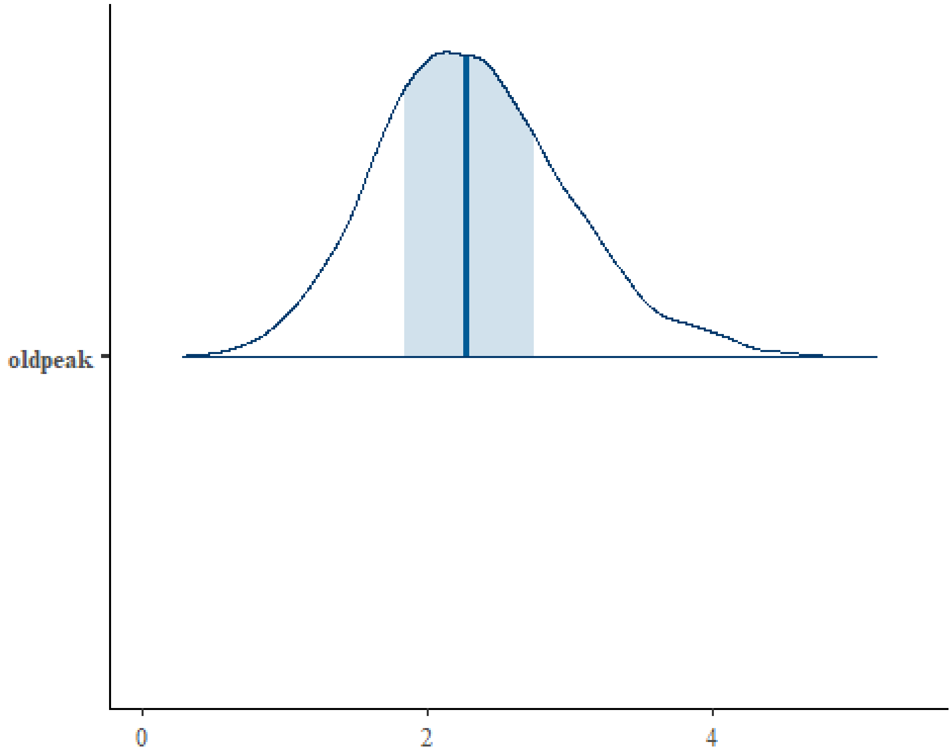
```
mcmc_areas(as.array(post1), pars = c('exang1'))
```



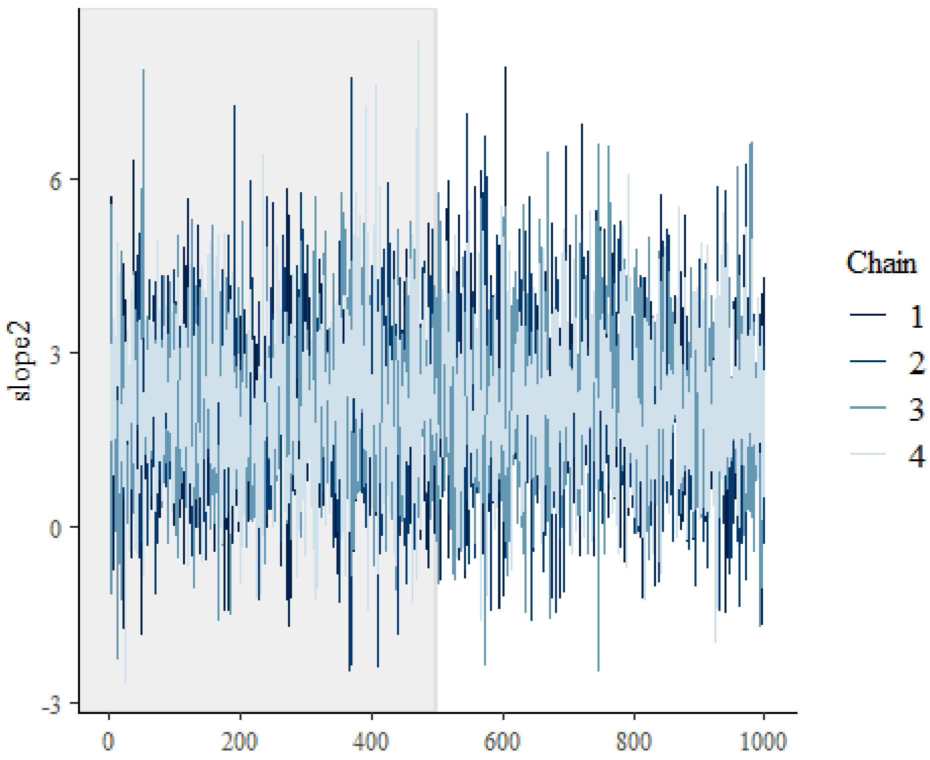
```
mcmc_trace(as.array(post1),  pars = c('oldpeak'), n_warmup = 500)
```



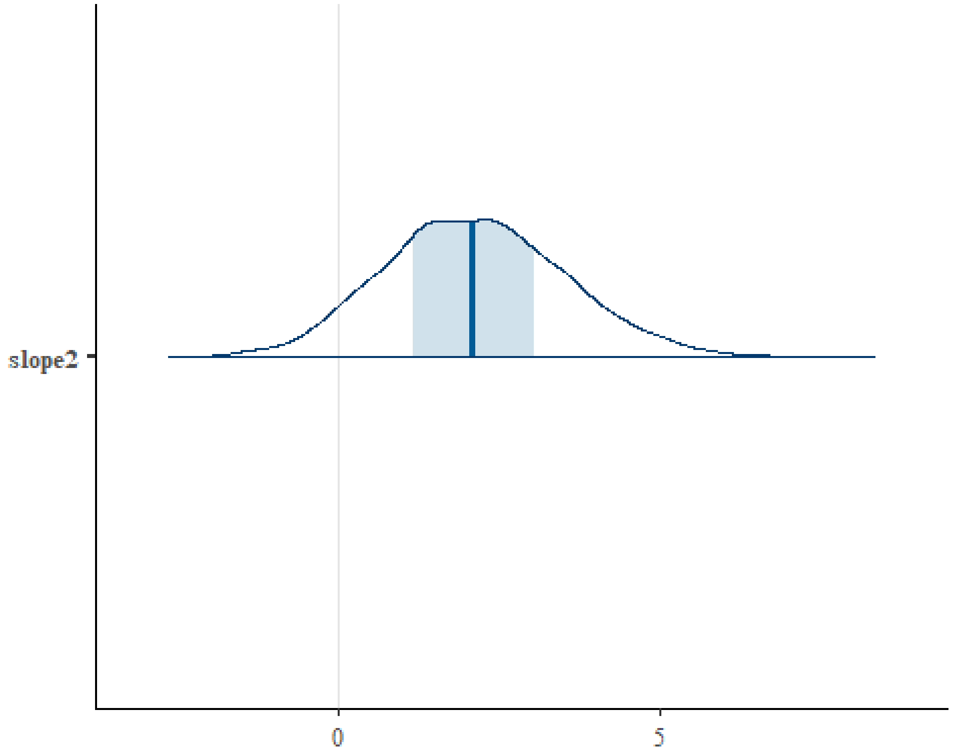
```
mcmc_areas(as.array(post1), pars = c('oldpeak'))
```



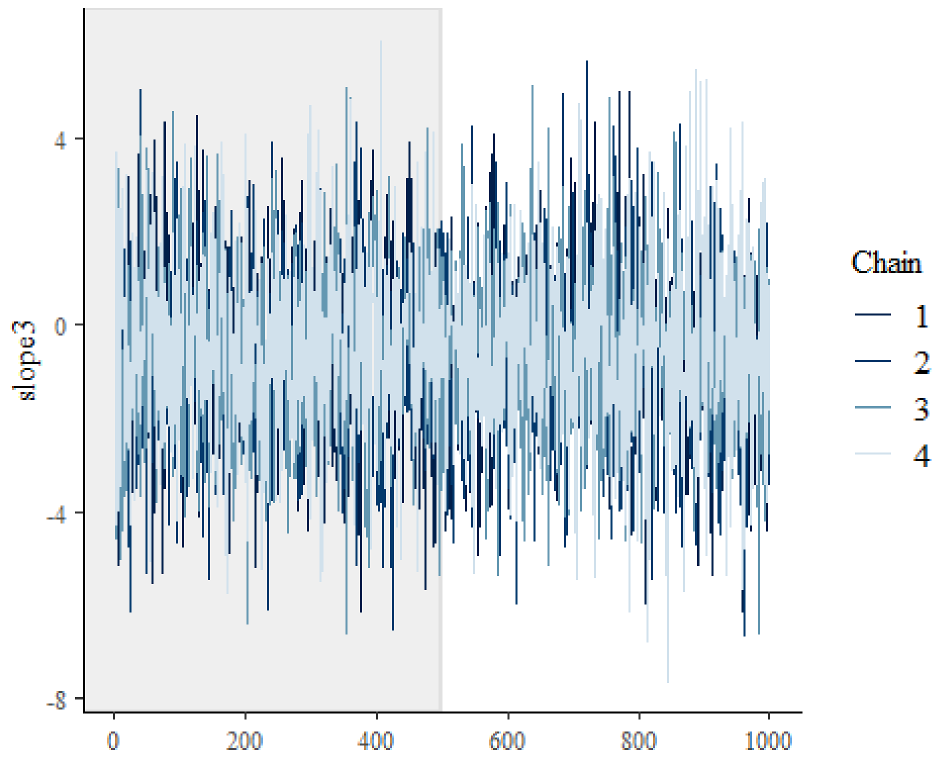
```
mcmc_trace(as.array(post1),  pars = c('slope2'), n_warmup = 500)
```



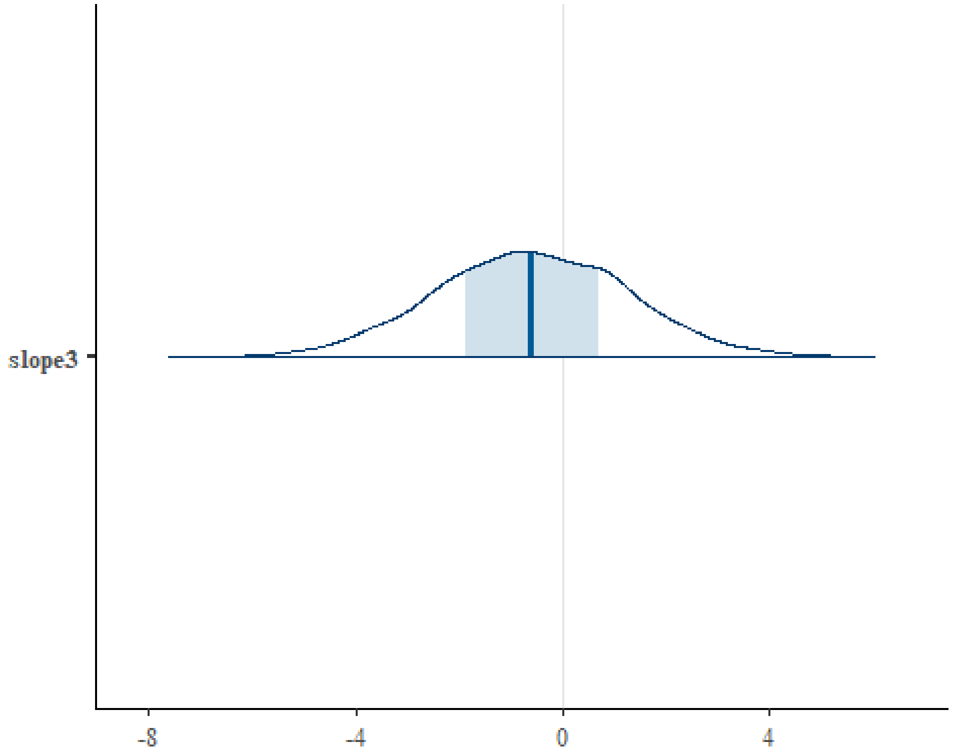
```
mcmc_areas(as.array(post1), pars = c('slope2'))
```



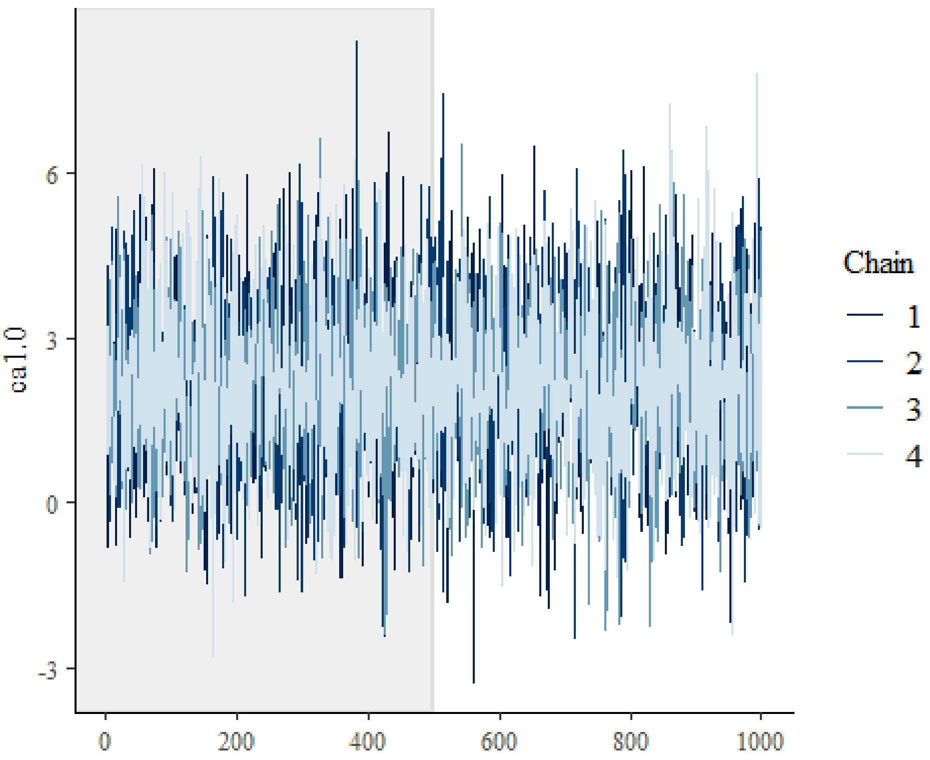
```
mcmc_trace(as.array(post1),  pars = c('slope3'), n_warmup = 500)
```



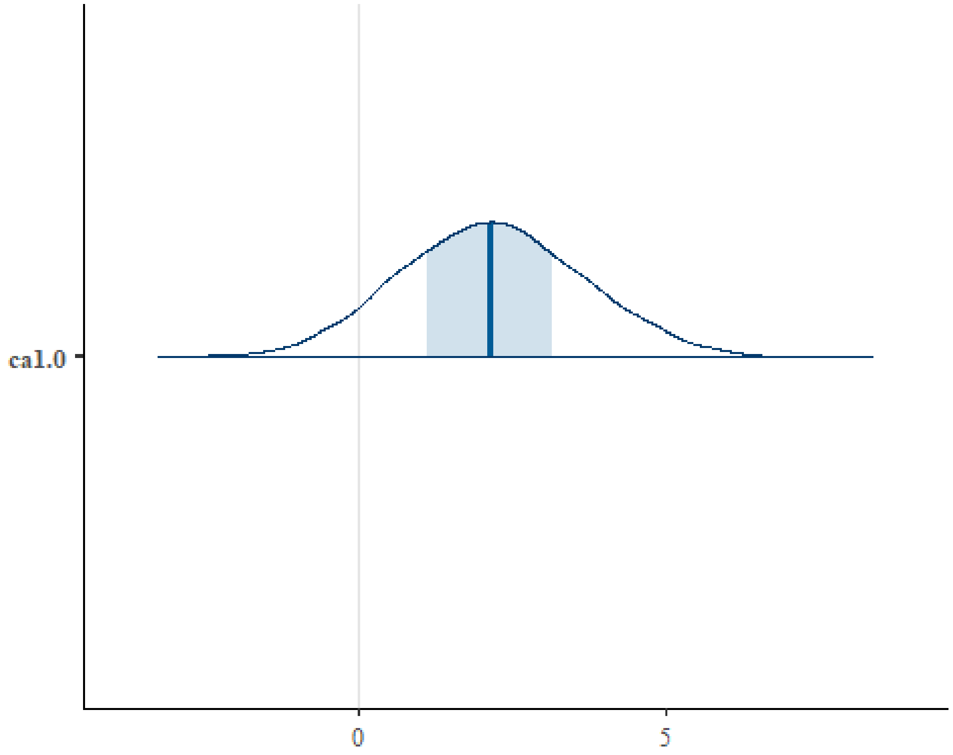
```r
mcmc_areas(as.array(post1), pars = c('slope3'))
```



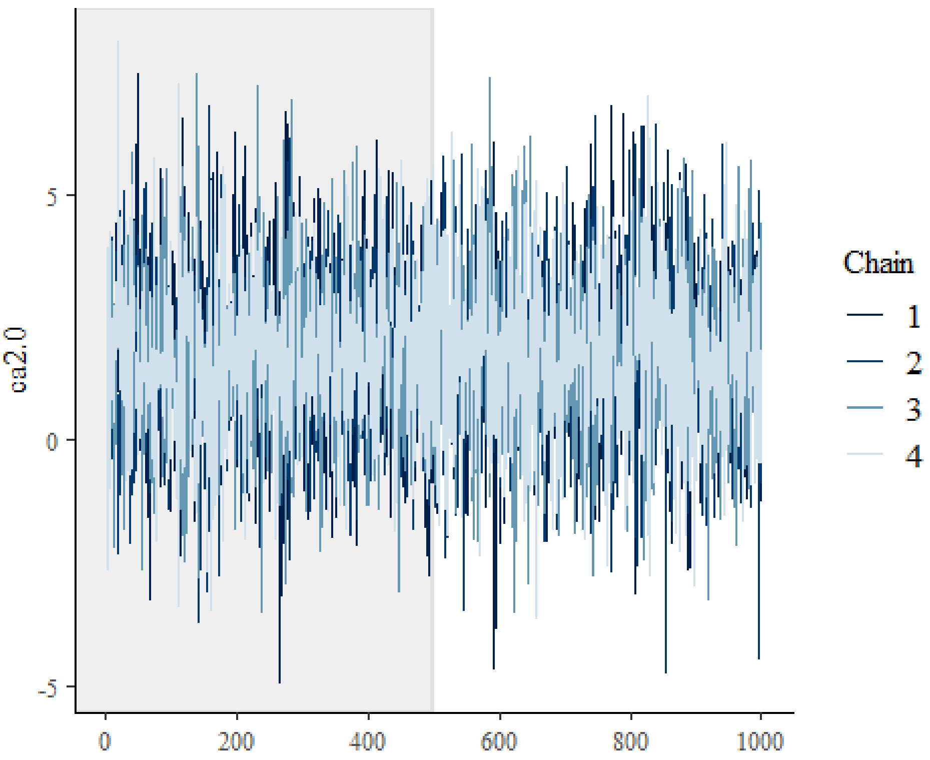
```r
mcmc_trace(as.array(post1),  pars = c('ca1.0'), n_warmup = 500)
```



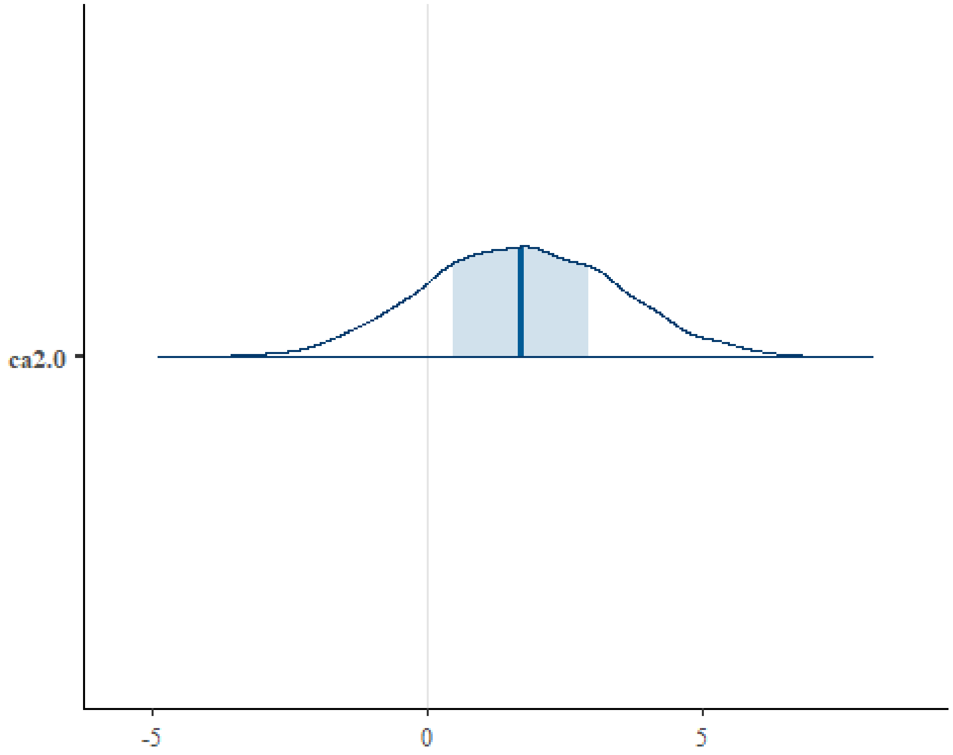
```
mcmc_areas(as.array(post1), pars = c('ca1.0'))
```



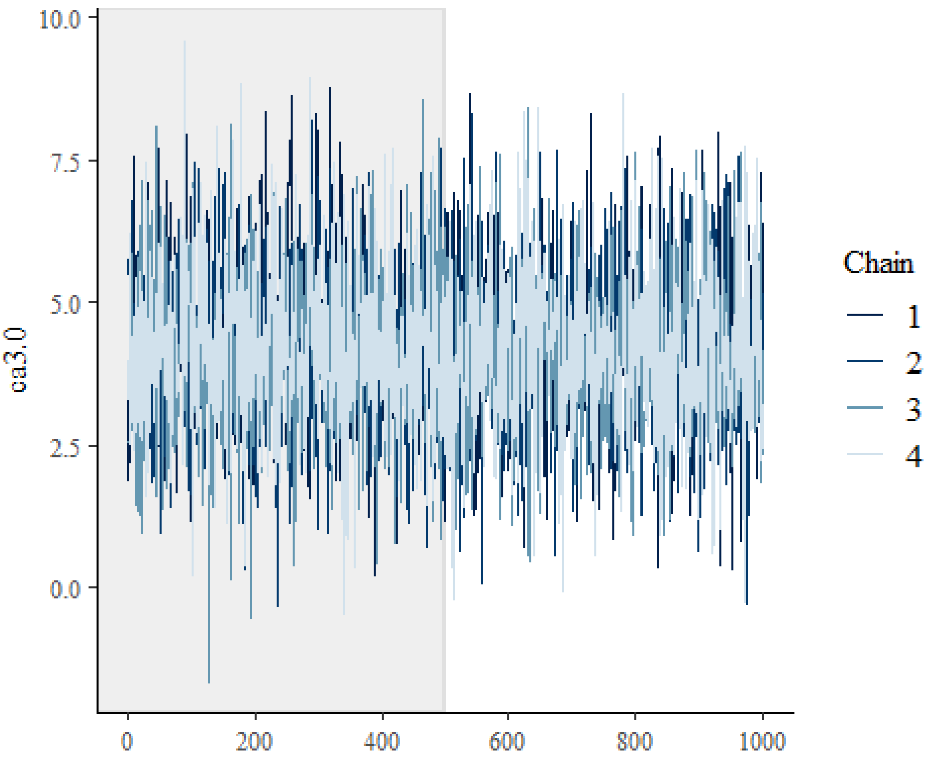
```
mcmc_trace(as.array(post1),  pars = c('ca2.0'), n_warmup = 500)
```



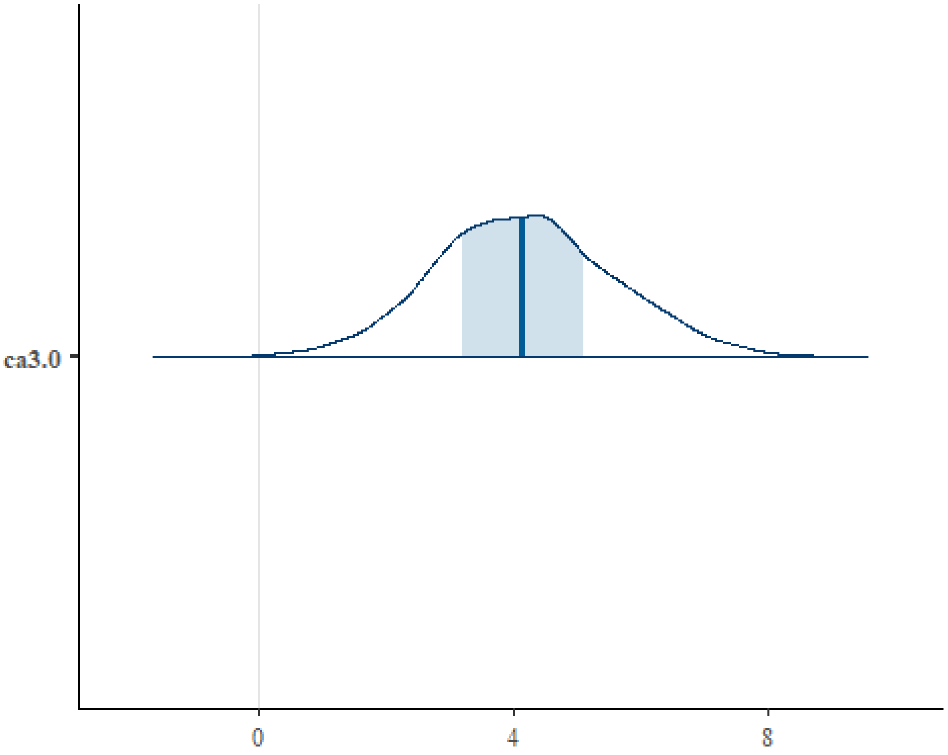
```
mcmc_areas(as.array(post1), pars = c('ca2.0'))
```



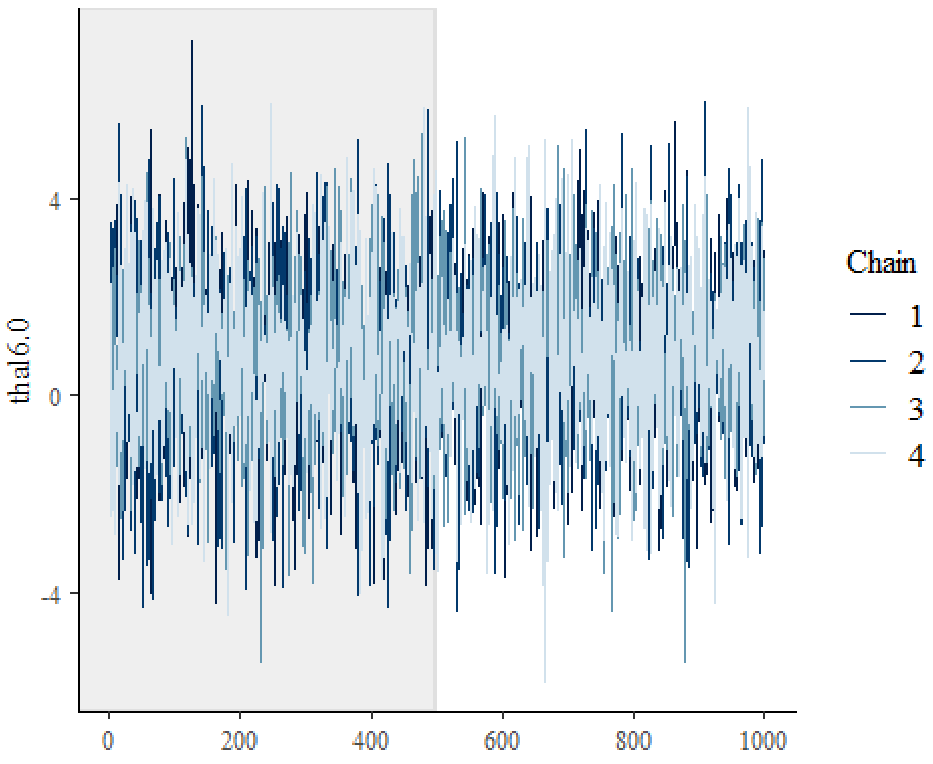
```
mcmc_trace(as.array(post1),  pars = c('ca3.0'), n_warmup = 500)
```



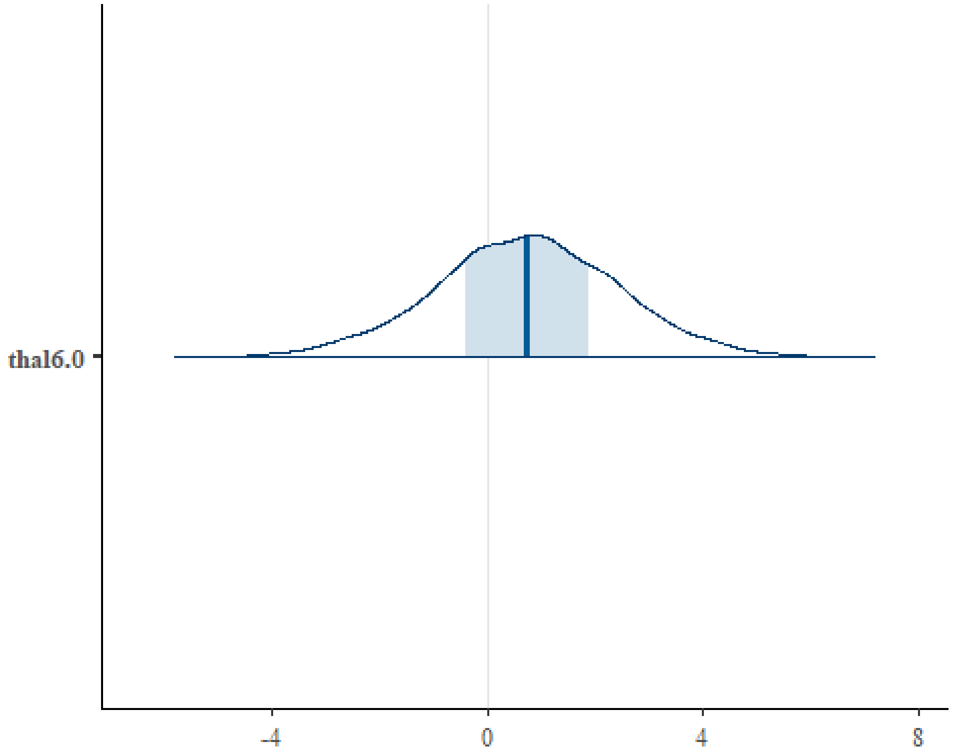
```
mcmc_areas(as.array(post1), pars = c('ca3.0'))
```



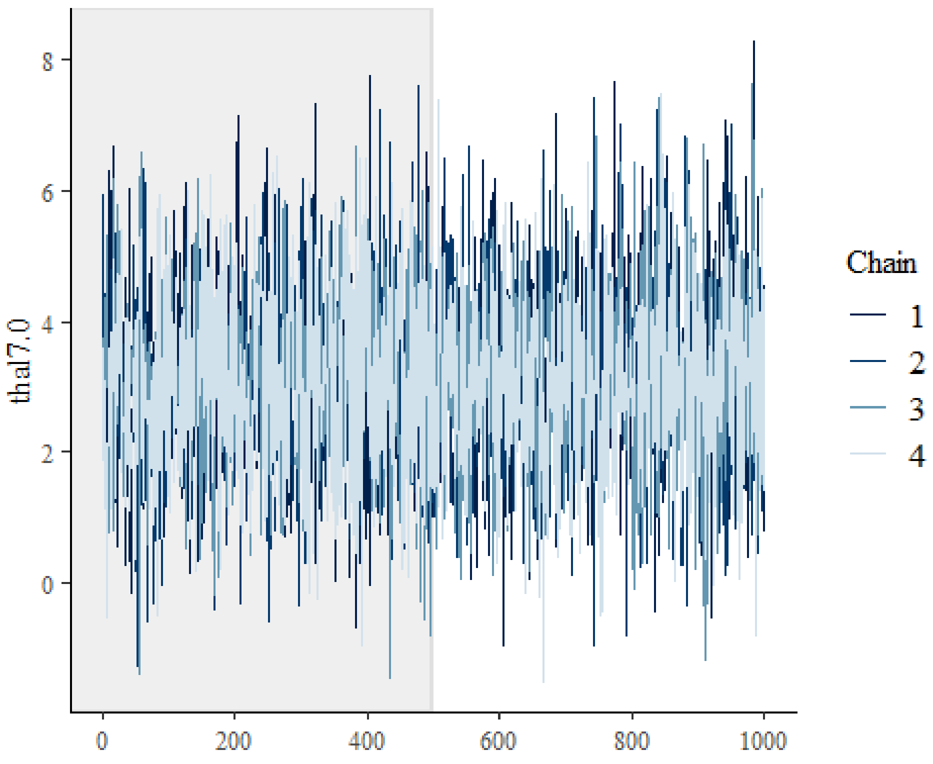
```
mcmc_trace(as.array(post1),  pars = c('thal6.0'), n_warmup = 500)
```



```
mcmc_areas(as.array(post1), pars = c('thal6.0'))
```



```
mcmc_trace(as.array(post1),  pars = c('thal7.0'), n_warmup = 500)
```


thal7.0
8
6
4
2
0
0
200
400
600
800
1000
Chain
1
2
3
4