Applied Bayesian Analysis : NCSU ST 540

Homework 6

Bruce Campbell

For this problem we use the 2016 election data described at https://www4.stat.ncsu.edu/~reich/ABA/code/election2016data with data provided in the R workspace https://www4.stat.ncsu.edu/~reich/ABA/code/election_2008_2016.RData

We restrict our analysis to the counties in North and South Carolina. In JAGS, we fit the logistic regression model

$$P(Z_i = 1) = \frac{1}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j X_{i\ j}}}$$

where Z_i is the binary indicator that GOP support in county i increased by at least 5% from 2012 to 2016 $Z_i = 1 : Yi > 5$ and $Z_i = 0$ otherwise, where Y is the change variable in the R workspace. $X_i j$ are the covariates in the R workspace. We standardize each covariate to have mean zero and variance one before fitting the model. The priors are $\beta_j \sim N(0, \tau^2)$

(1) Fit the model with $\tau = 1$ and $\tau = 100$

First we make some notes on the data and the preprocessing steps.

```
https://www.kaggle.com/benhamner/2016-us-election/data
demographic data on counties from US census
3195 rows and 54 columns.
```

The metadata in the county facts table is located in a dictionary. We might need this for interpretation of the coefficients.

```
county_facts_dictionary.csv
description of the columns in county_facts
https://www.kaggle.com/benhamner/2016-us-election/data
```

The election data

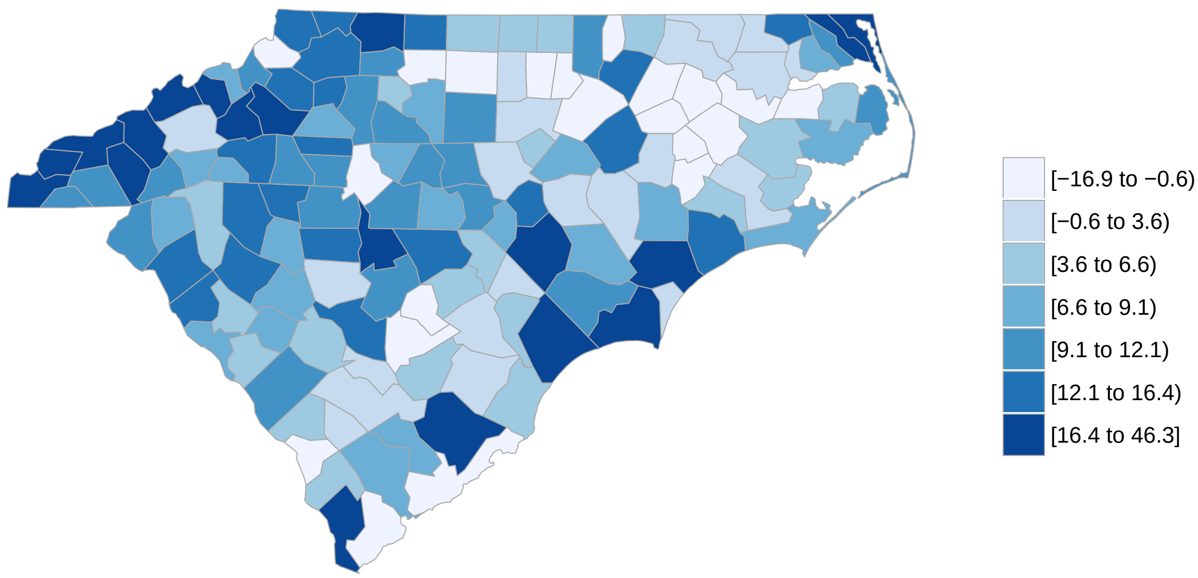
```
County_Election_08_16.csv
https://www4.stat.ncsu.edu/~reich/ABA/code/election2016data
county-level voting patterns in the 2016 Presidential elections
3112 rows and 14 columns
```

The county data is joined to the election data via the key fips_code. We load the processed data below and start our analysis with the joined dataframe all_dat.

```r
library(rjags)
library(coda)
library(choroplethr)
library(modeest)
```

```r
load("election_2008_2016.RData")

carolinas <- all_dat[all_dat$state_abbreviation ==
    "NC" | all_dat$state_abbreviation ==
    "SC", ]
gop.percent.increase <- 100 * (carolinas$gop_2016 -
    carolinas$gop_2012)/carolinas$gop_2012
gop.percent.increase.gt.5 <- gop.percent.increase >
    5


# Borrowed from instructor codebase to
# create our predictors and reposenses
# for the carolinas data set.
fips <- carolinas[, 1]
Y <- round(100 * (carolinas$gop_2016 -
    carolinas$gop_2012)/carolinas$gop_2012,
    1)
Z <- Y > 5

these <- c(3, 7, 10, 15, 20, 21, 25, 27,
    31, 32, 47, 51, 22, 44:45)
X <- as.matrix(carolinas[, these + 15])
X <- scale(X)
names <- dict[these, ]
colnames(X) <- names[, 1]

county_plot <- function(fips, Y, main = "",
    units = "") {
    temp <- as.data.frame(list(region = fips,
        value = Y))
    # county_choropleth(temp,title=main,legend=units)
    county_choropleth(temp, title = main,
        county_zoom = fips)
}
county_plot(fips, Y, "Percent change in GOP support from 2012 to 2016",
    unit = "Percent increase")
```

Percent change in GOP support from 2012 to 2016



Fit with $\tau = 1$

```
n.chains <- 4
DEBUG <- TRUE
if(DEBUG)
{
nSamples <- 2000
} else
{
nSamples <- 2000
}
n <- nrow(X)
tau <- 1
p <- ncol(X)
logistic_model <- "model{
# Likelihood
for(i in 1:n){
Z[i] ~ dbern(q[i])
logit(q[i]) <- intercept +inprod(X[i,],beta[])
}
#Priors
```

```
intercept ~ dnorm(0,tau^2)
for(j in 1:p){
beta[j] ~ dnorm(0,tau^2)
}
}"
model.carolinas <- jags.model(textConnection(logistic_model), data = list(Z=Z,X=X,n=n,p=p,tau=
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 146
##    Unobserved stochastic nodes: 16
##    Total graph size: 2943
##
## Initializing model
```

```
update(model.carolinas, nSamples, progress.bar="none"); # Burnin
samp.coeff <- coda.samples(model.carolinas, variable.names=c("intercept","beta"),n.iter=2*nSamp
```

Fit with $\tau = 100$

```
tau <- 100
model.carolinas.uninformative <- jags.model(textConnection(logistic_model), data = list(Z=Z,X=
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 146
##    Unobserved stochastic nodes: 16
##    Total graph size: 2943
##
## Initializing model
```

```
update(model.carolinas.uninformative, nSamples, progress.bar="none"); # Burnin
samp.coeff.uninformative <- coda.samples(model.carolinas.uninformative, variable.names=c("inte
```

(2) Assess convergence of the sampler for both priors.

In this section we sample from our model after burn in. Although all of the plots are not presented we assesed convergence by; - viewing the time sereies for the intercept and each of the predictors. For this we utilized the coda package. - ran multiple chains and viewed evaluated the autocorrelation plots. - calculated the posterior means for the intercept and the

j - utilized the mlv funtions in the modeest to calculate the MAP estimated of the posterior modes - we fit a frequentist model an evaluated the estimated coefficients against the posterior means and modes - compared the 95% prediction intervals for the intercepts against the p-values from the logistic regression maximum likelihood model

Code for this is below, we run some of it conditionlally though the DEBUG variable.

We did run the model without standardizing the feature data and noted evidence that the chain might be experienceing convergence issues. There was significant autocorrelation of the chains when the data was not standardized.

$\tau = 1$ Posterior quantiles

```r
summary(samp.coeff)
```

```
##
## Iterations = 3001:7000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 4000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean     SD Naive SE Time-series SE
## beta[1]     1.06357 0.4867 0.003848       0.011069
## beta[2]    -0.67248 0.3640 0.002877       0.006844
## beta[3]    -1.86919 0.3942 0.003117       0.007564
## beta[4]    -0.77015 0.2961 0.002341       0.005753
## beta[5]    -0.39419 0.4917 0.003887       0.012369
## beta[6]    -1.30169 0.5617 0.004440       0.014676
## beta[7]     0.84576 0.3950 0.003122       0.007896
## beta[8]     0.15615 0.5563 0.004398       0.016088
## beta[9]    -0.49776 0.6196 0.004898       0.018552
## beta[10]   -0.32148 0.4817 0.003808       0.012089
## beta[11]   -0.26016 0.3309 0.002616       0.005338
## beta[12]   -0.26294 0.5765 0.004558       0.011287
## beta[13]    0.18763 0.5564 0.004399       0.011965
## beta[14]    0.09525 0.5032 0.003978       0.009277
## beta[15]   -0.64304 0.5922 0.004681       0.011063
## intercept   0.98352 0.2664 0.002106       0.003538
##
## 2. Quantiles for each variable:
##
##                2.5%     25%     50%       75%    97.5%
## beta[1]     0.13510  0.7298  1.0529  1.386453  2.05647
## beta[2]    -1.38024 -0.9181 -0.6748 -0.428822  0.03981
## beta[3]    -2.67347 -2.1278 -1.8533 -1.598304 -1.13459
## beta[4]    -1.35798 -0.9686 -0.7650 -0.570893 -0.19383
## beta[5]    -1.36480 -0.7219 -0.3908 -0.062090  0.54145
```

```
## beta[6]   -2.42764 -1.6708 -1.2994 -0.925595 -0.22689
## beta[7]    0.08255  0.5772  0.8455  1.116196  1.61278
## beta[8]   -0.93562 -0.2117  0.1539  0.528308  1.25477
## beta[9]   -1.72589 -0.9153 -0.5010 -0.069653  0.71114
## beta[10]  -1.25616 -0.6454 -0.3257 -0.006709  0.64176
## beta[11]  -0.93387 -0.4764 -0.2571 -0.037302  0.37946
## beta[12]  -1.44960 -0.6351 -0.2342  0.134887  0.81051
## beta[13]  -0.92916 -0.1797  0.1960  0.564461  1.25727
## beta[14]  -0.93372 -0.2338  0.1048  0.439479  1.05220
## beta[15]  -1.84681 -1.0292 -0.6231 -0.230516  0.45618
## intercept  0.47208  0.8011  0.9800  1.162438  1.50960
```

$\tau = 1$ Sample again and estimate the mean and MAP mode of the posterior dostributions.

```
samp.coeff.jags <- jags.samples(model.carolinas,
    variable.names = c("intercept", "beta"),
    n.iter = nSamples, progress.bar = "none")
posterior_means <- lapply(samp.coeff.jags,
    apply, 1, "mean")
pander(posterior_means, caption = "posterior means second sample")
```

- **beta**: *1.032*, *-0.6955*, *-1.853*, *-0.7445*, *-0.3285*, *-1.313*, *0.8631*, *0.1879*, *-0.5572*, *-0.3307*, *-0.2712*, *-0.2654*, *0.1734*, *0.107* and *-0.6303*
- **intercept**: *0.9824*

```
posterior_modes <- lapply(samp.coeff.jags,
    apply, 1, "mlv")
posterior_modes
```

```
## $beta
## $beta[[1]]
## Mode (most likely value): 0.9639632
## Bickel's modal skewness: 0.08825
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[2]]
## Mode (most likely value): -0.6782029
## Bickel's modal skewness: -0.0335
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[3]]
## Mode (most likely value): -1.891806
## Bickel's modal skewness: 0.09025
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[4]]
```

```
## Mode (most likely value): -0.7244063
## Bickel's modal skewness: -0.04875
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[5]]
## Mode (most likely value): -0.2972024
## Bickel's modal skewness: -0.0415
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[6]]
## Mode (most likely value): -1.268088
## Bickel's modal skewness: -0.031
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[7]]
## Mode (most likely value): 0.8800287
## Bickel's modal skewness: -0.0395
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[8]]
## Mode (most likely value): 0.1687014
## Bickel's modal skewness: 0.026
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[9]]
## Mode (most likely value): -0.5275032
## Bickel's modal skewness: -0.0535
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[10]]
## Mode (most likely value): -0.3465806
## Bickel's modal skewness: 0.01975
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[11]]
## Mode (most likely value): -0.2510136
## Bickel's modal skewness: -0.02475
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[12]]
## Mode (most likely value): -0.1984057
## Bickel's modal skewness: -0.06475
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[13]]
## Mode (most likely value): 0.2127147
## Bickel's modal skewness: -0.0395
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
```

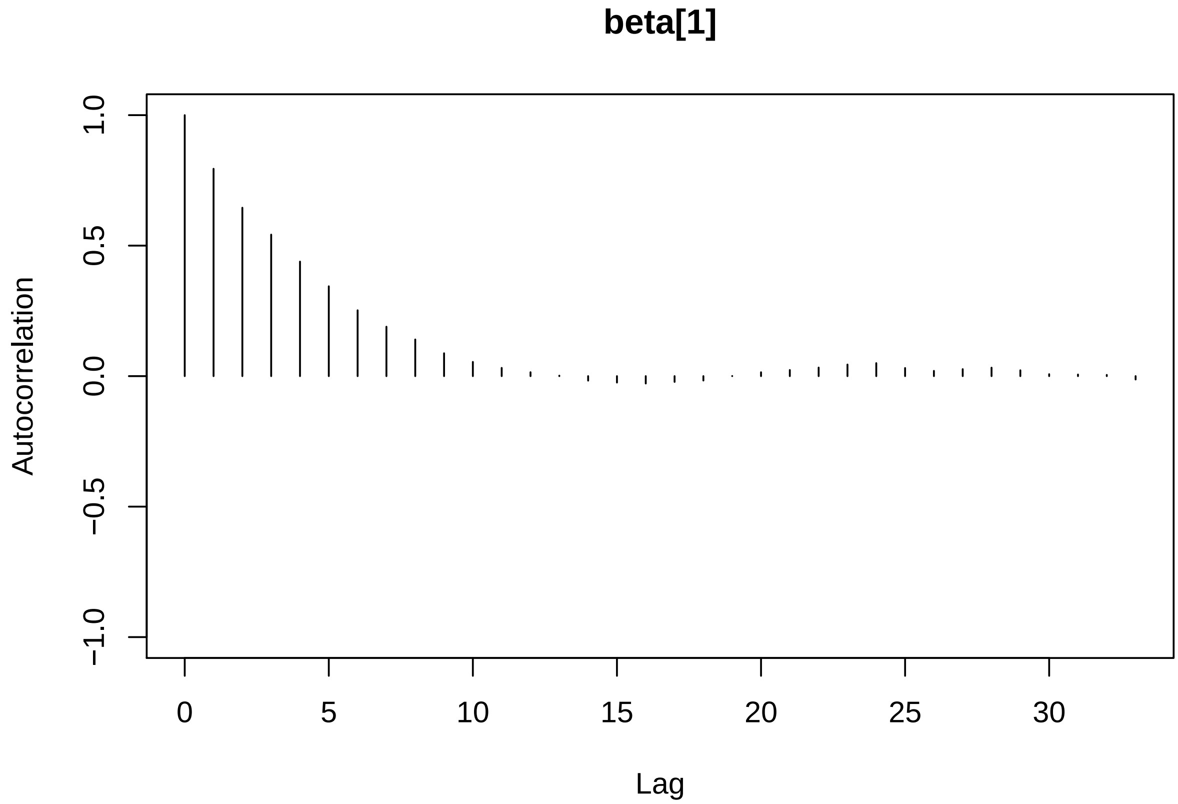
```
##
## $beta[[14]]
## Mode (most likely value): 0.08226178
## Bickel's modal skewness: 0.04575
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[15]]
## Mode (most likely value): -0.5754378
## Bickel's modal skewness: -0.037
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
##
## $intercept
## $intercept[[1]]
## Mode (most likely value): 0.9934523
## Bickel's modal skewness: -0.04625
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
```

$\tau = 1$ Plot the time series, empirical posterior distribution, and the autocoerrelation

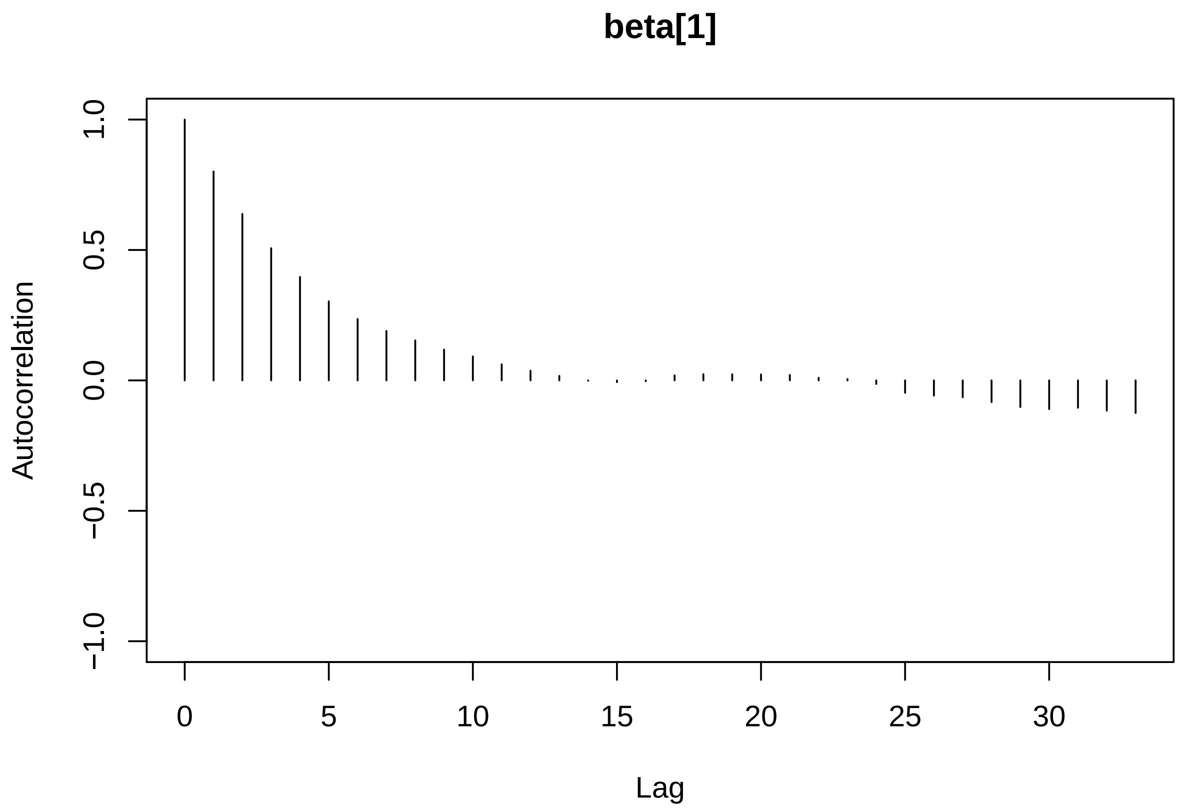
fucntion for the coefficients

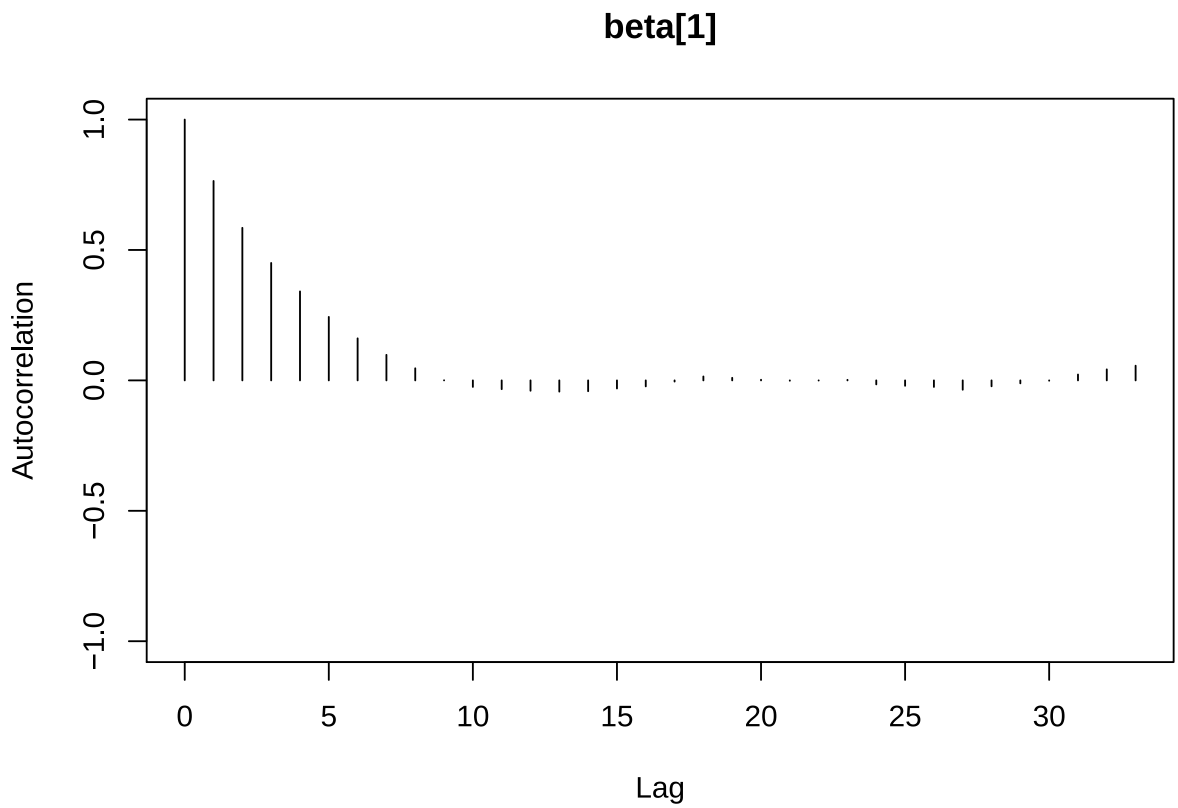
We only plot the intercept for the final report. Set the DEBUG flag to TRUE in order to include all of the coefficients.

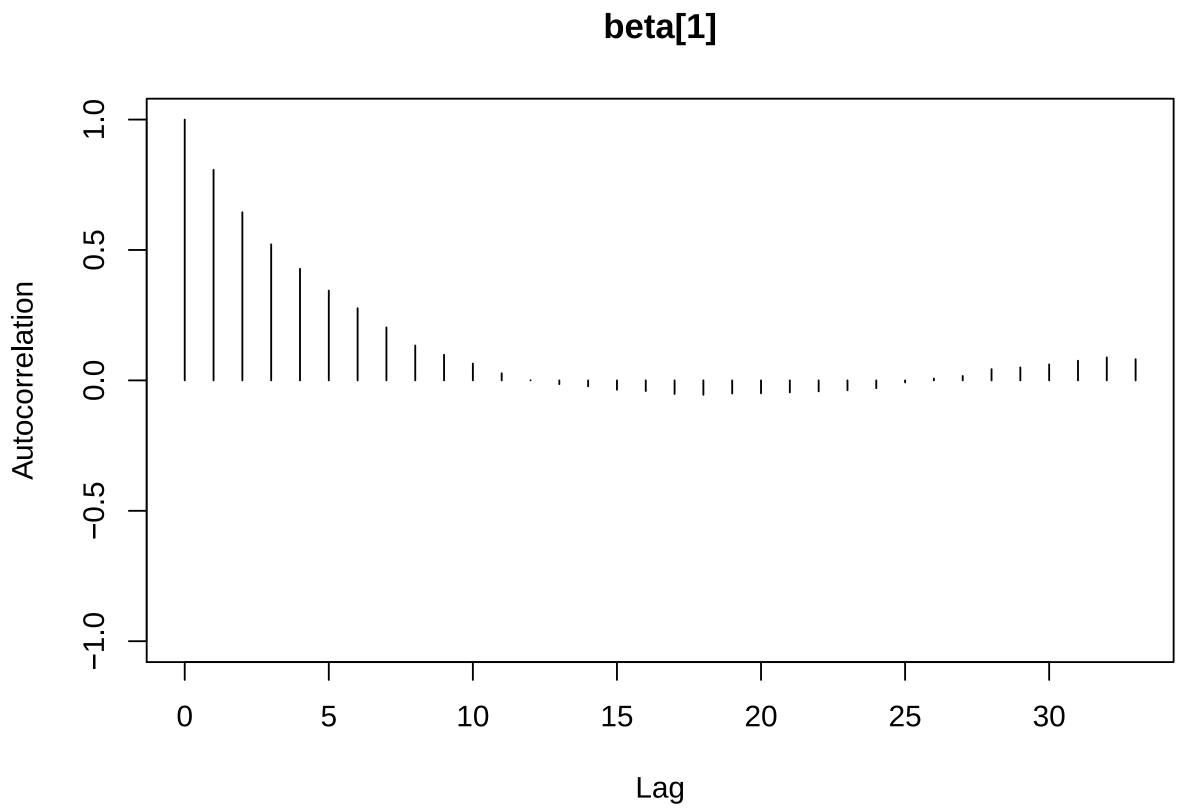
```
if (DEBUG) {
    for (i in 1:p) {
        samp.coeff <- coda.samples(model.carolinas,
            variable.names = c(paste("beta[",
                i, "]", sep = "")), n.iter = nSamples,
            progress.bar = "none")
        autocorr.plot(samp.coeff)
        plot(samp.coeff)
    }
    samp.coeff <- coda.samples(model.carolinas,
        variable.names = "intercept", n.iter = nSamples,
        progress.bar = "none")
    autocorr.plot(samp.coeff)
    gelman.plot(samp.coeff)
    plot(samp.coeff)
} else {
    samp.coeff <- coda.samples(model.carolinas,
        variable.names = "intercept", n.iter = nSamples,
        progress.bar = "none")
    autocorr.plot(samp.coeff)
    gelman.plot(samp.coeff)
    plot(samp.coeff)
}
```

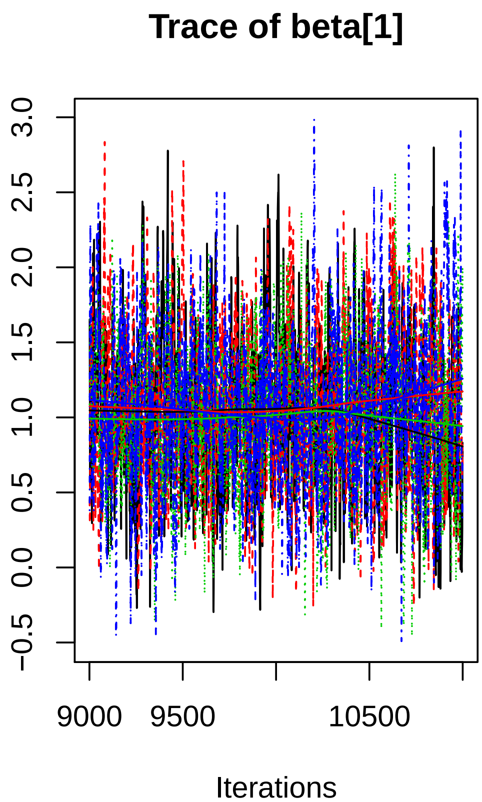


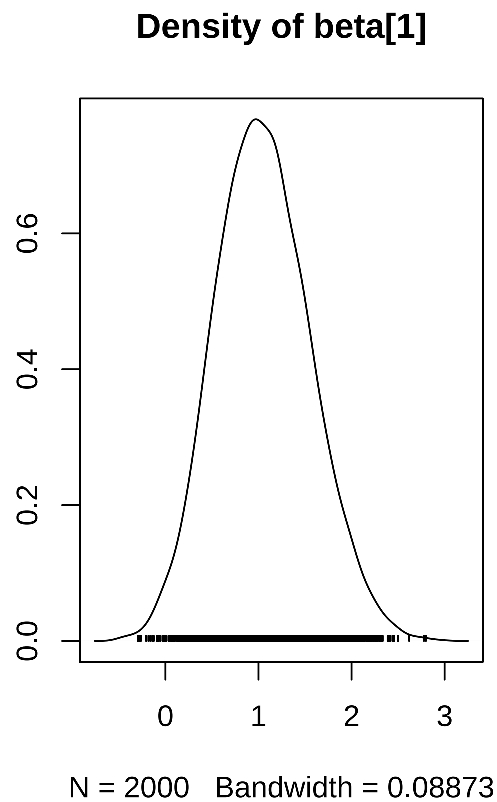
beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

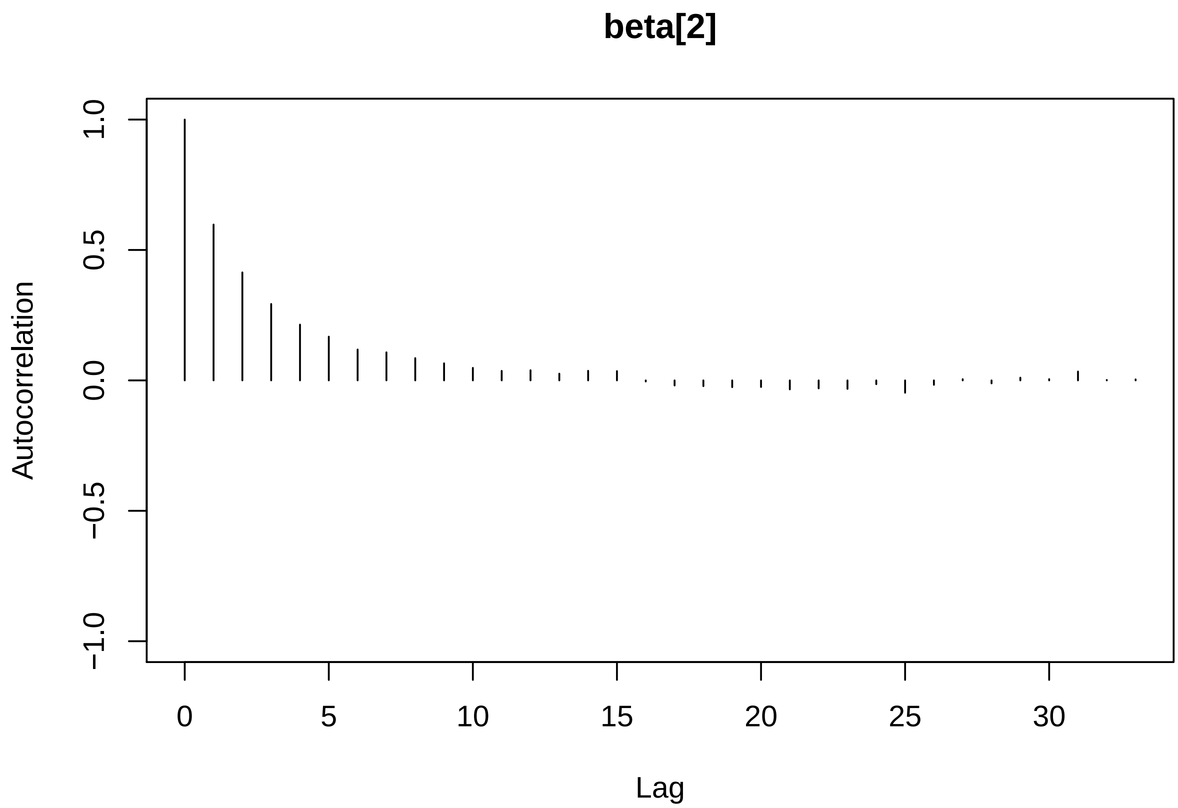

beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

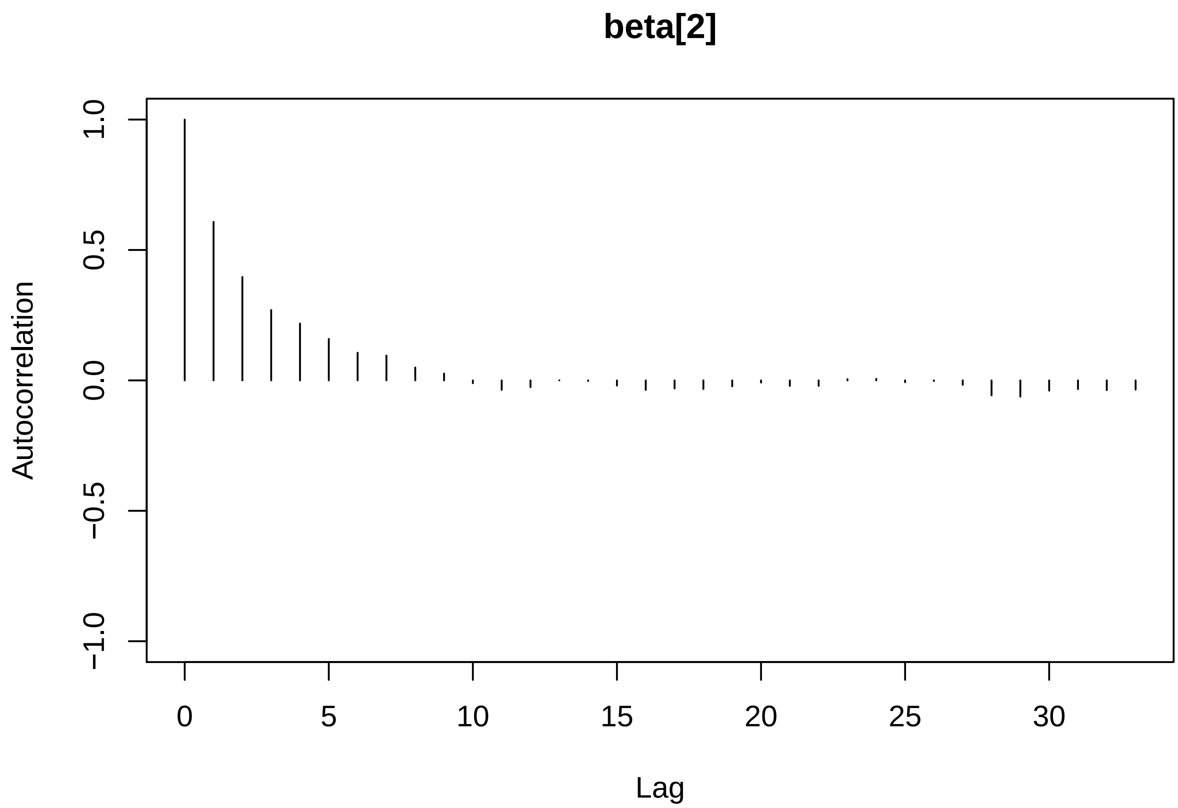

beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

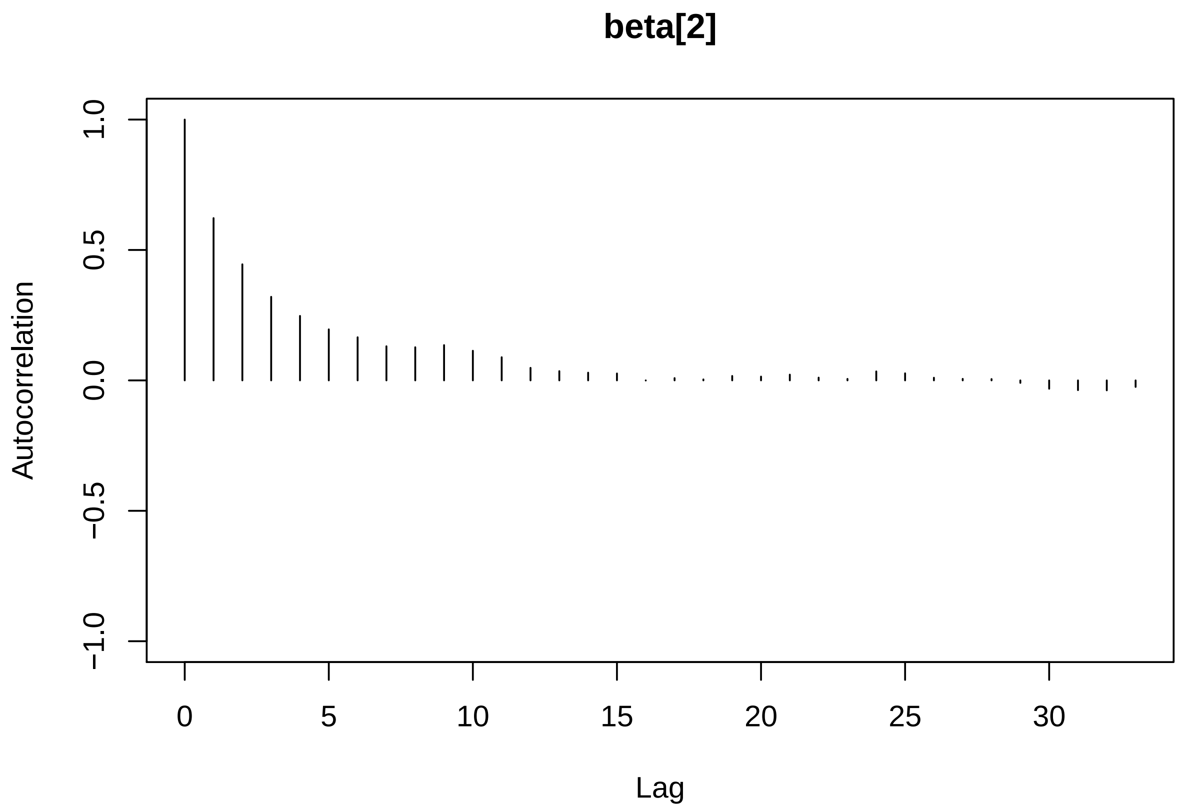

beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

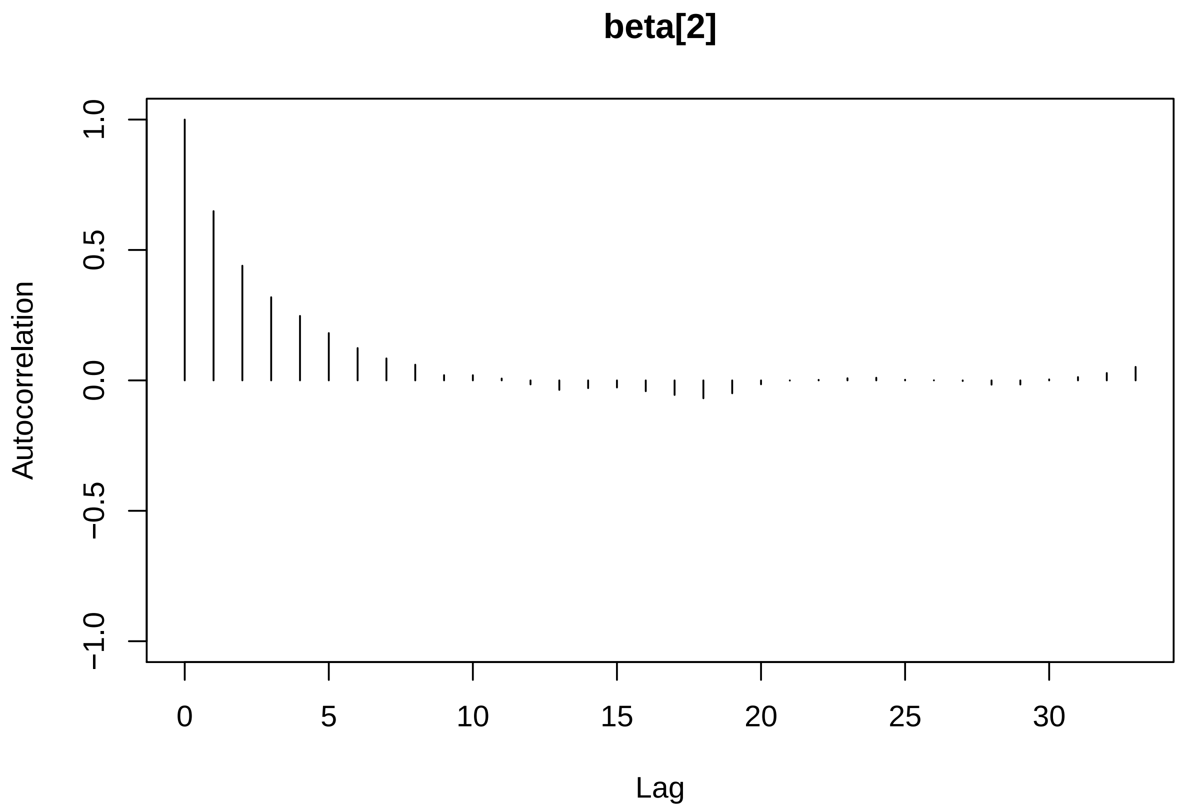

Trace of beta[1]
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
9000
9500
10500
Iterations


Density of beta[1]
0.6
0.4
0.2
0.0
0
1
2
3
N = 2000 Bandwidth = 0.08873

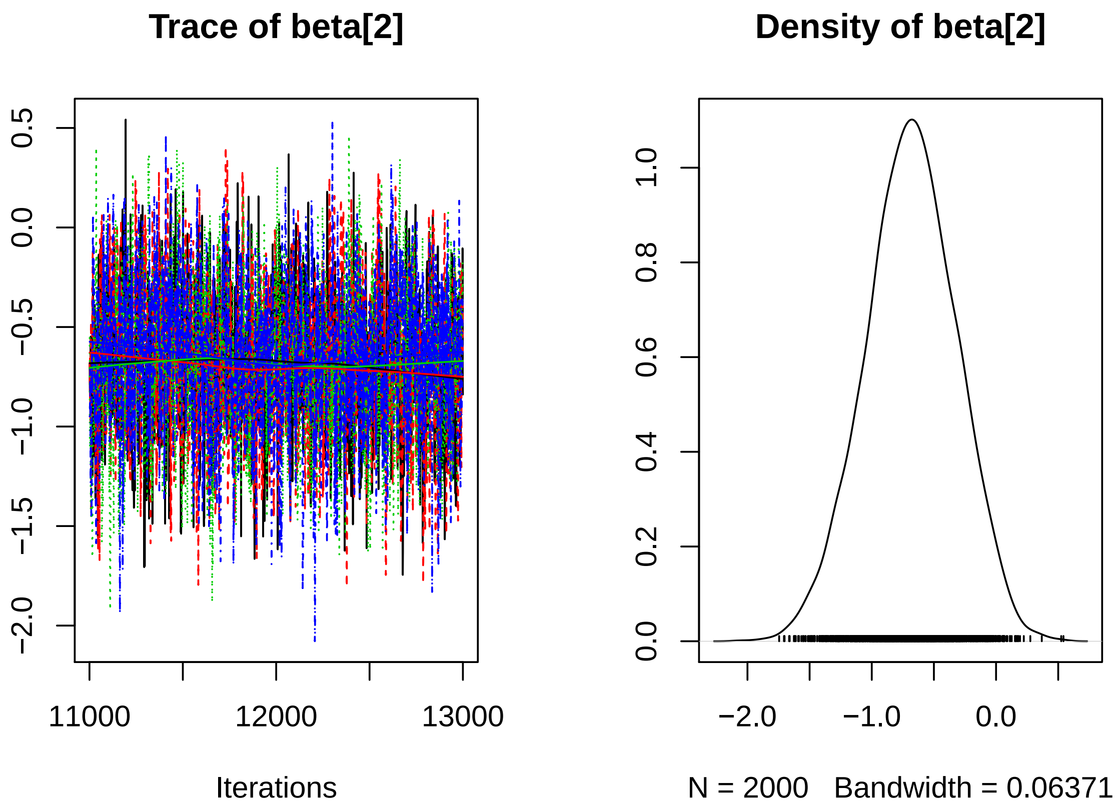

beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

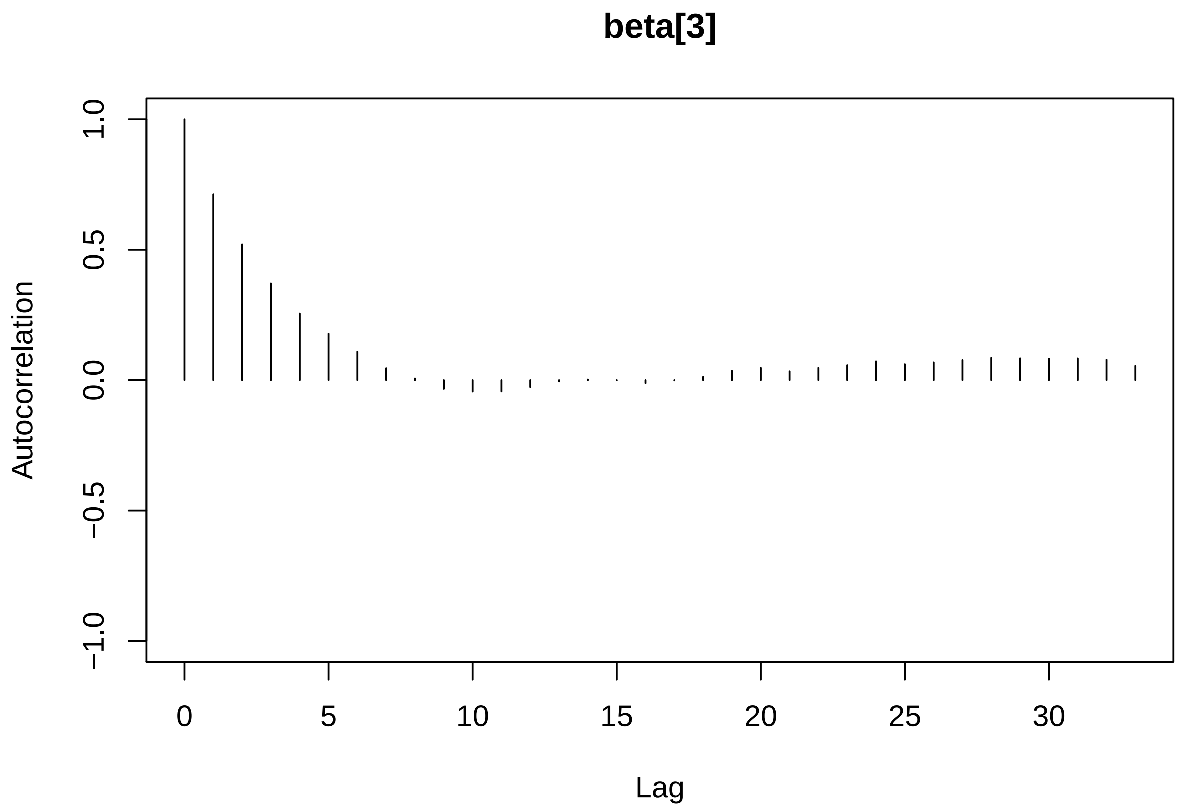

beta[2]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag


beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

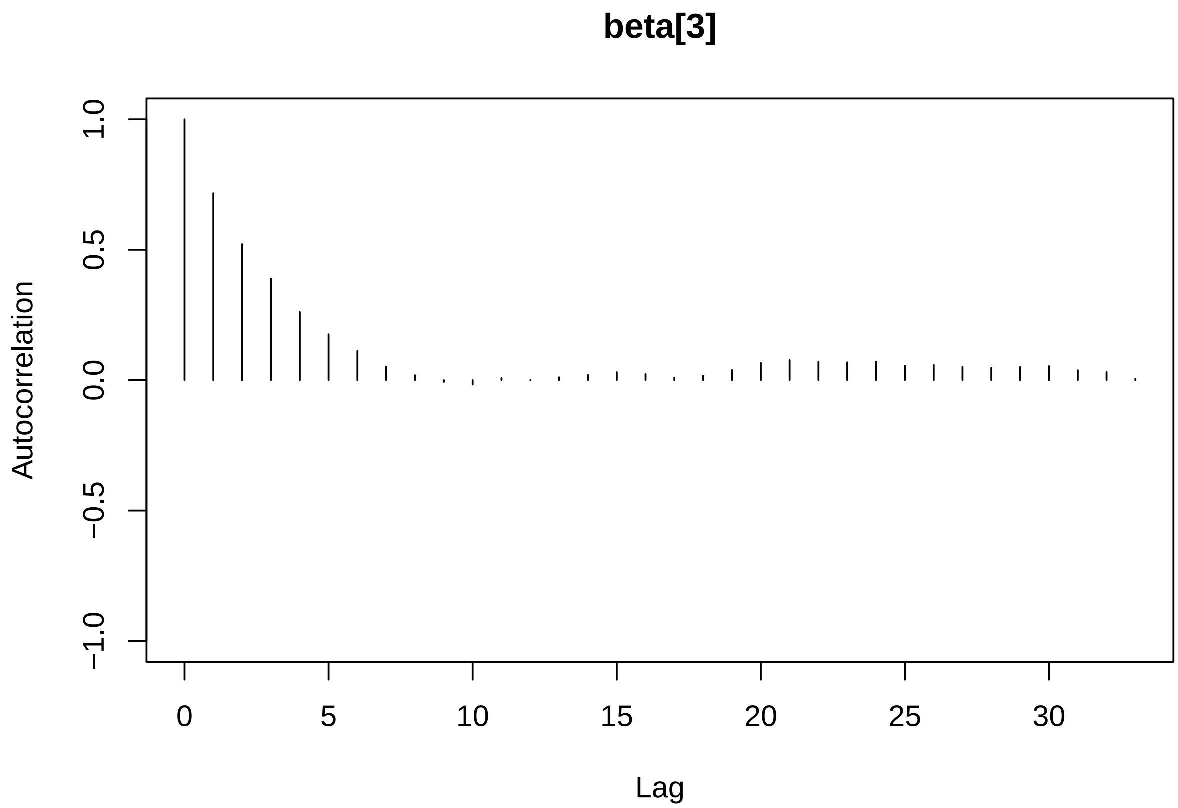


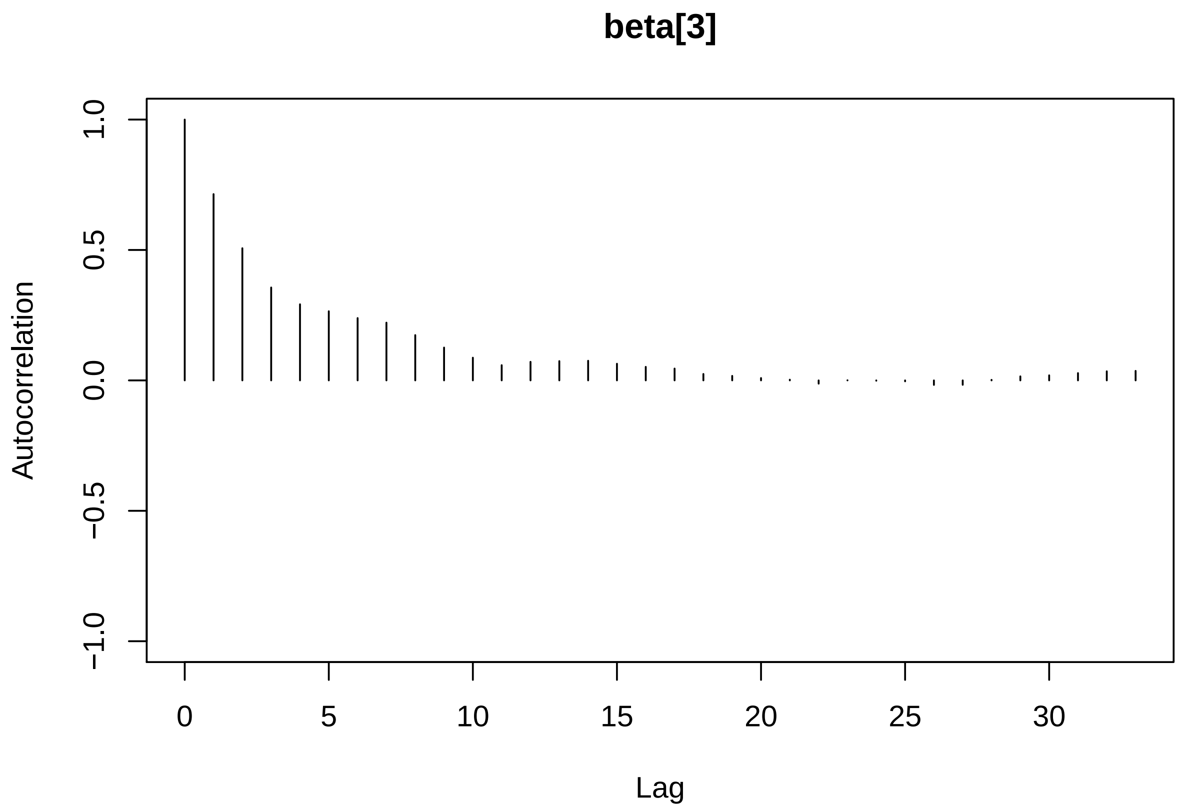
beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

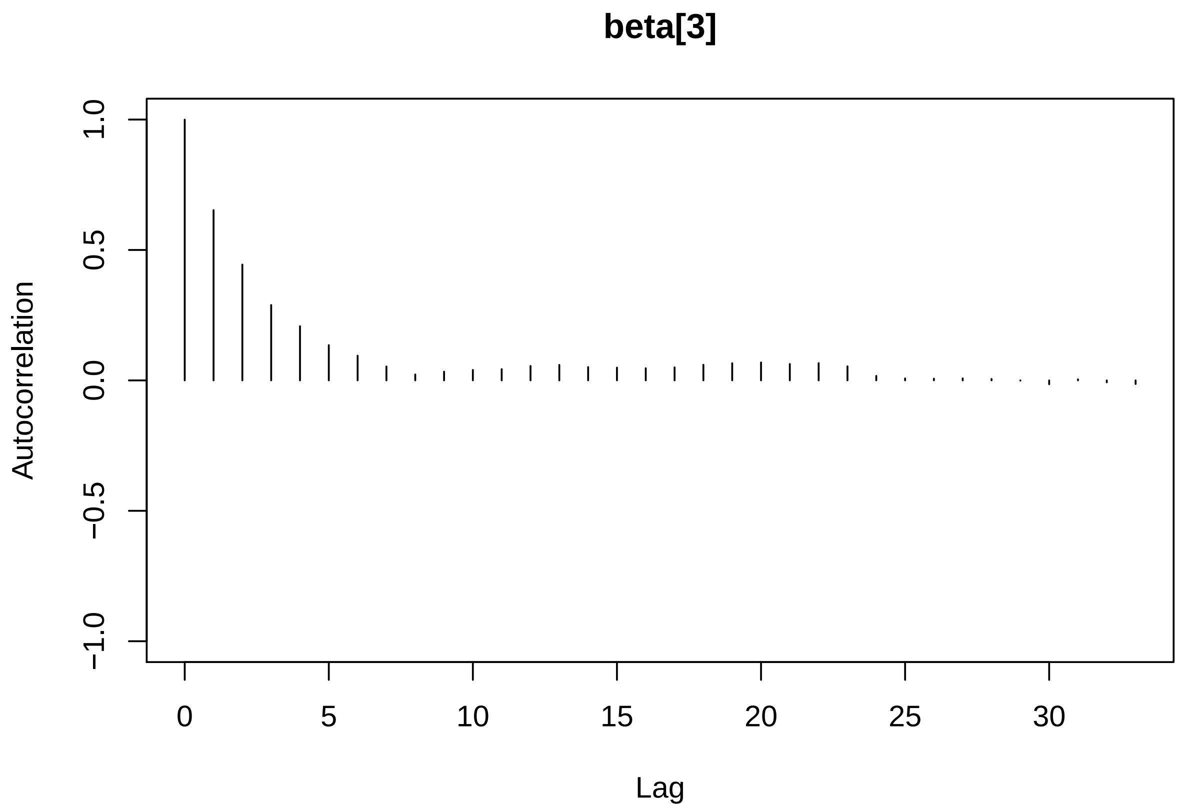

Trace of beta[2]
0.5
0.0
−0.5
−1.0
−1.5
−2.0
11000
12000
13000
Iterations
Density of beta[2]
1.0
0.8
0.6
0.4
0.2
0.0
−2.0
−1.0
0.0
N = 2000 Bandwidth = 0.06371

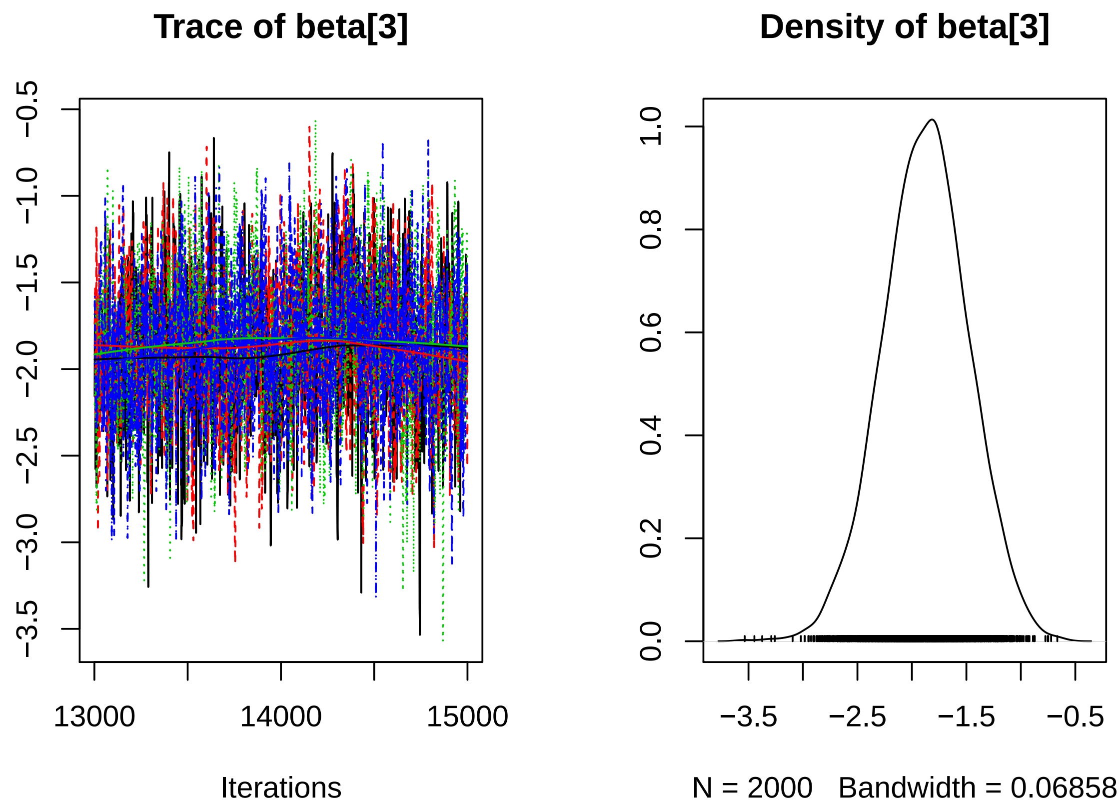


beta[3]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

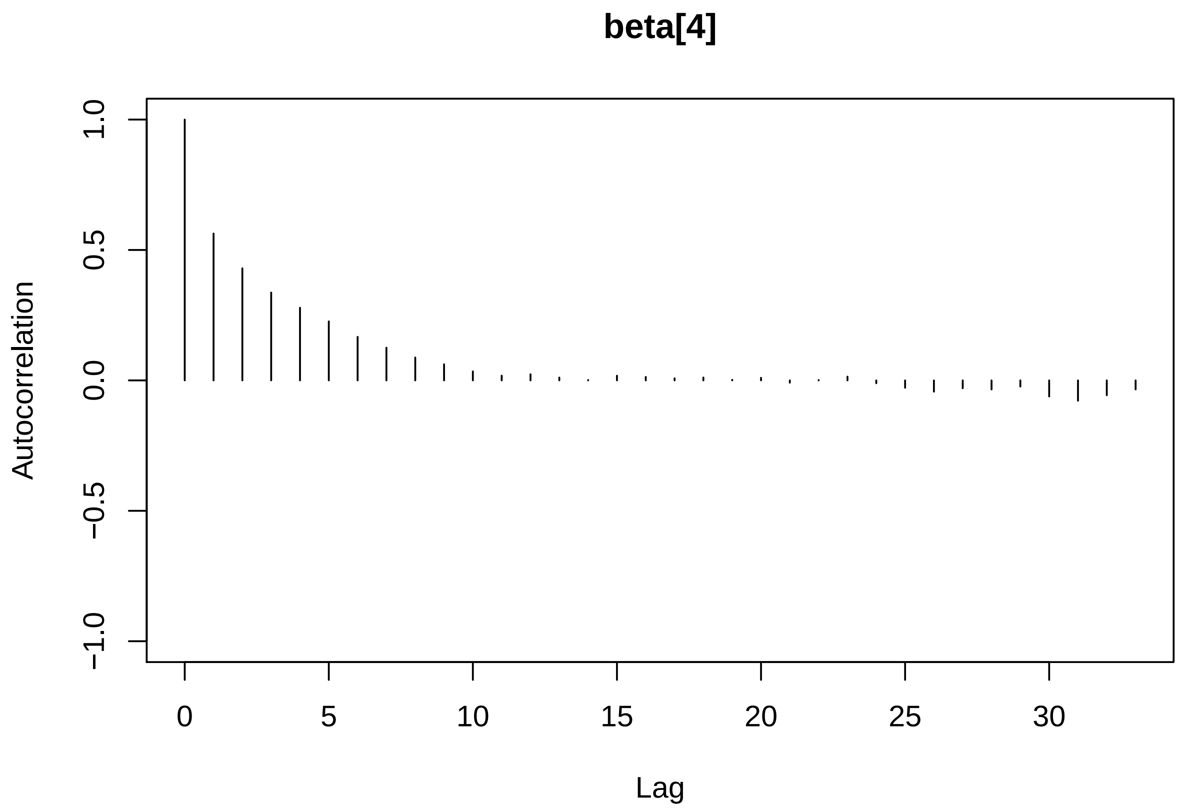

beta[3]
Autocorrelation
Lag
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30

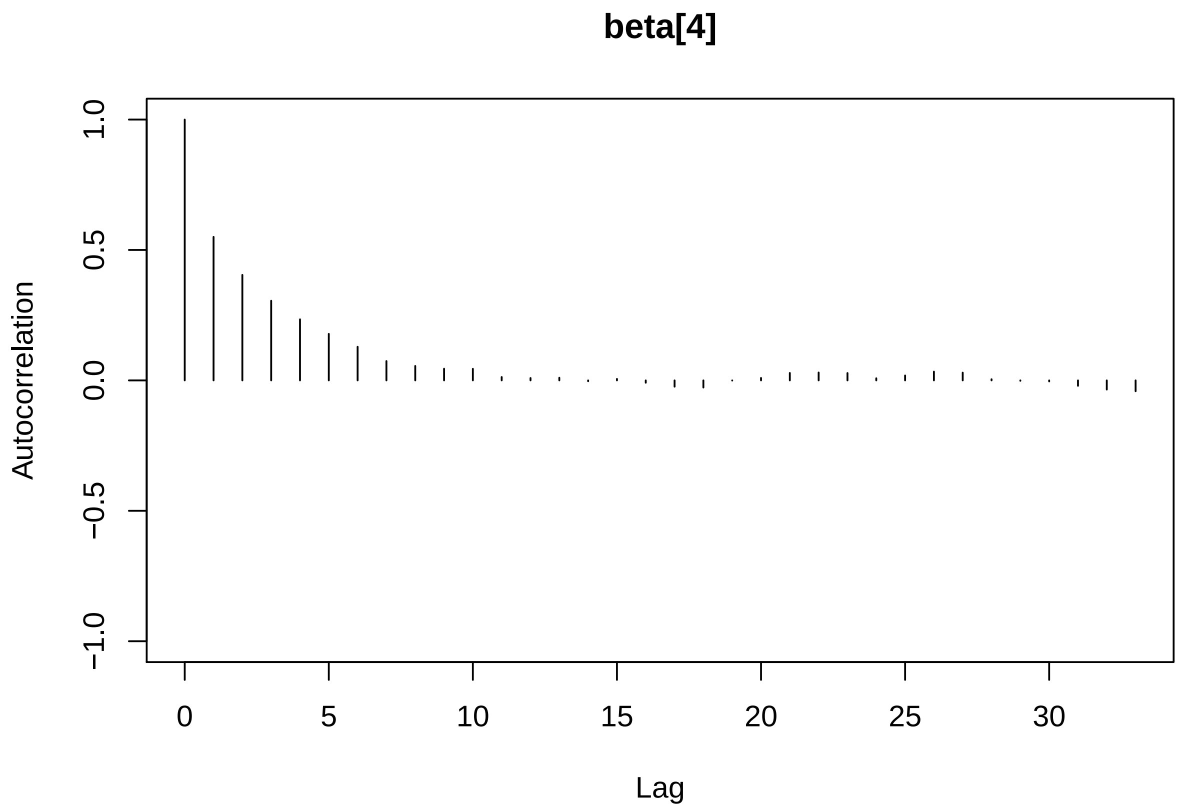

beta[3]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

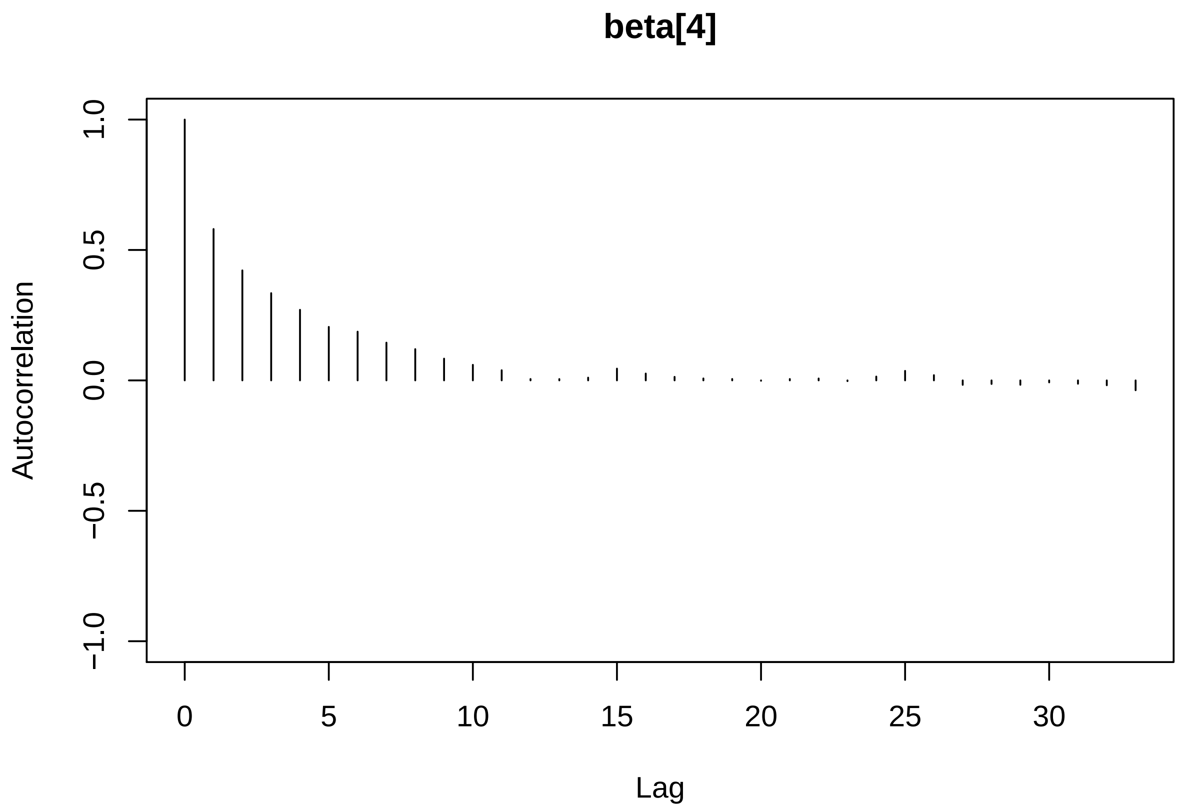

beta[3]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

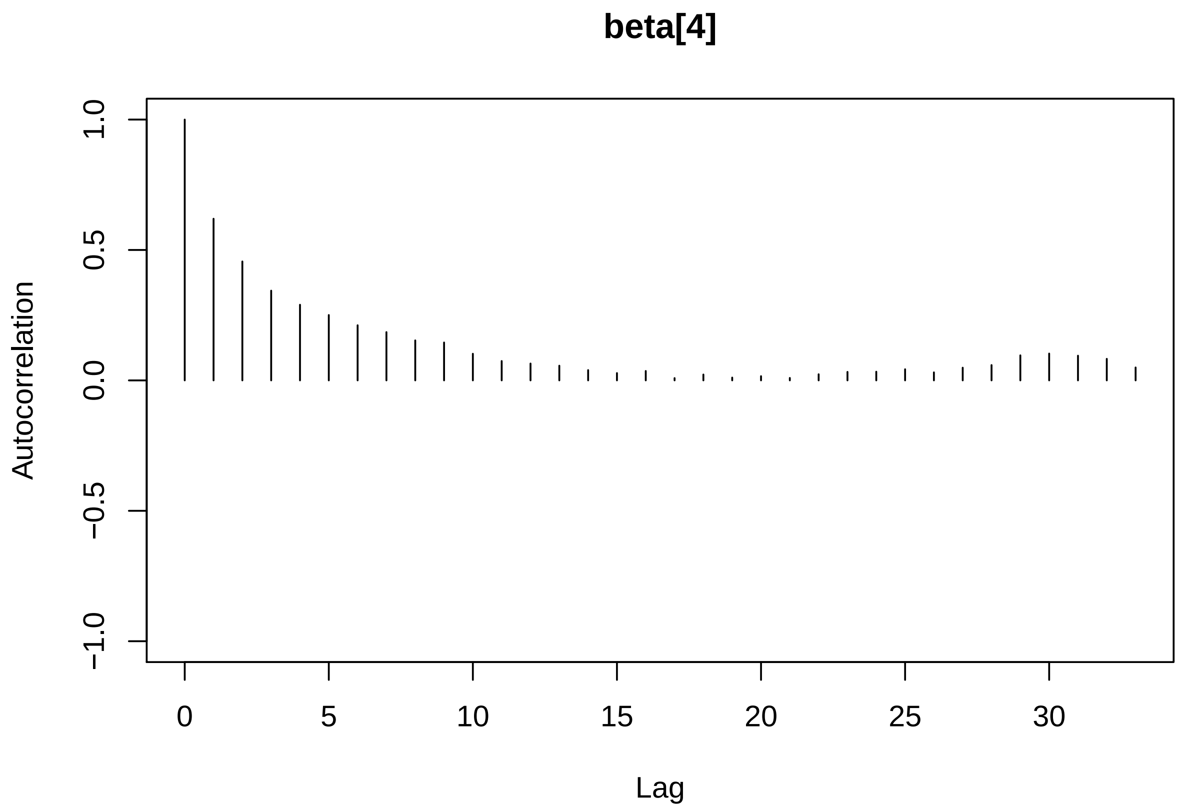


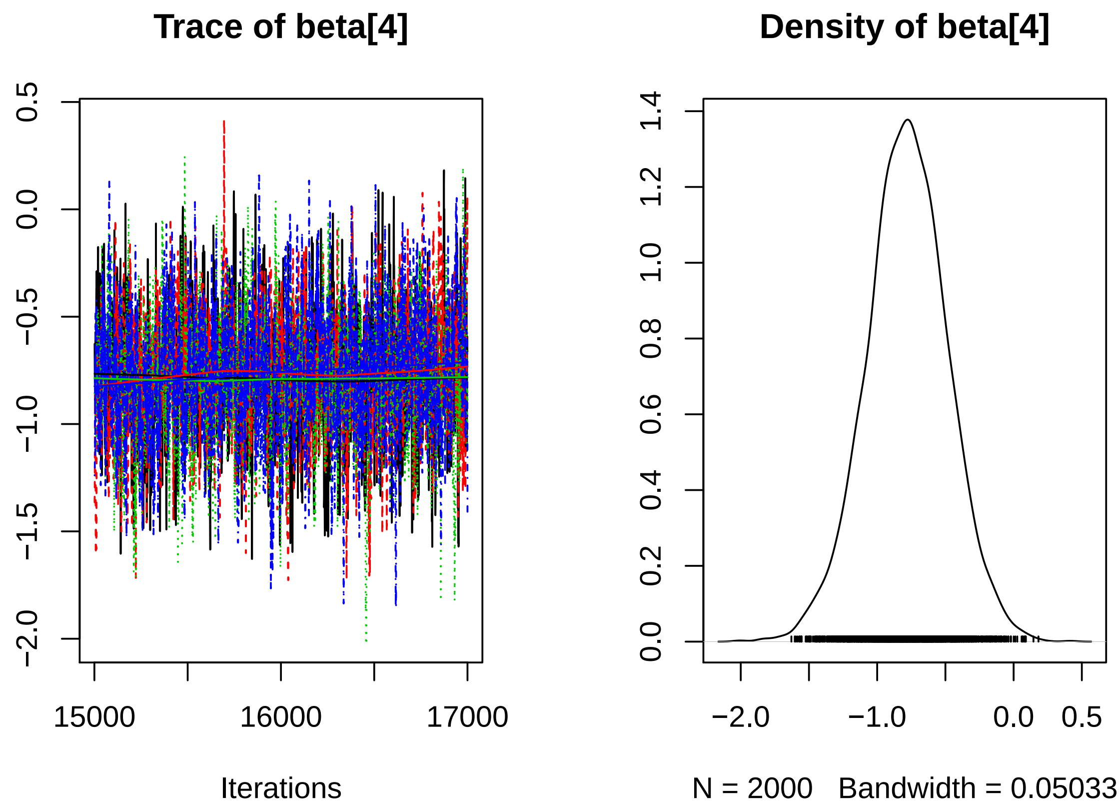
Trace of beta[3]
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
13000
14000
15000
Iterations
Density of beta[3]
1.0
0.8
0.6
0.4
0.2
0.0
−3.5
−2.5
−1.5
−0.5
N = 2000 Bandwidth = 0.06858


beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

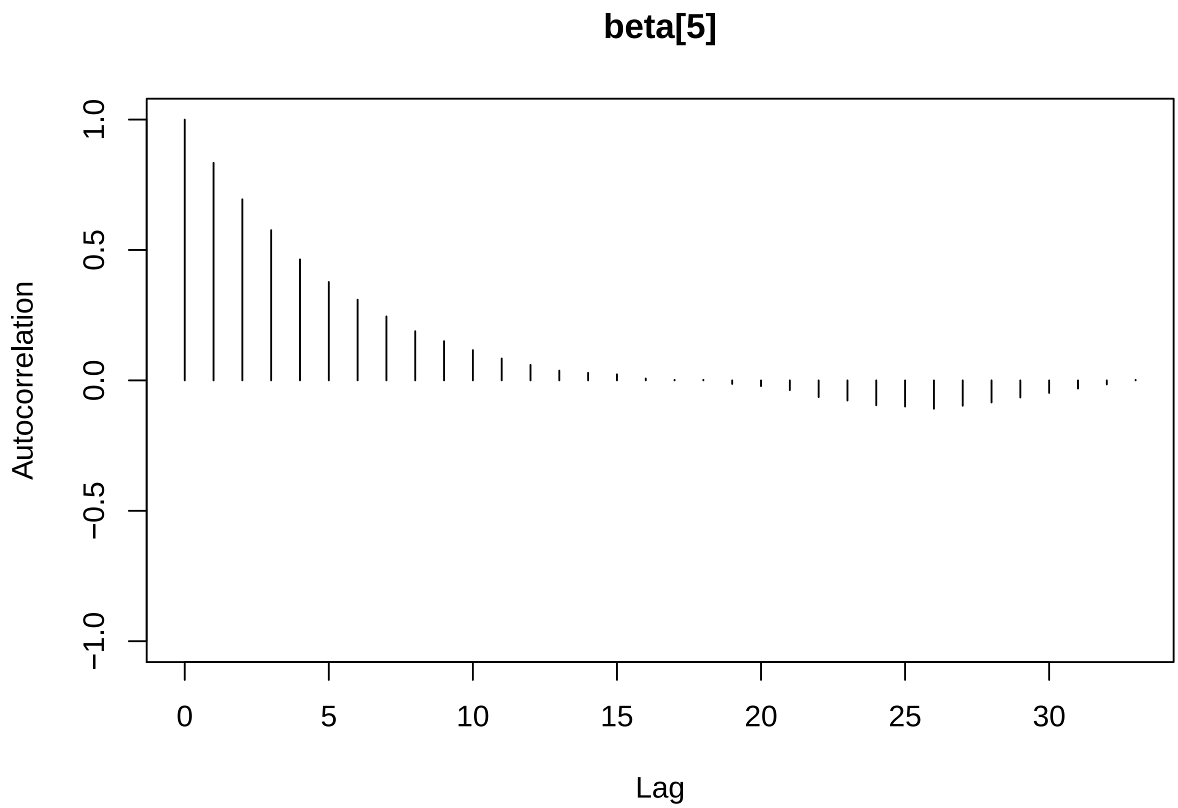

beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

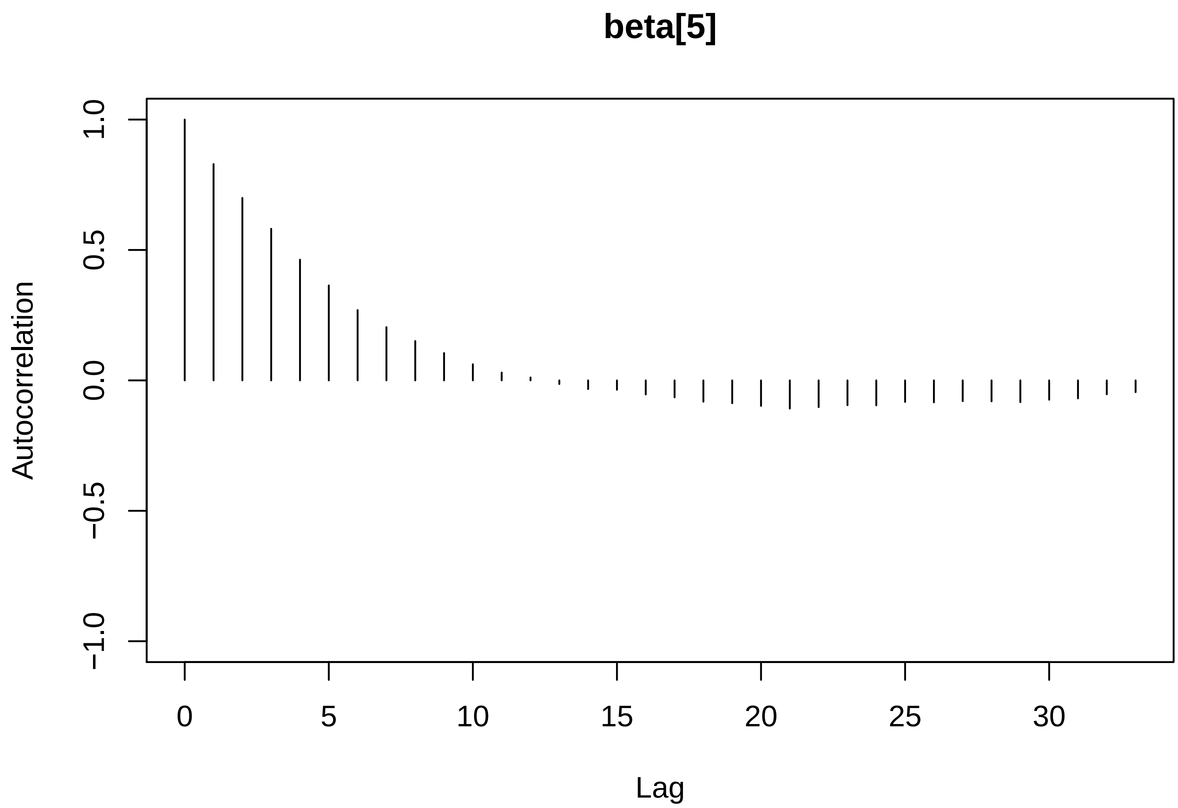

beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

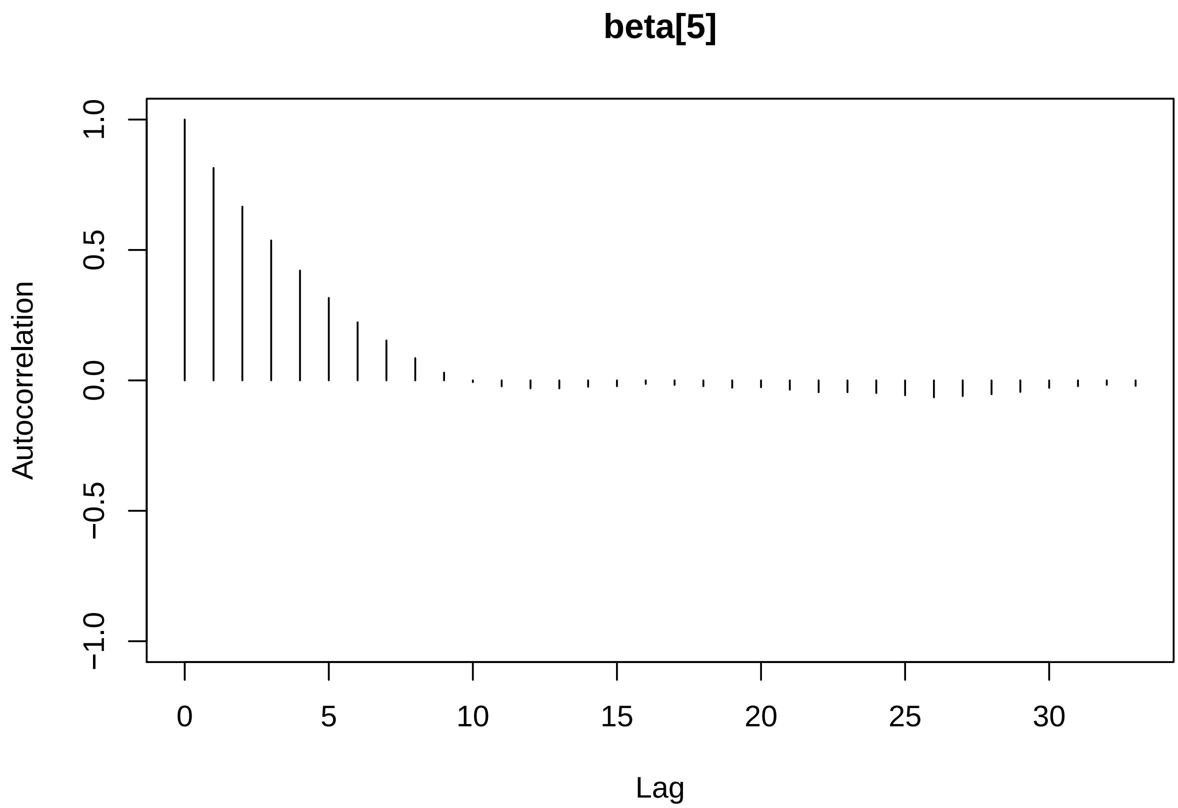

beta[4]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

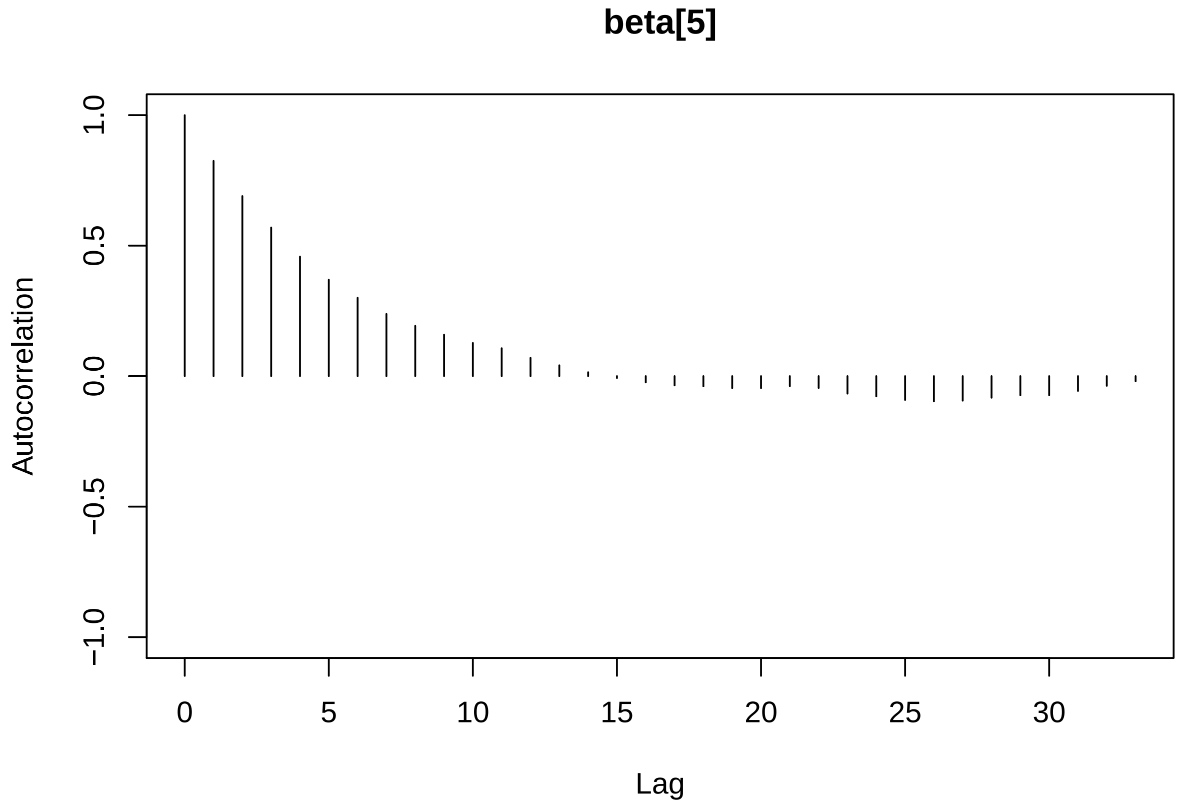


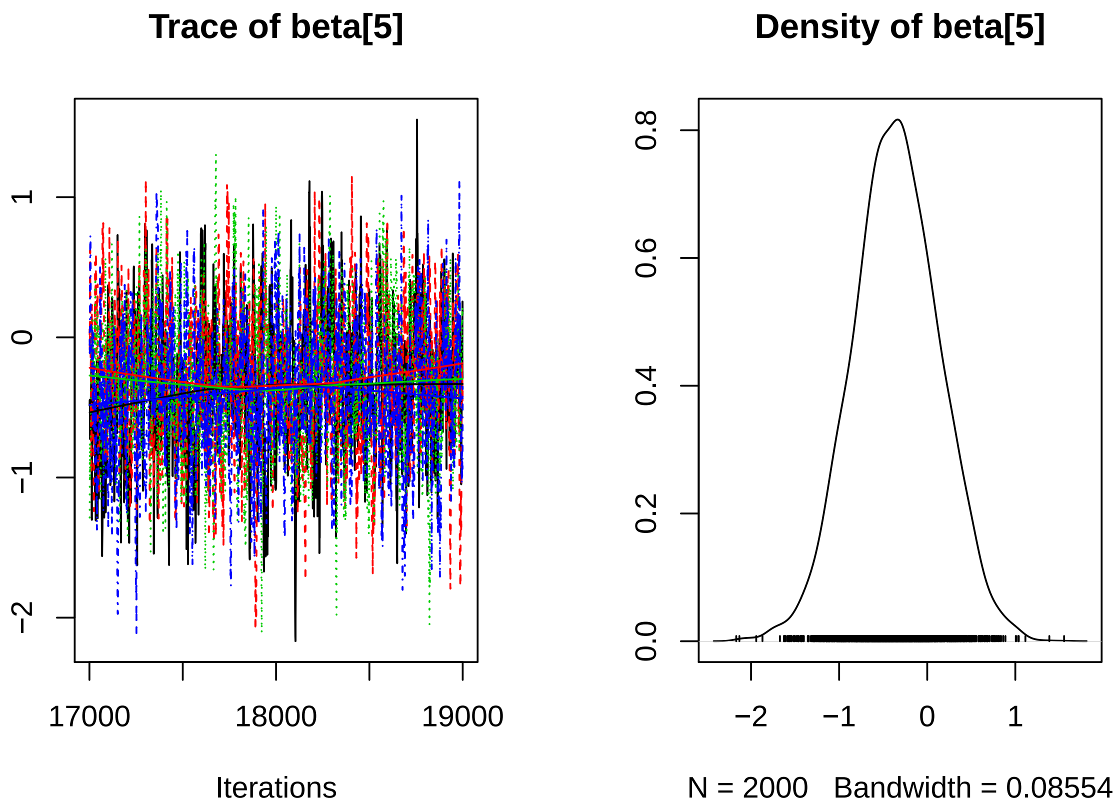
Trace of beta[4]
0.5
0.0
−0.5
−1.0
−1.5
−2.0
15000
16000
17000
Iterations
Density of beta[4]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2.0
−1.0
0.0
0.5
N = 2000 Bandwidth = 0.05033

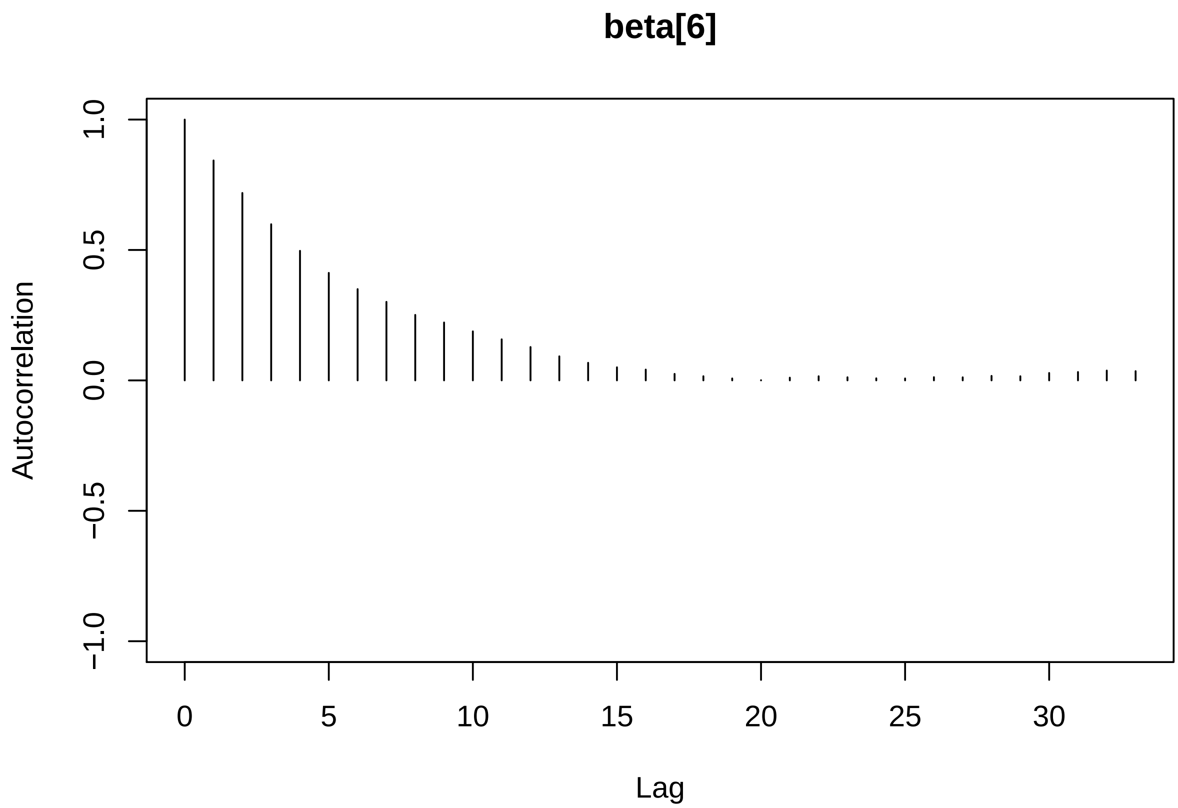

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

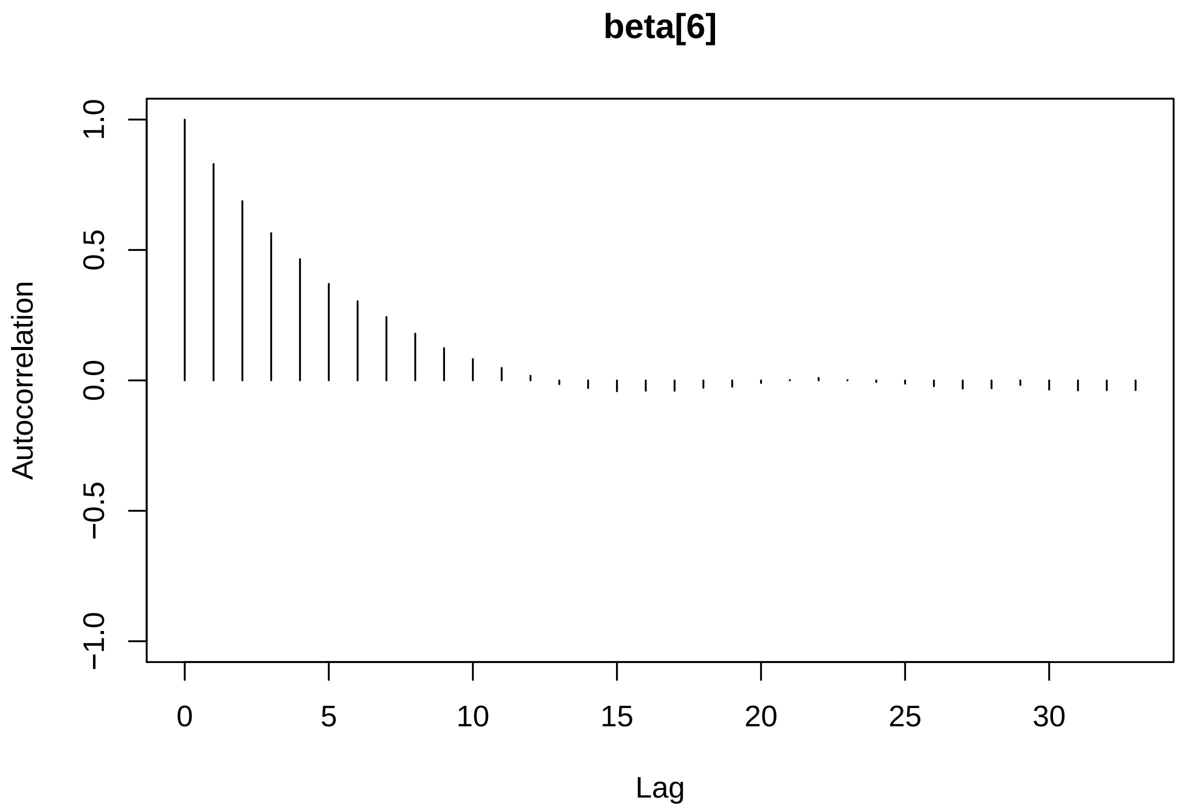

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

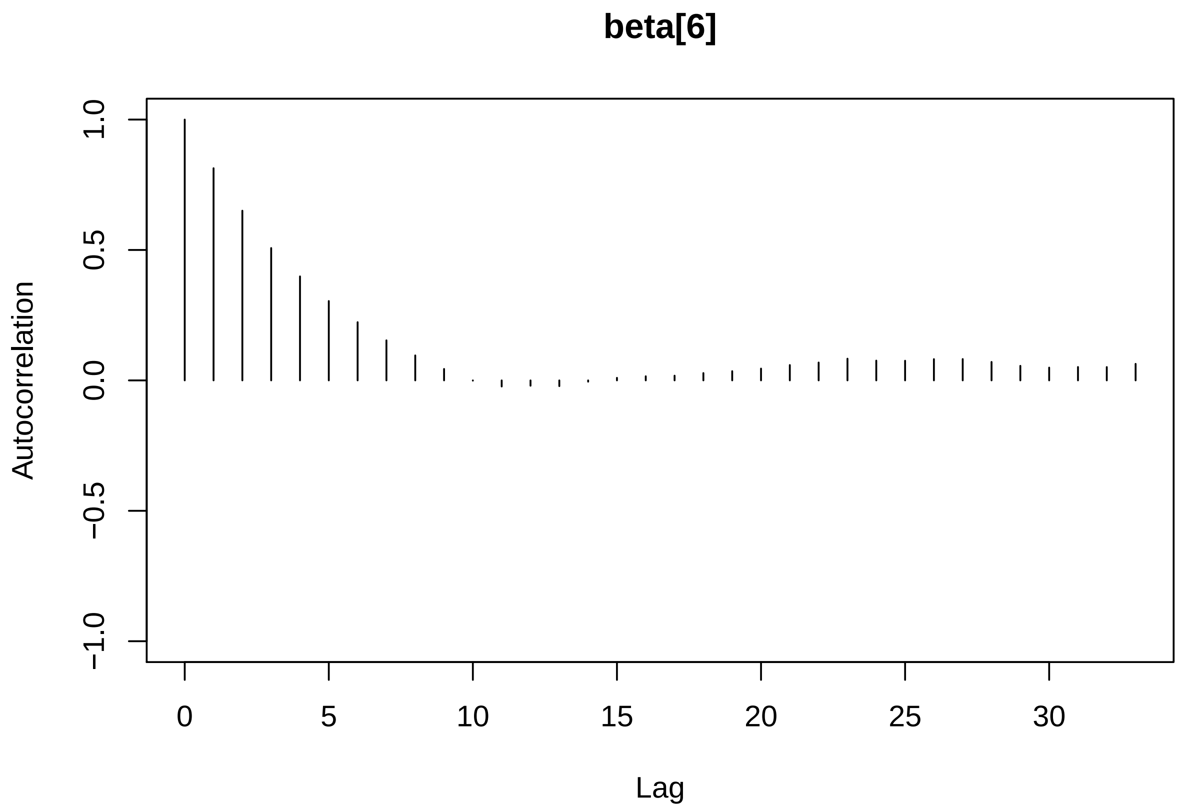

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

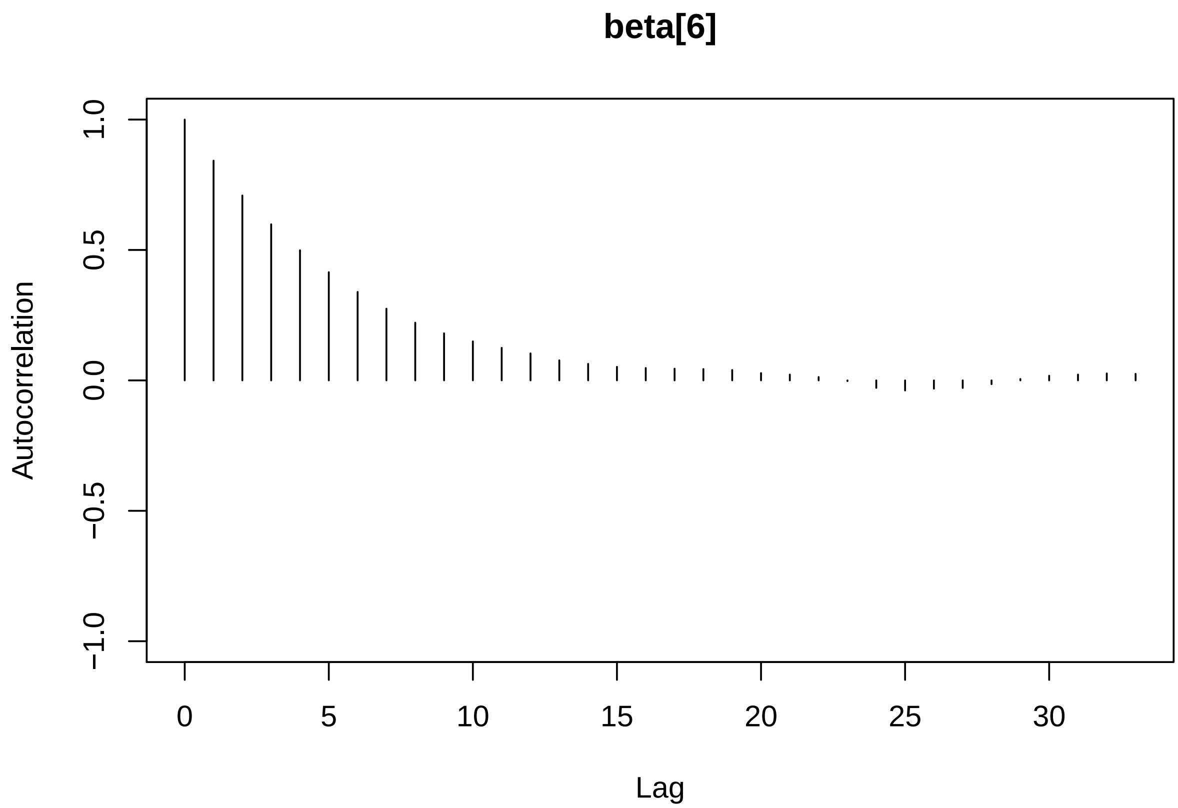

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

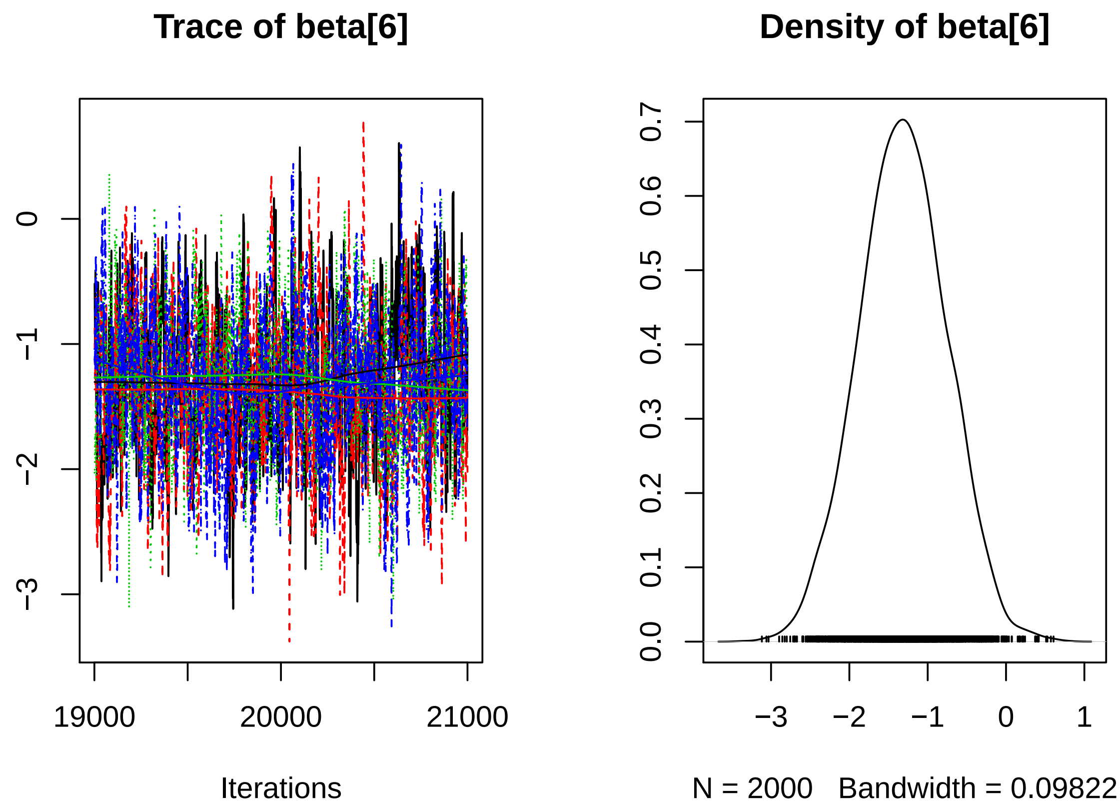

Trace of beta[5]
1
0
−1
−2
17000
18000
19000
Iterations
Density of beta[5]
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
N = 2000   Bandwidth = 0.08554

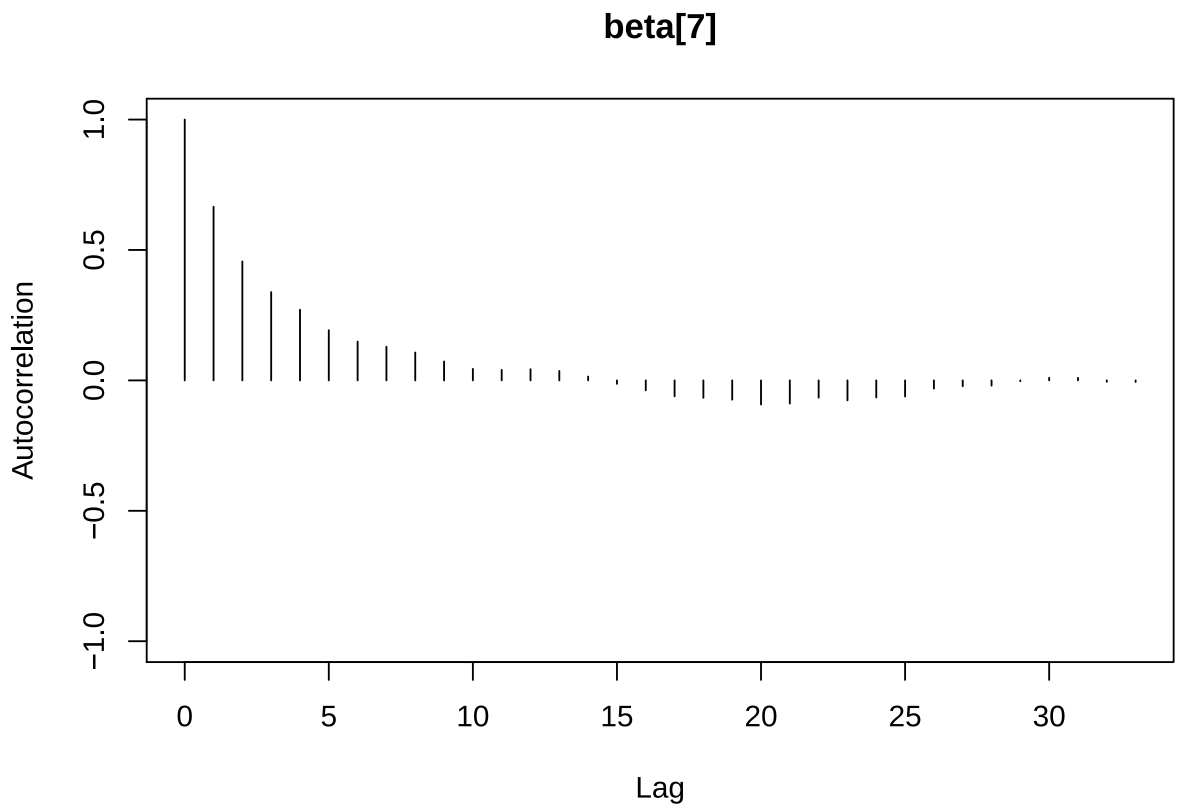

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

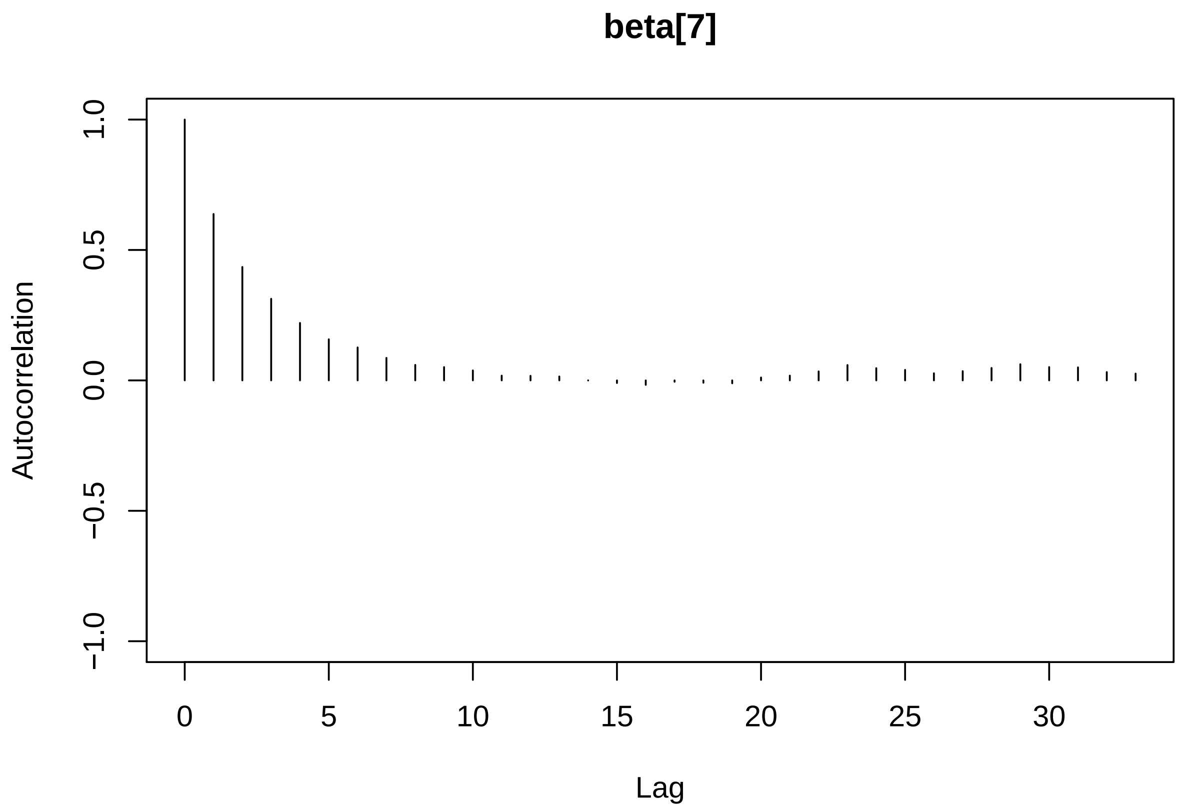

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

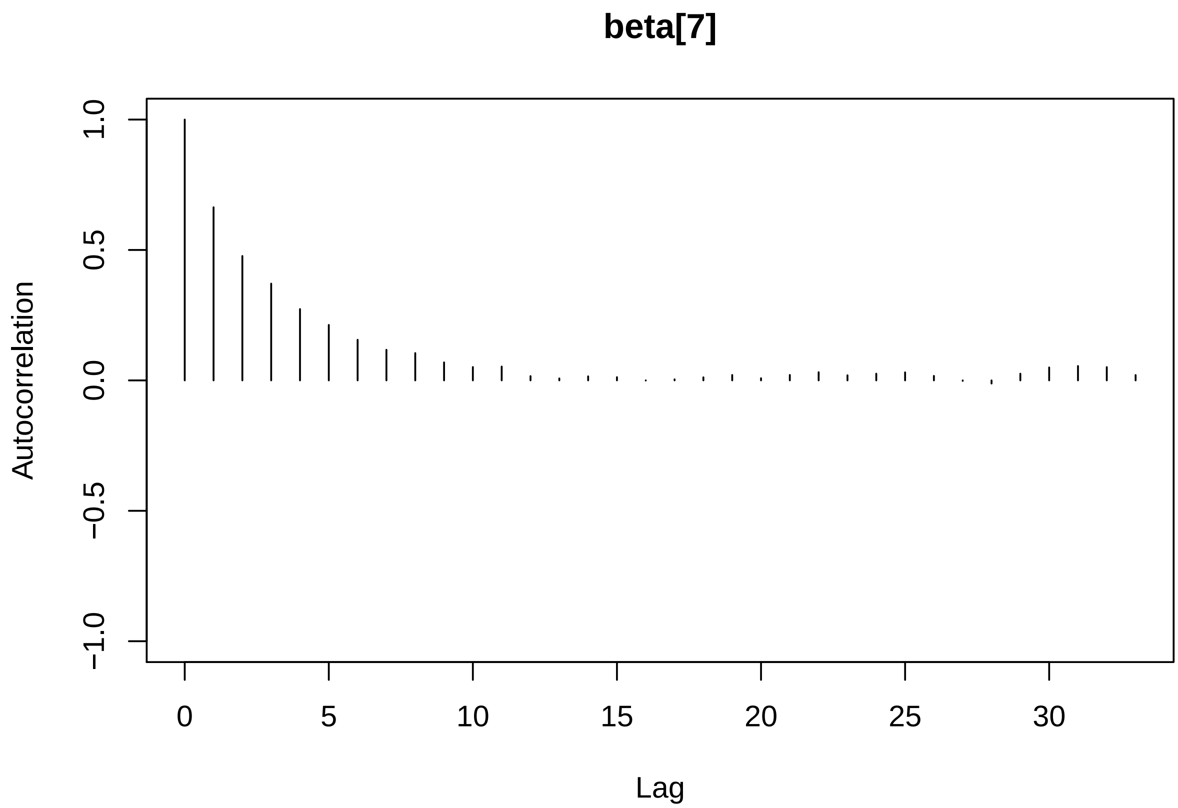

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

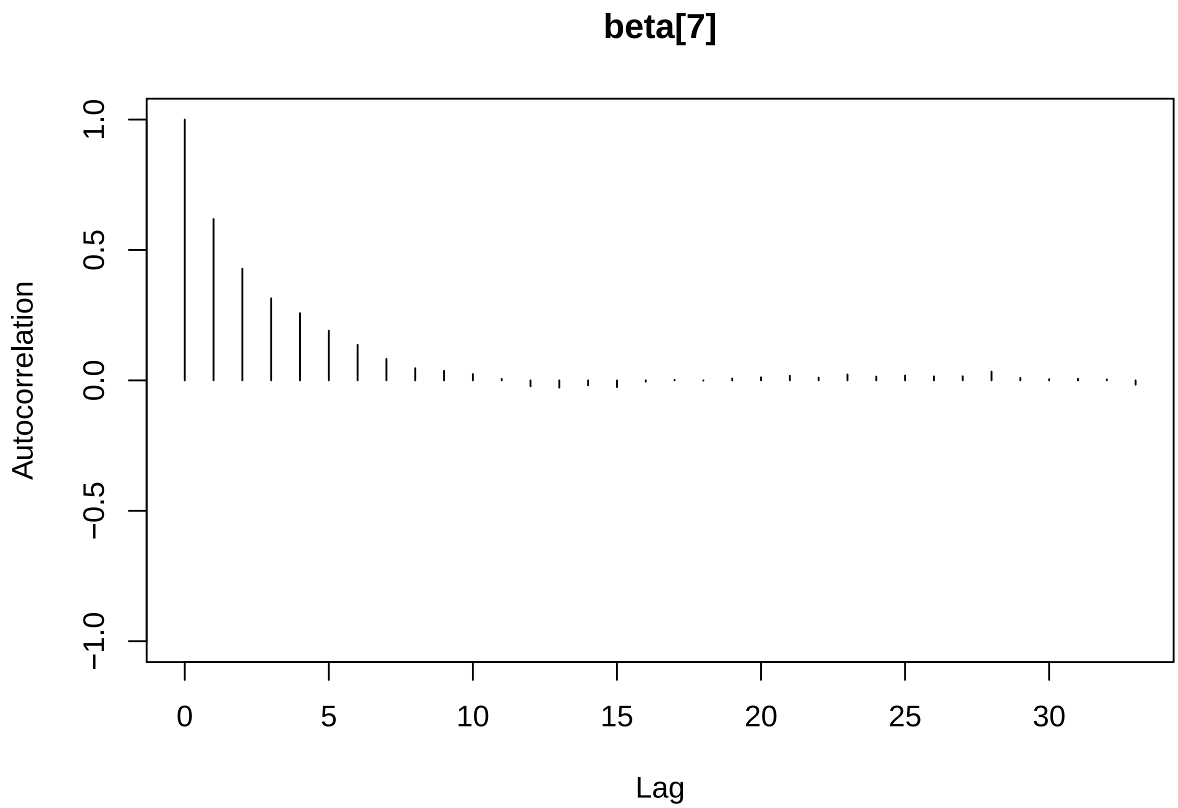

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


Trace of beta[6]
0
−1
−2
−3
19000
20000
21000
Iterations
Density of beta[6]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
−3
−2
−1
0
1
N = 2000 Bandwidth = 0.09822

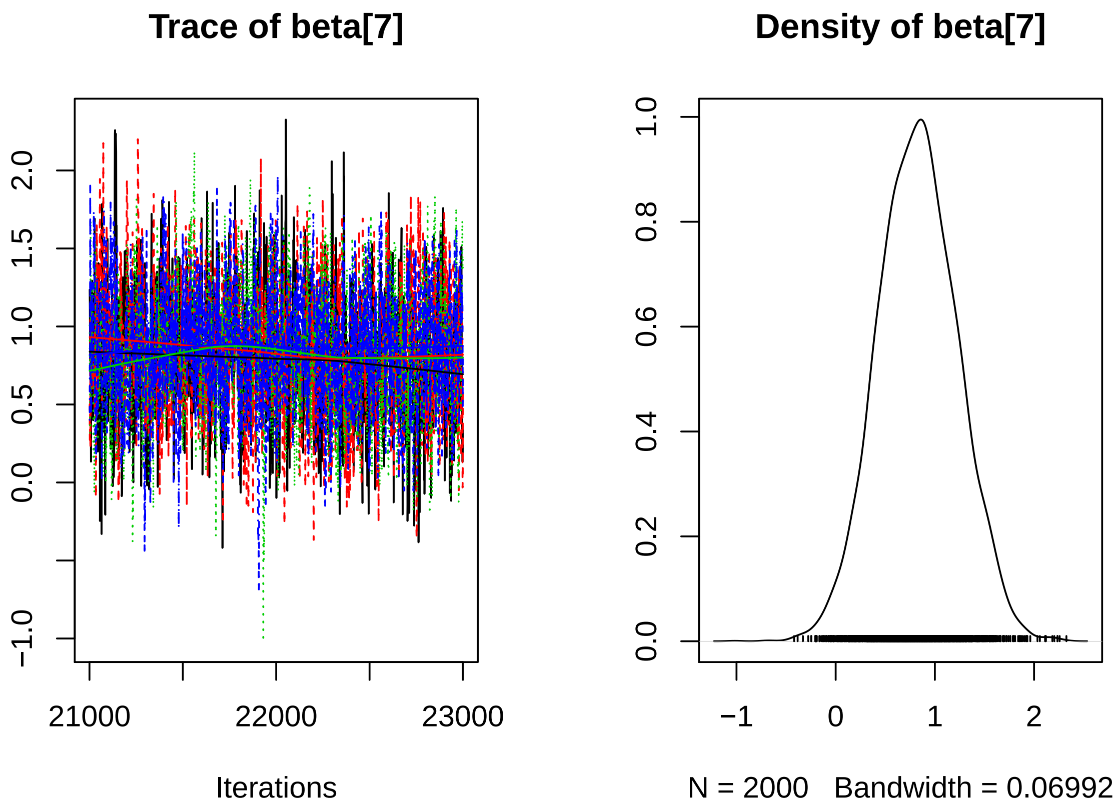

beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

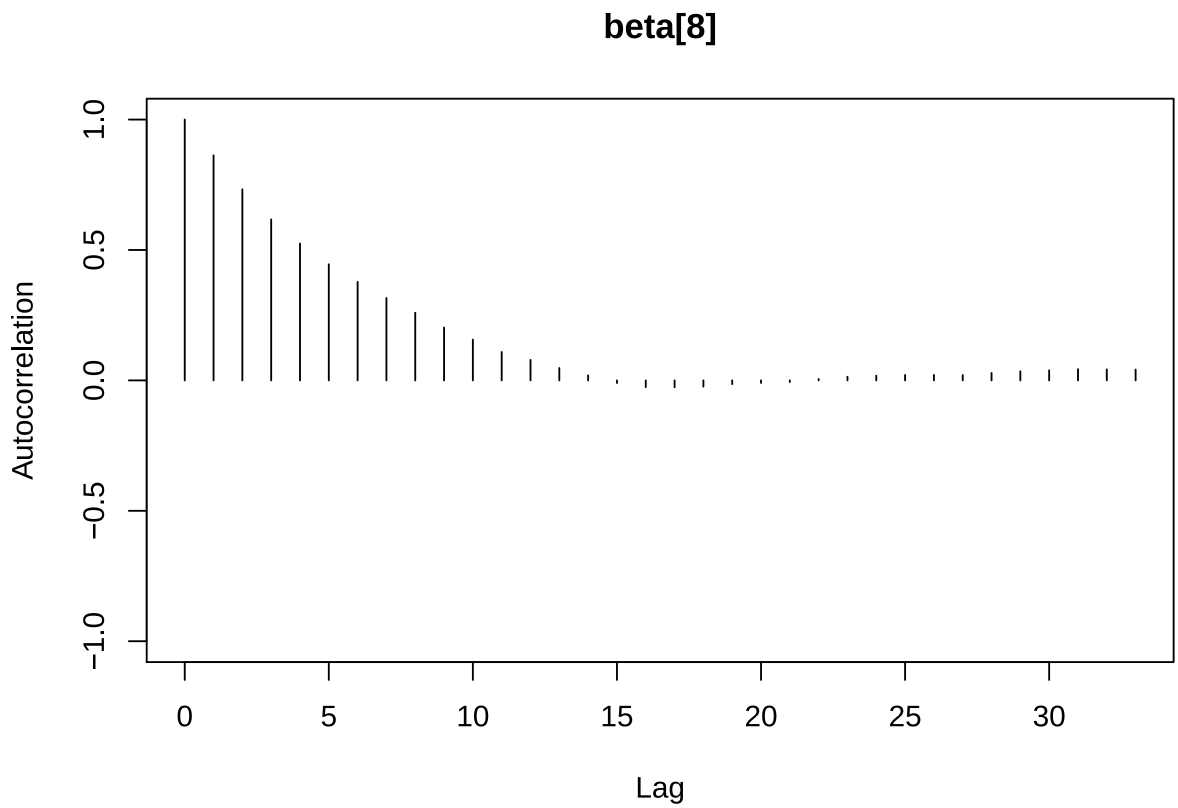

beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[7]
Autocorrelation
Lag
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30

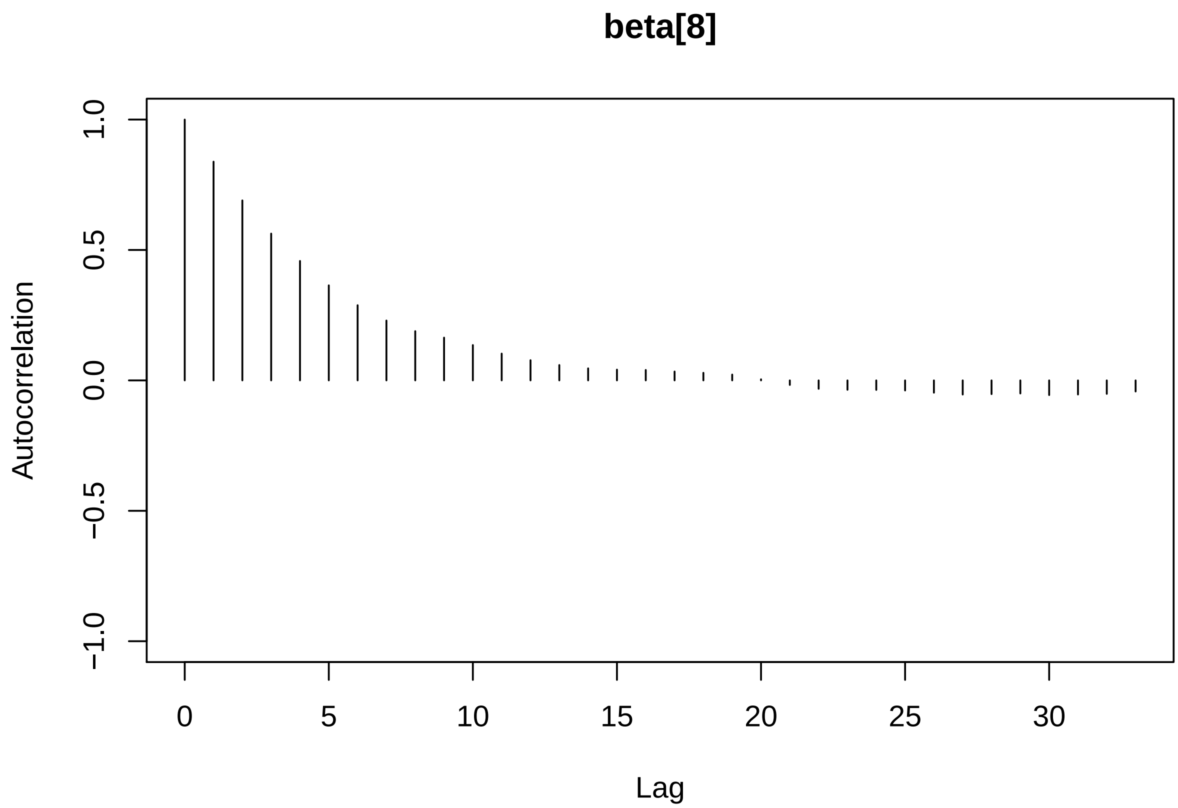


beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

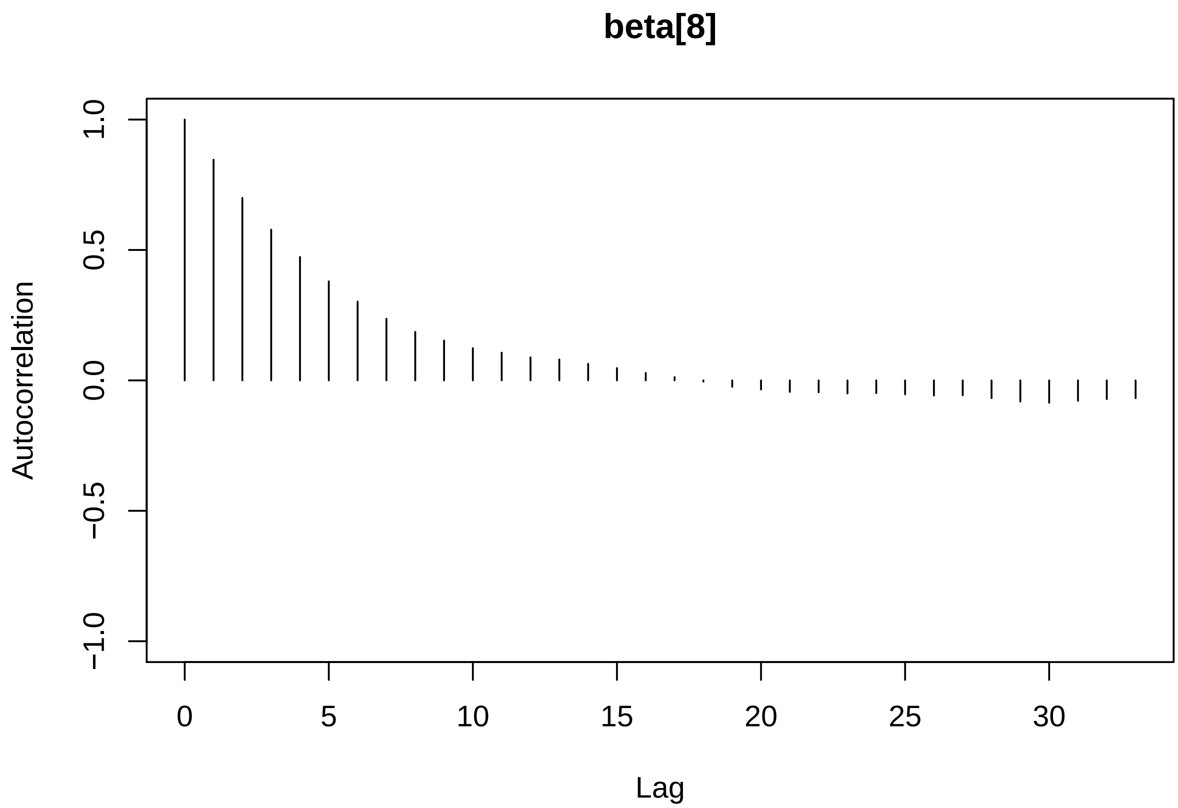

Trace of beta[7]
Iterations
Density of beta[7]
N = 2000 Bandwidth = 0.06992

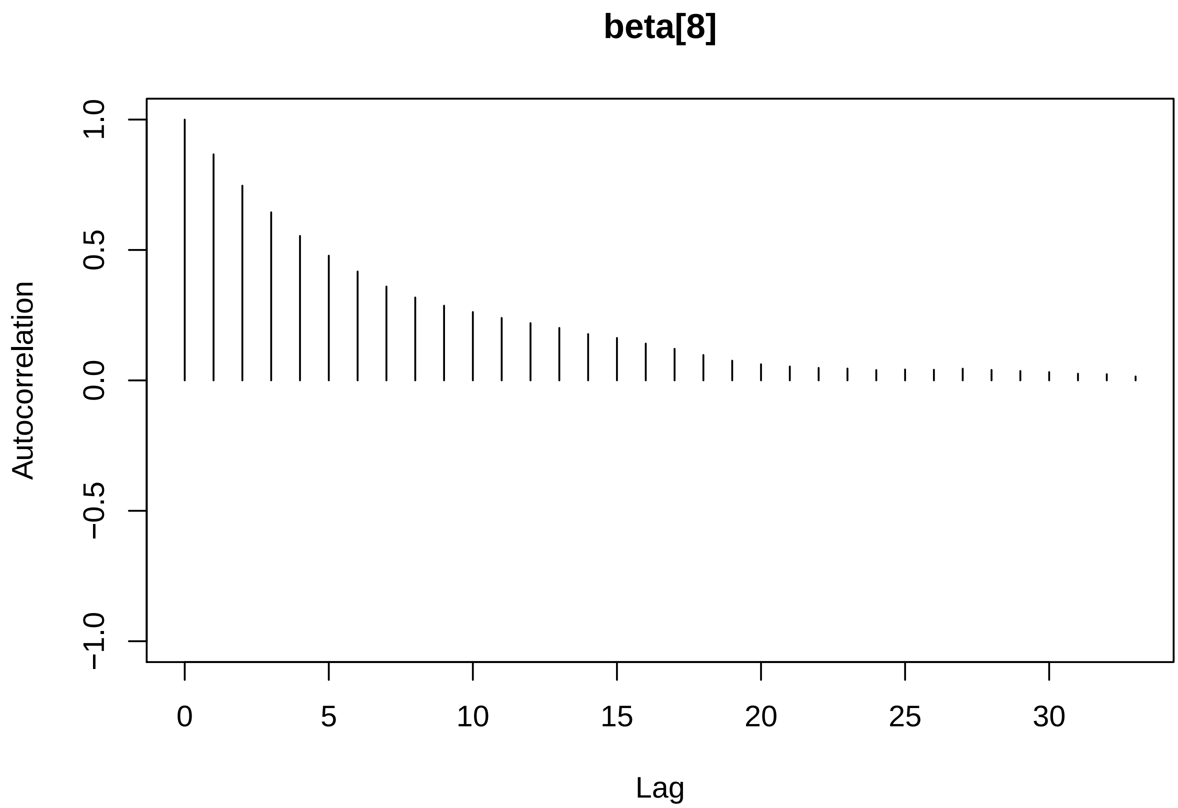


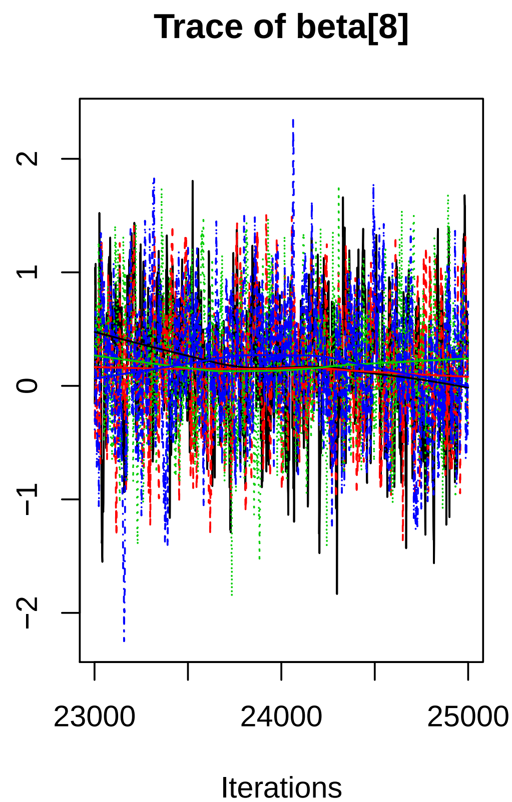
beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

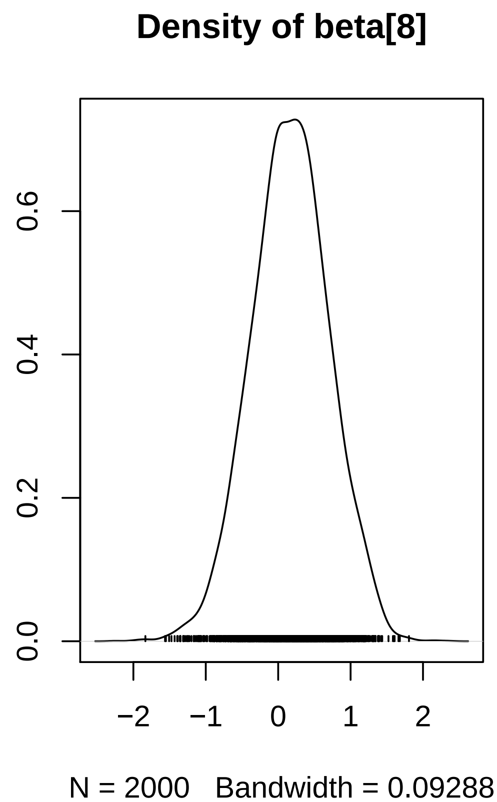


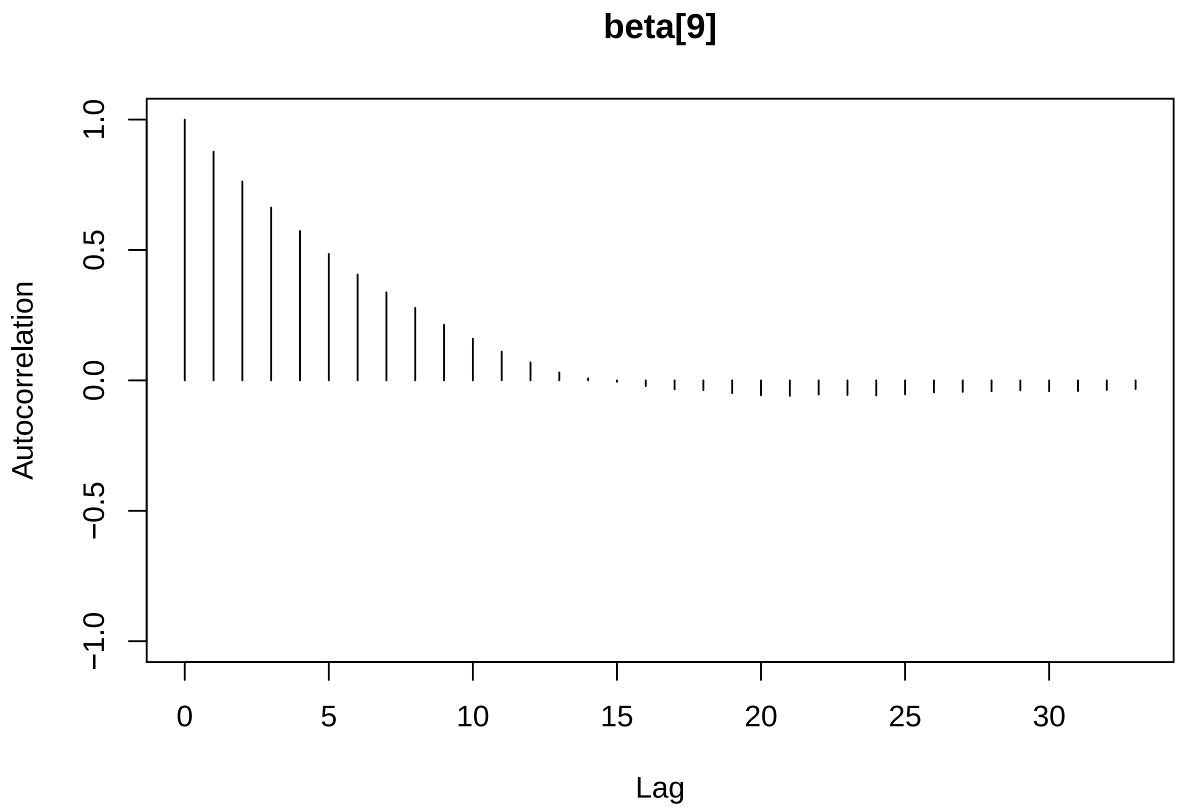
beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

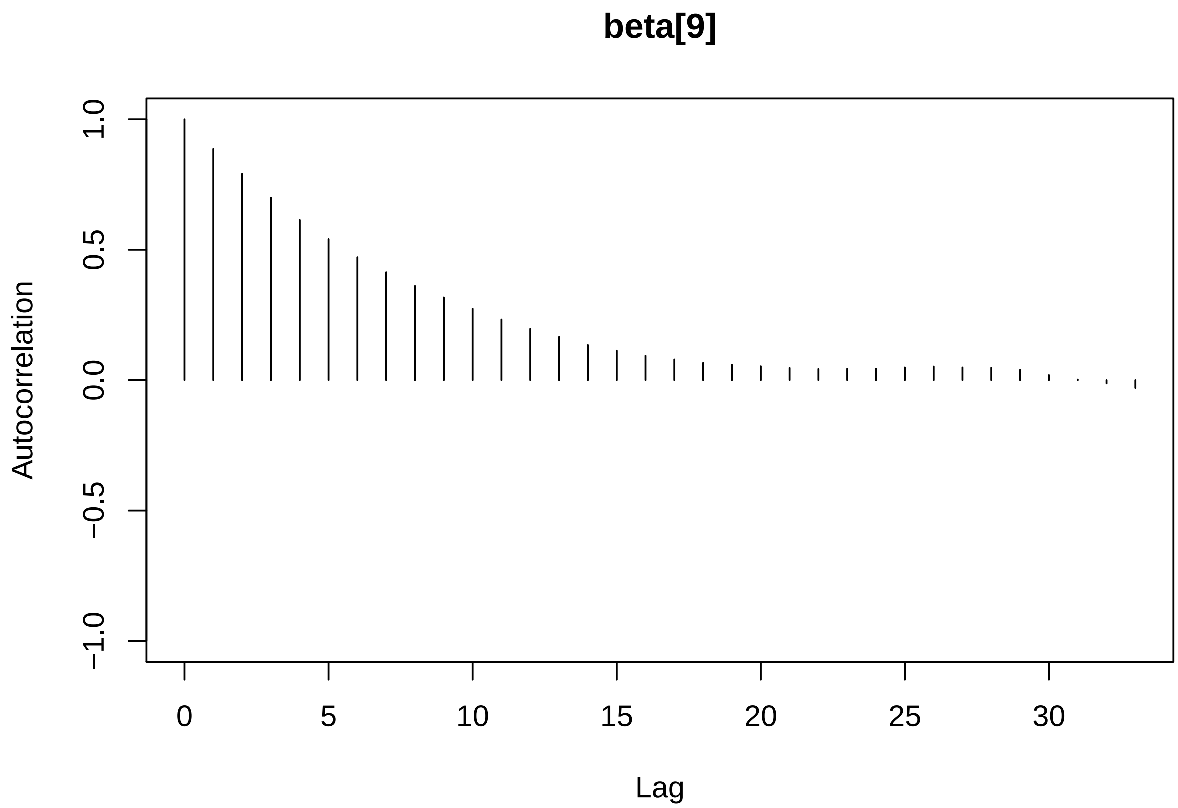

beta[8]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

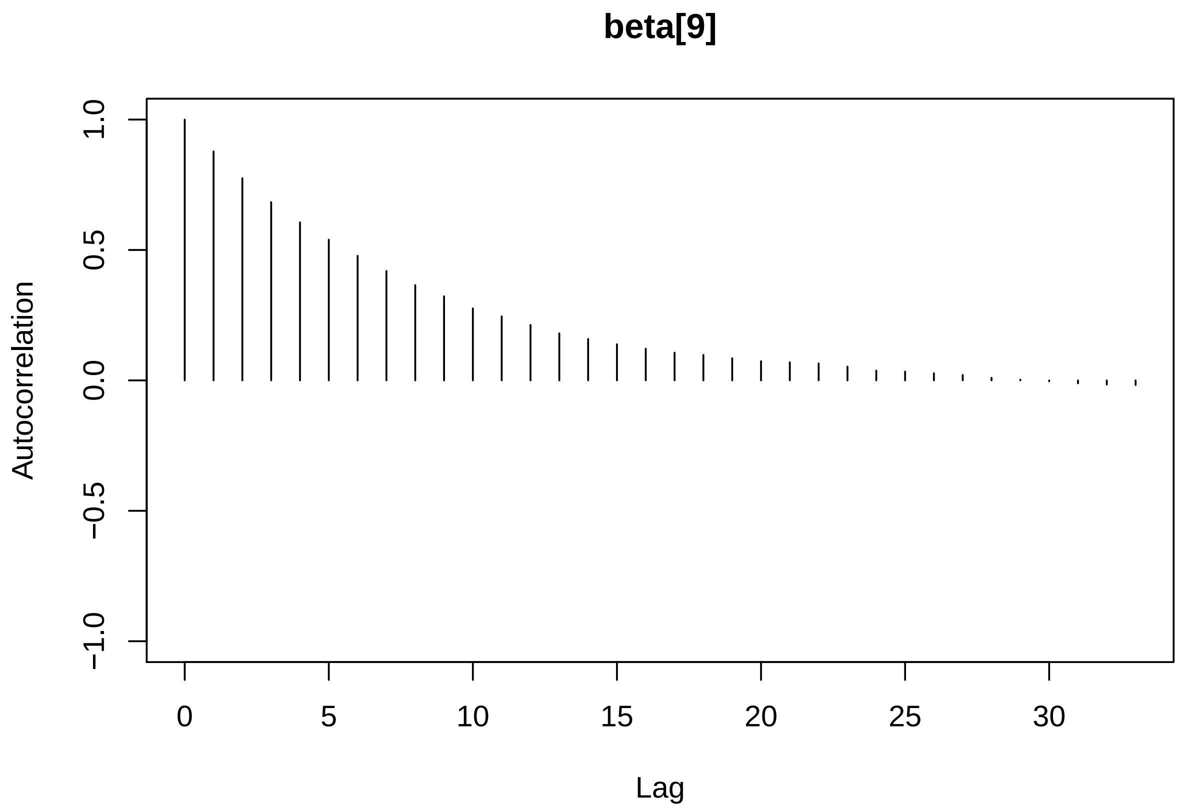

beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


Trace of beta[8]
2
1
0
−1
−2
23000
24000
25000
Iterations

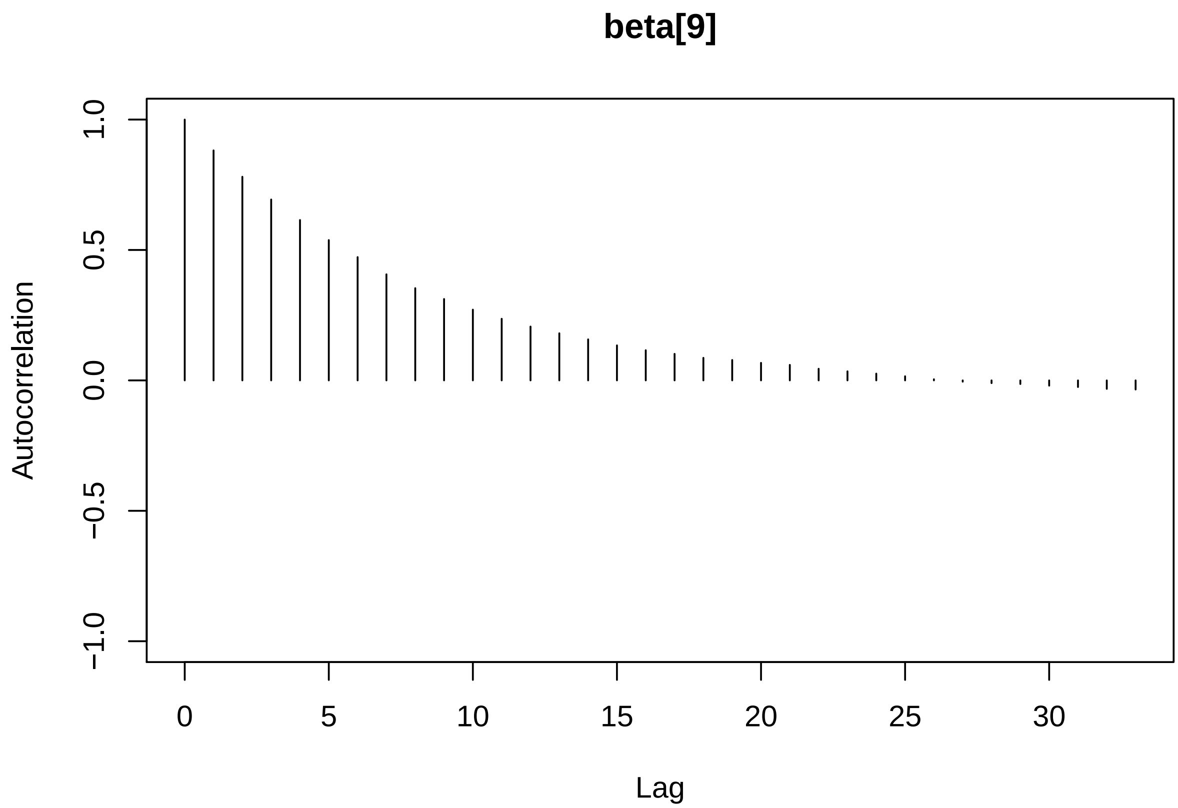

Density of beta[8]
0.6
0.4
0.2
0.0
−2
−1
0
1
2
N = 2000 Bandwidth = 0.09288

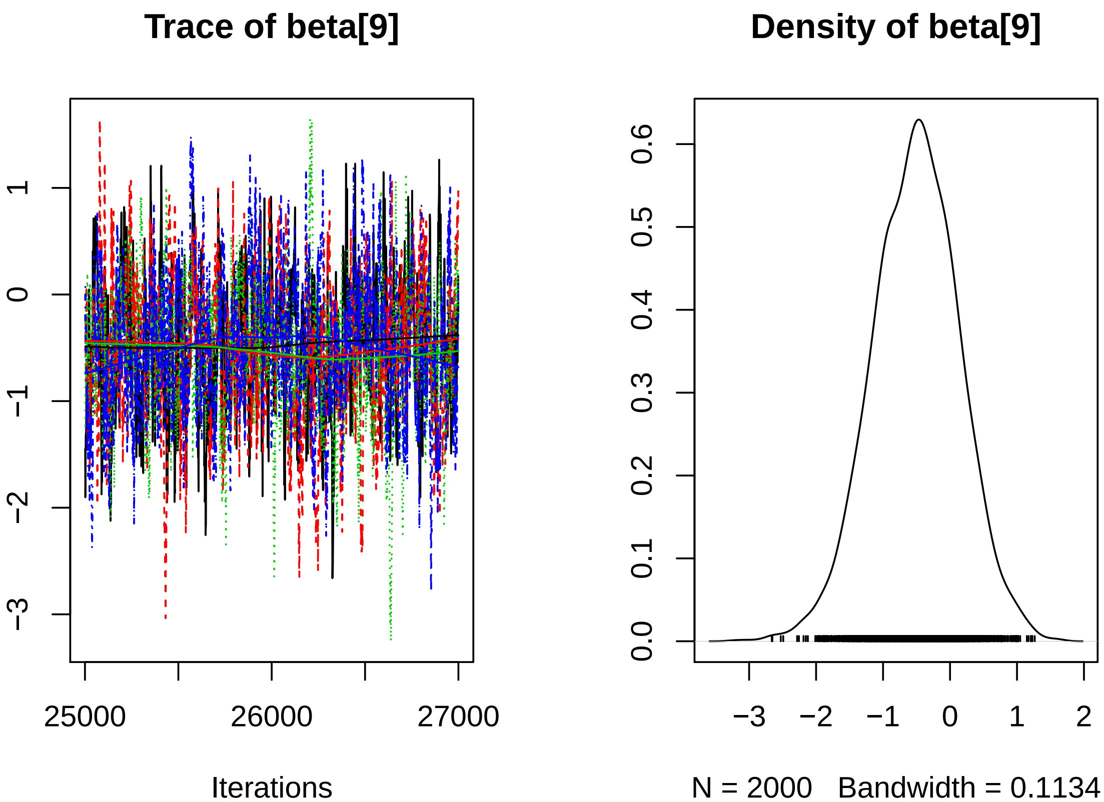

beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

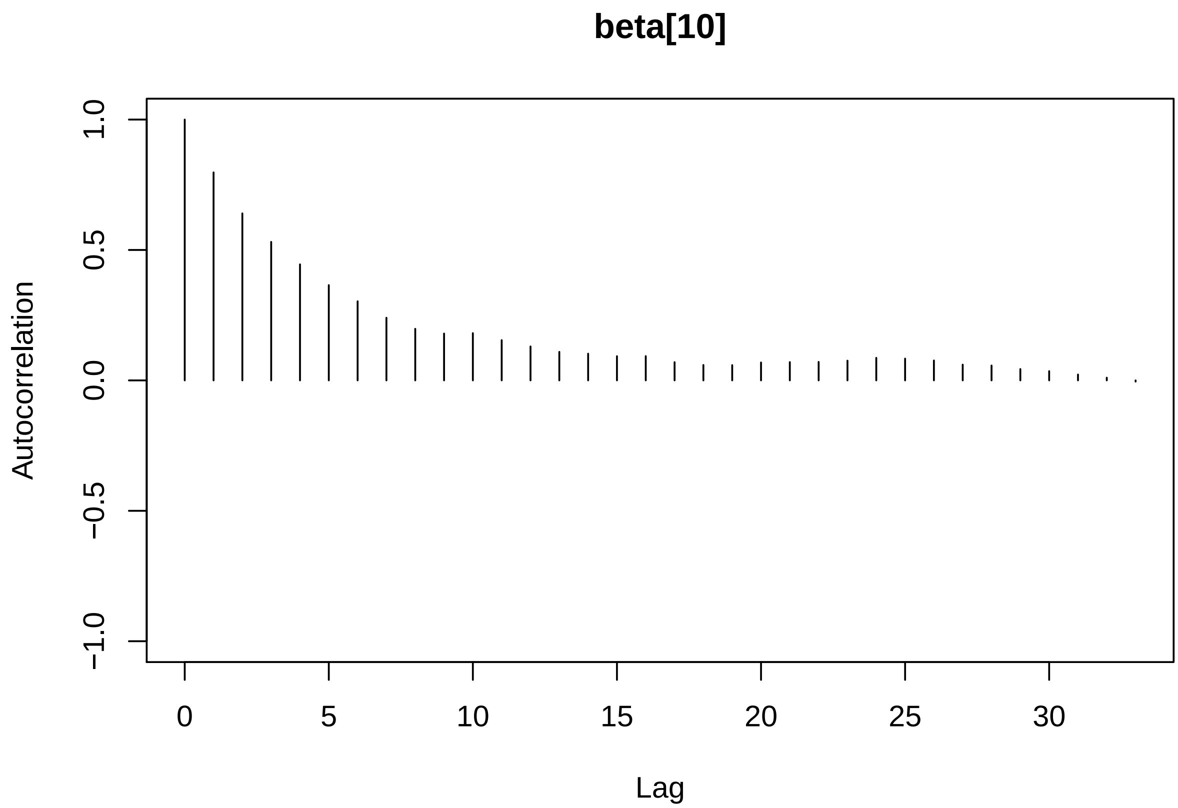

beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

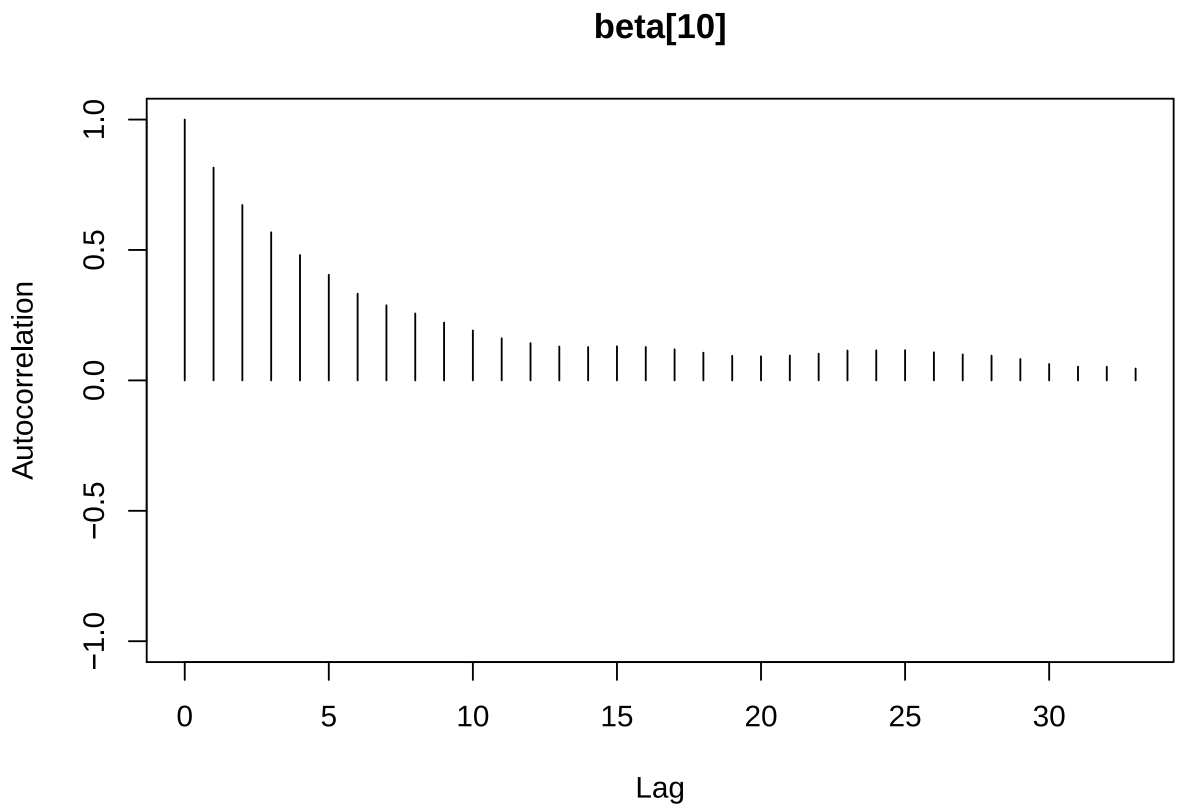


beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

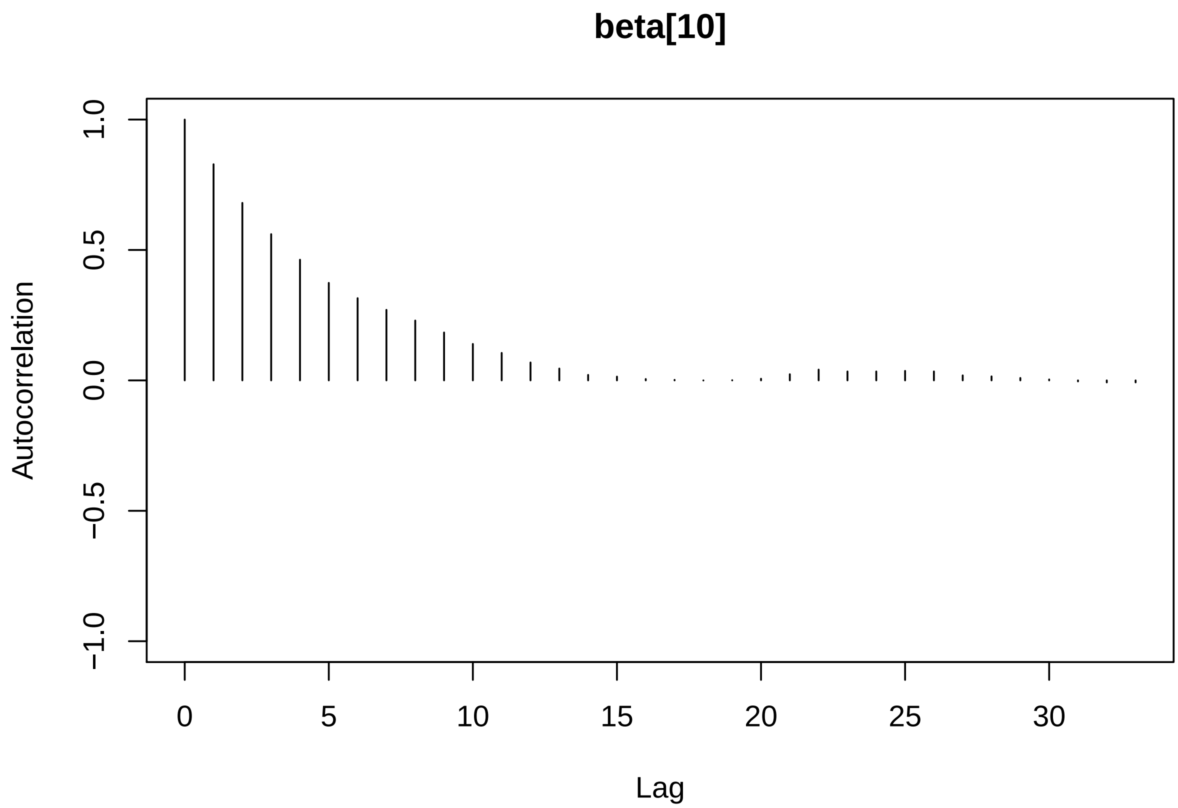

beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

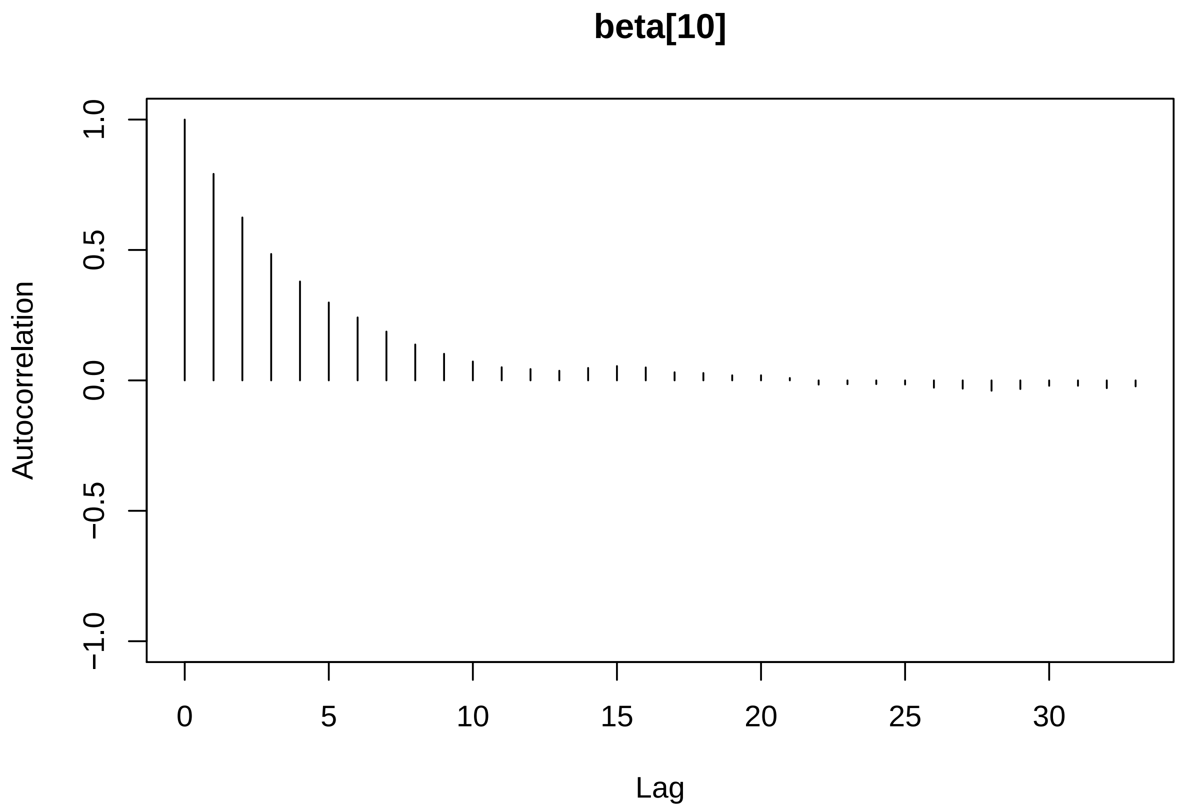

Trace of beta[9]
Iterations
Density of beta[9]
N = 2000 Bandwidth = 0.1134


beta[10]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

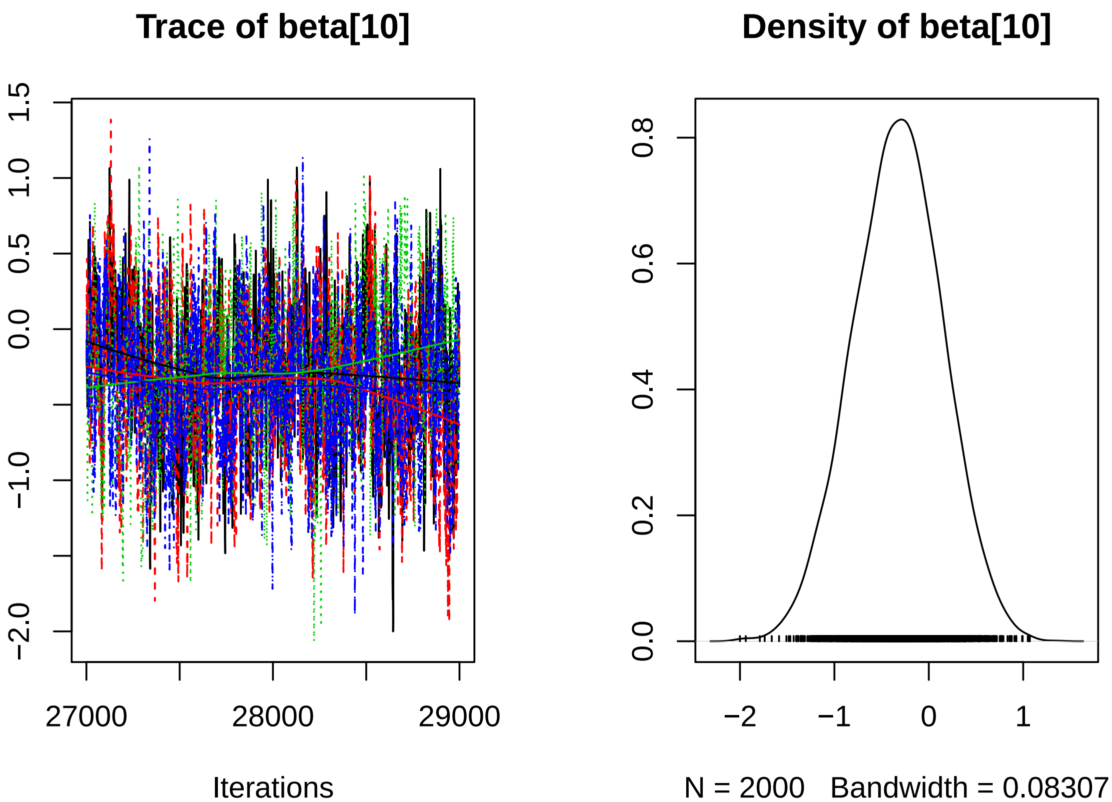


beta[10]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

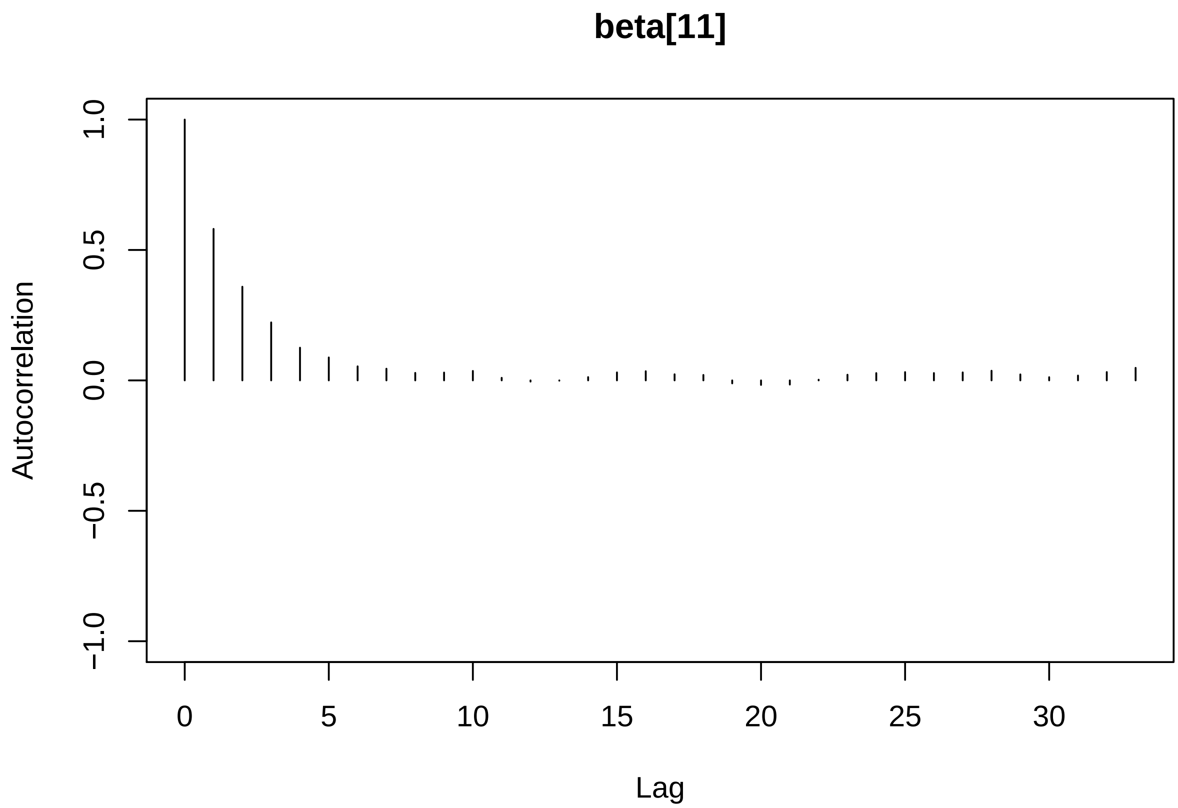

beta[10]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

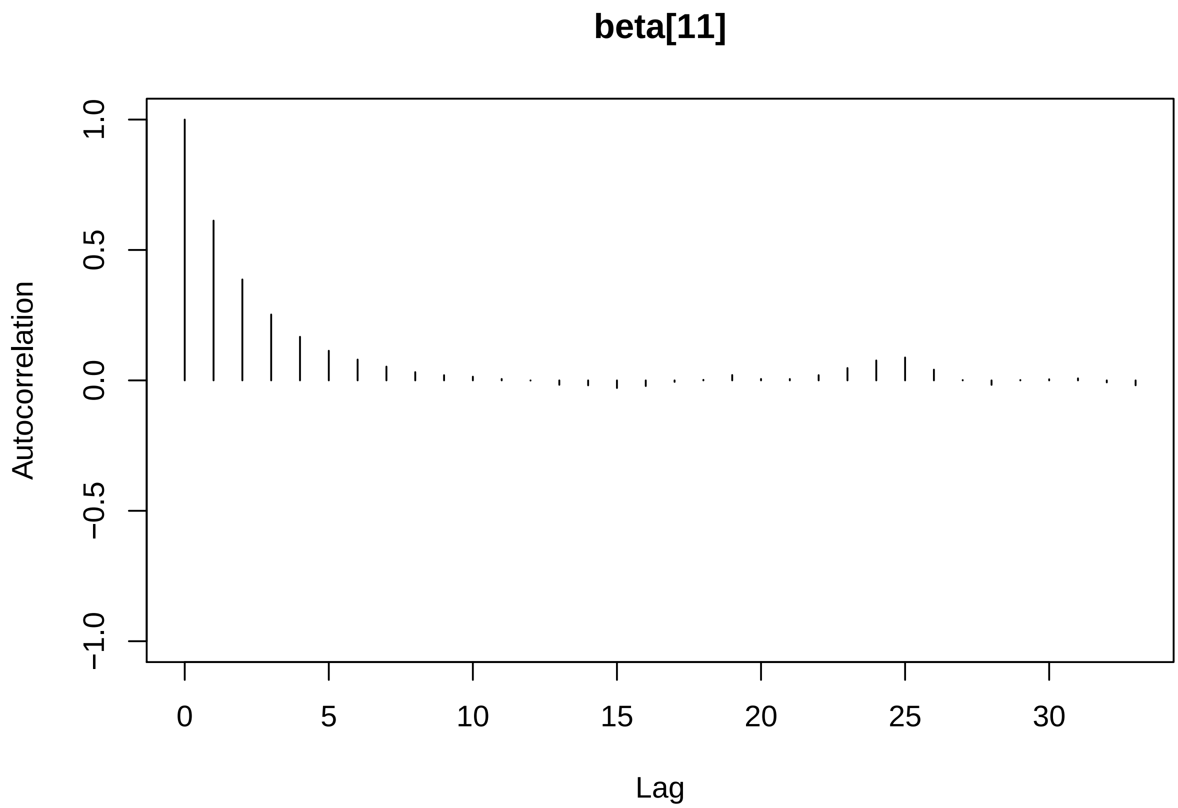


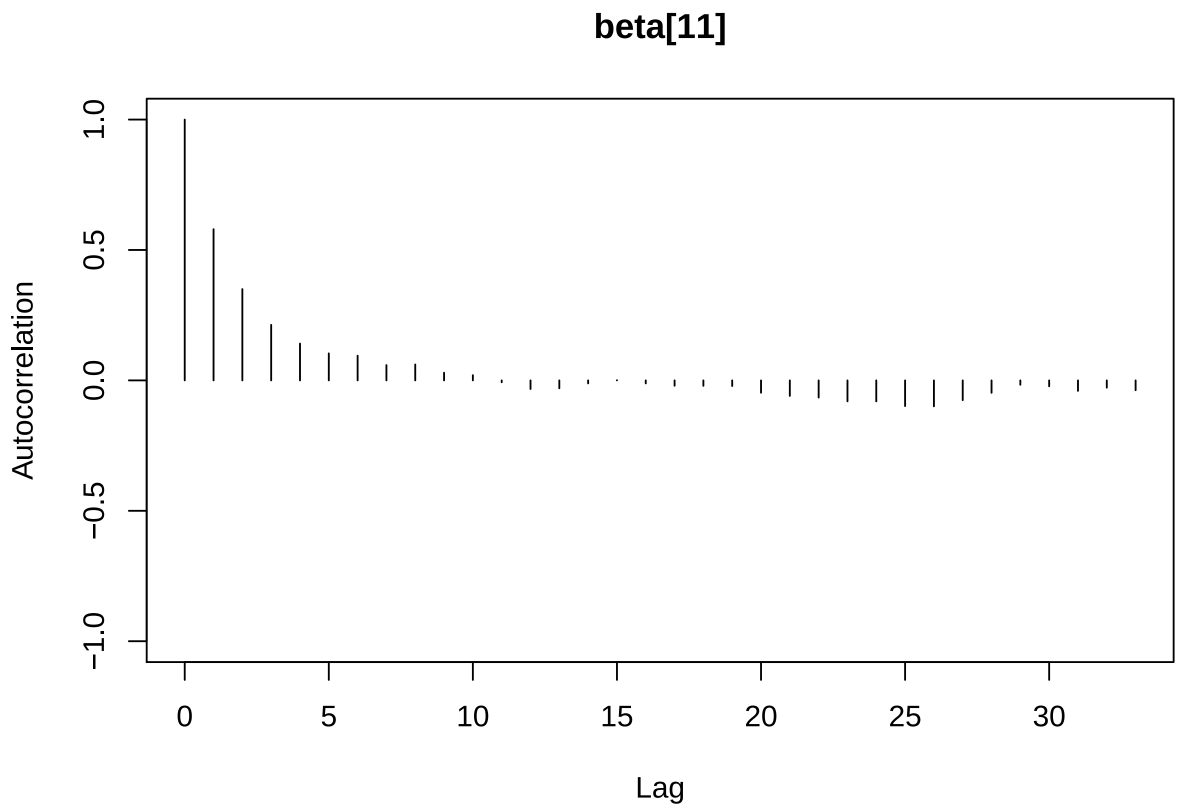
beta[10]
Autocorrelation
−1.0
−0.5
0.0
0.5
1.0
0
5
10
15
20
25
30
Lag

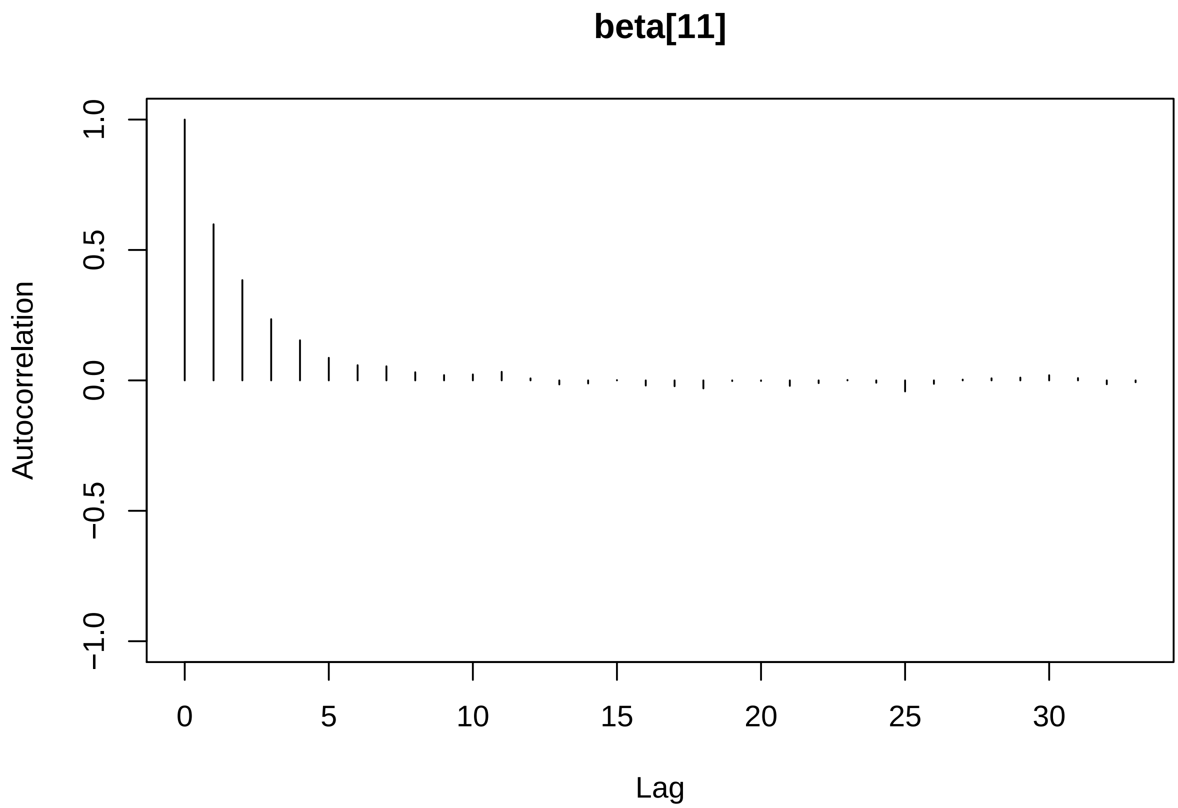


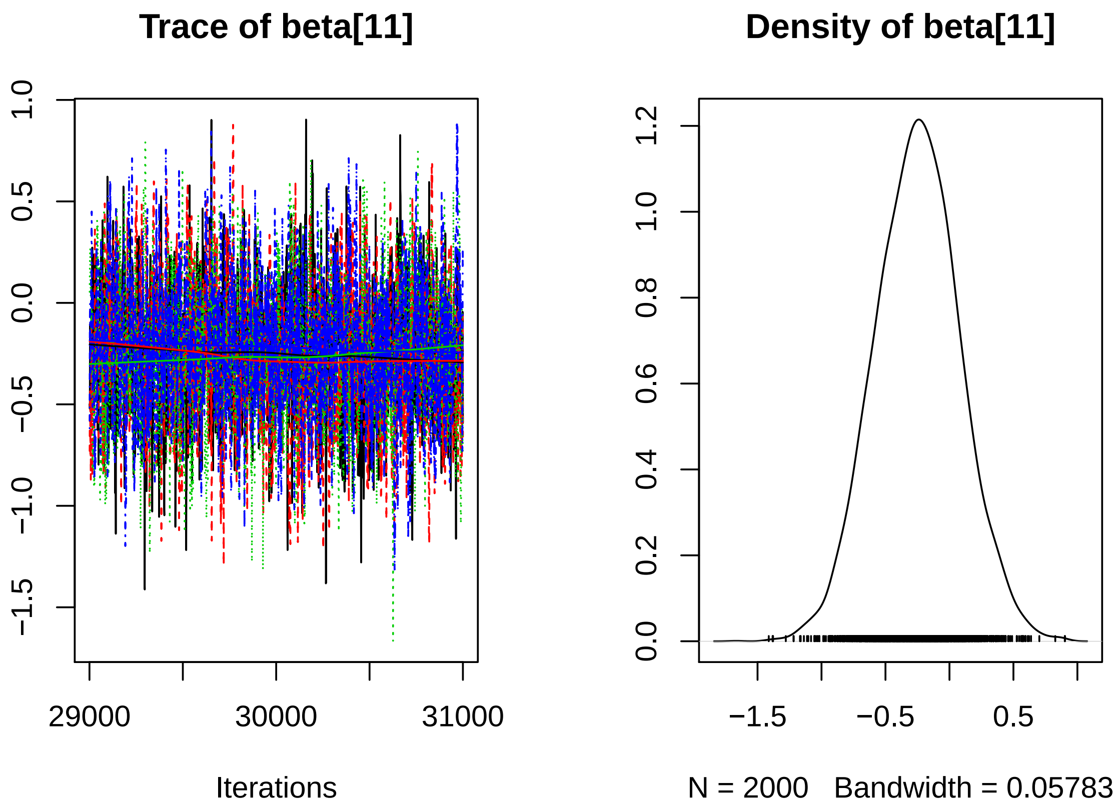
Trace of beta[10]
Density of beta[10]
1.5
1.0
0.5
0.0
−1.0
−2.0
27000
28000
29000
Iterations
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
N = 2000 Bandwidth = 0.08307

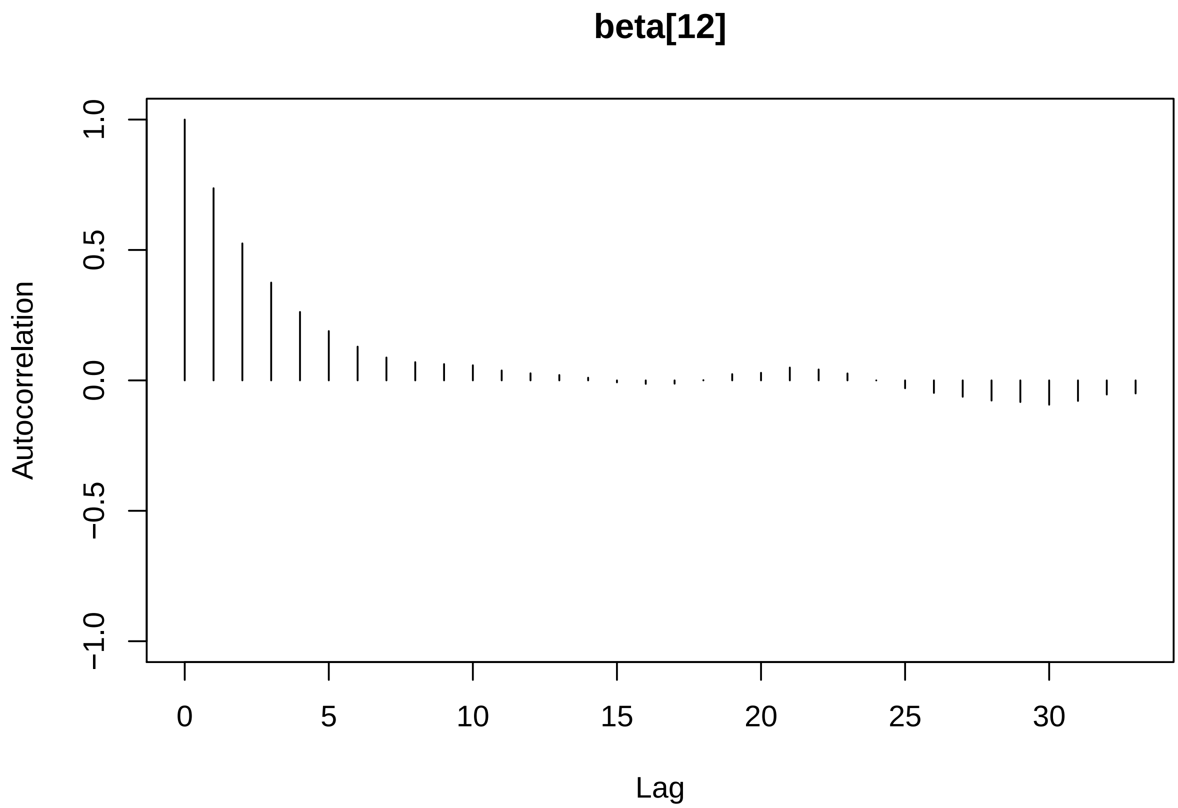

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

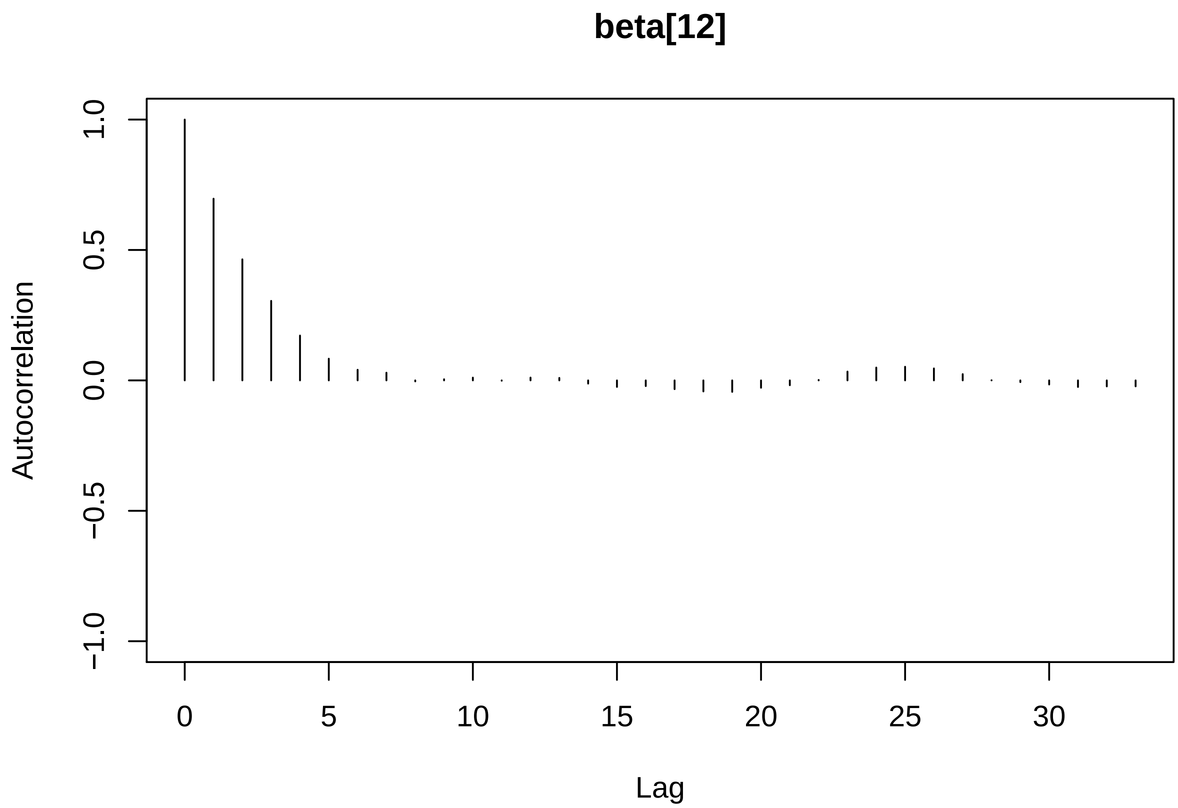

beta[11]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

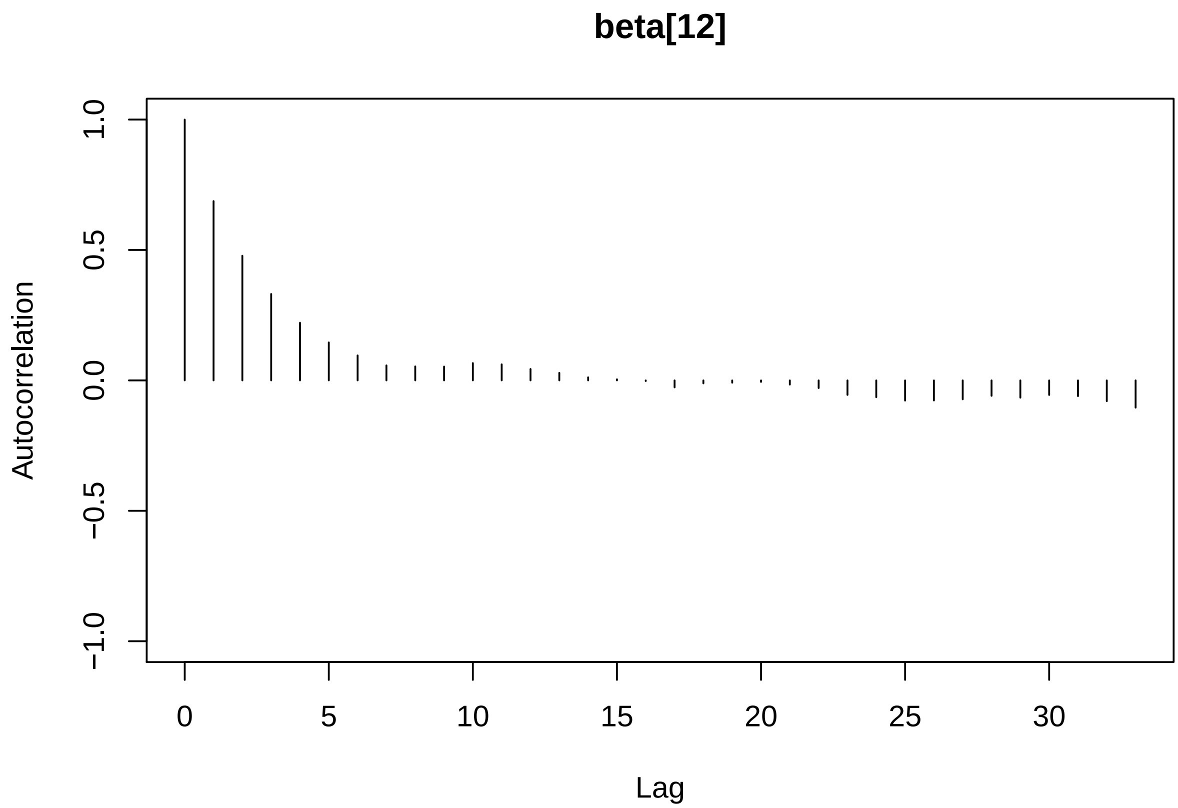

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

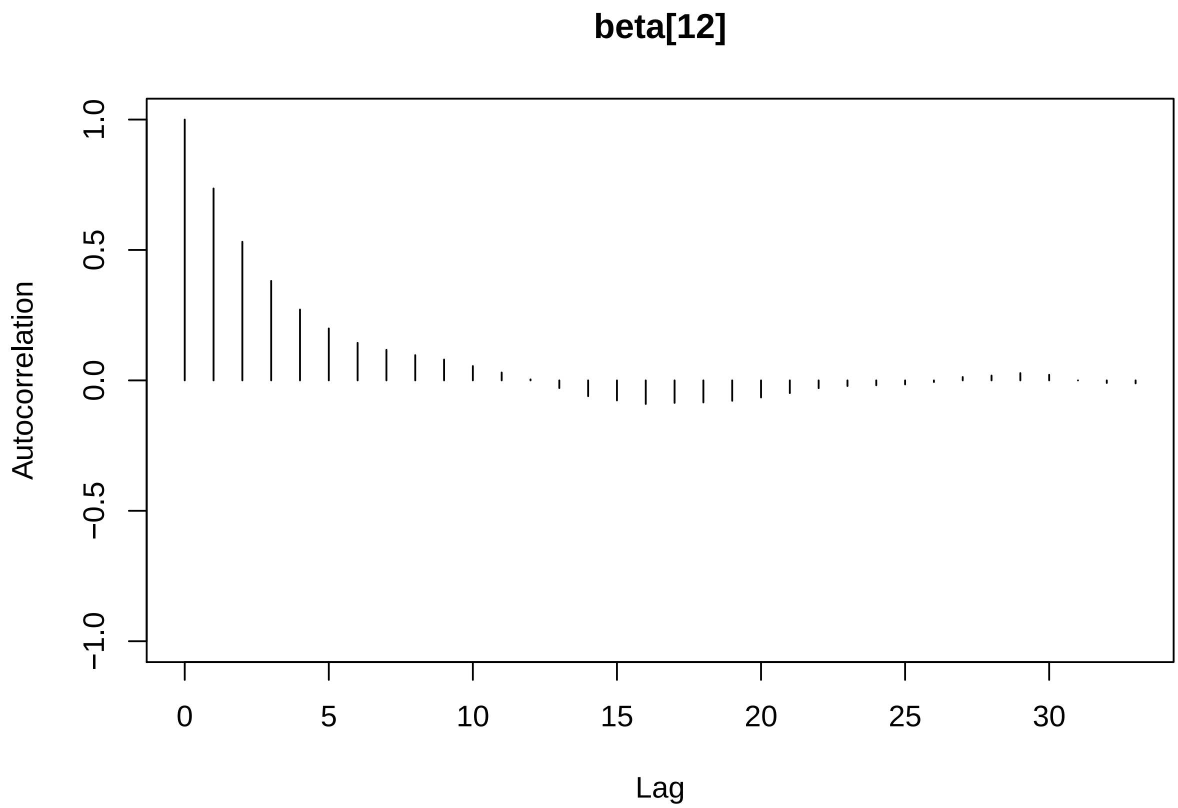

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

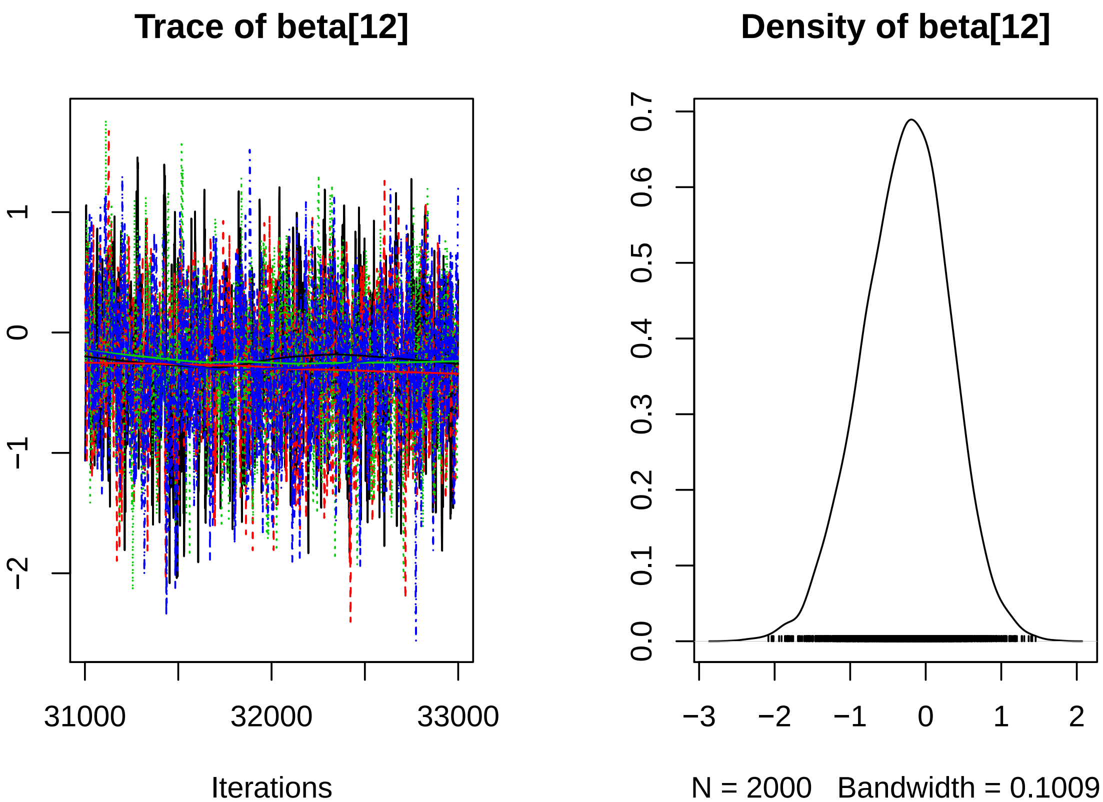

Trace of beta[11]
1.0
0.5
0.0
−0.5
−1.0
−1.5
29000
30000
31000
Iterations
Density of beta[11]
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.5
−0.5
0.5
N = 2000 Bandwidth = 0.05783

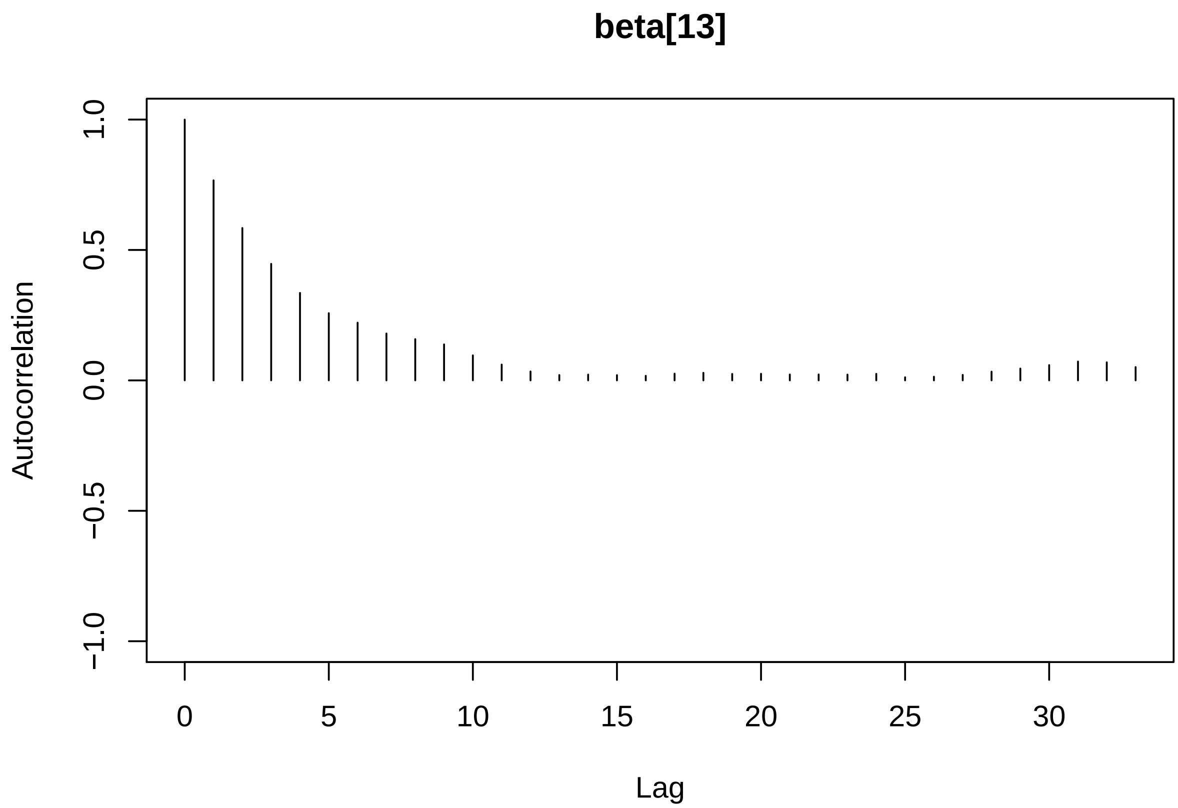

beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

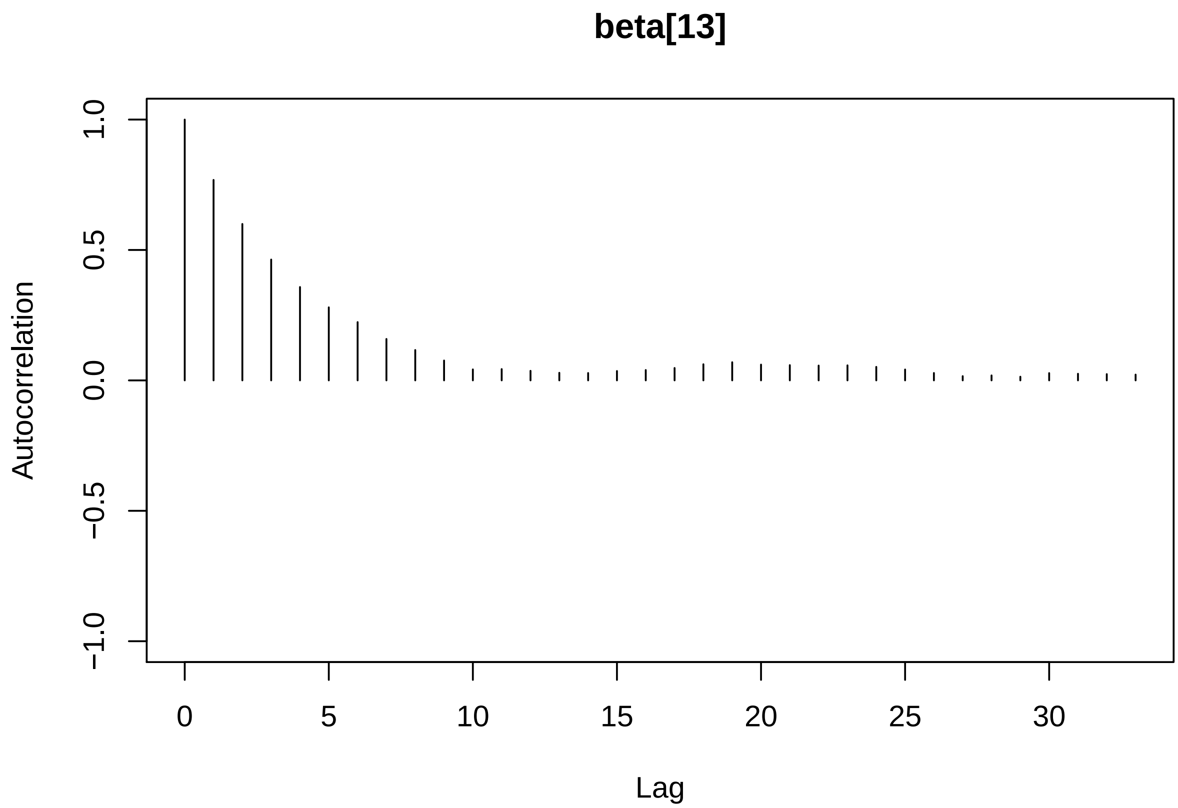

beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

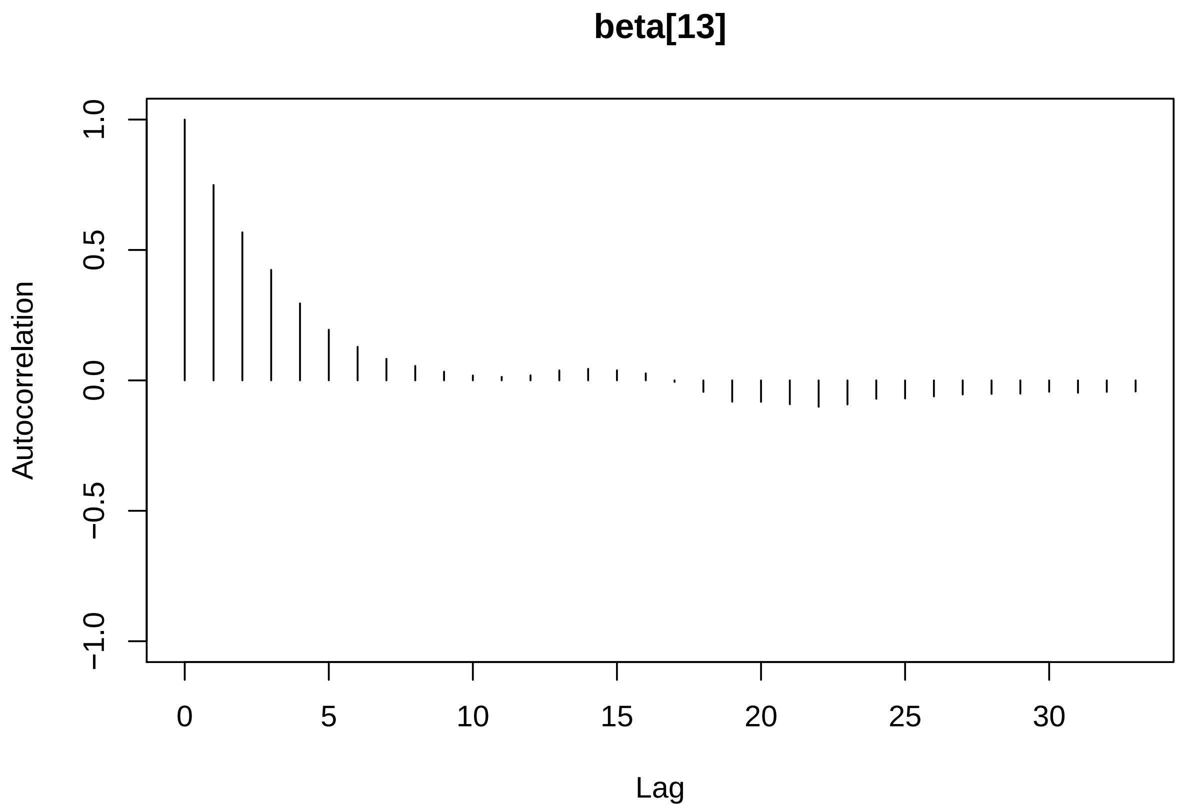

beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

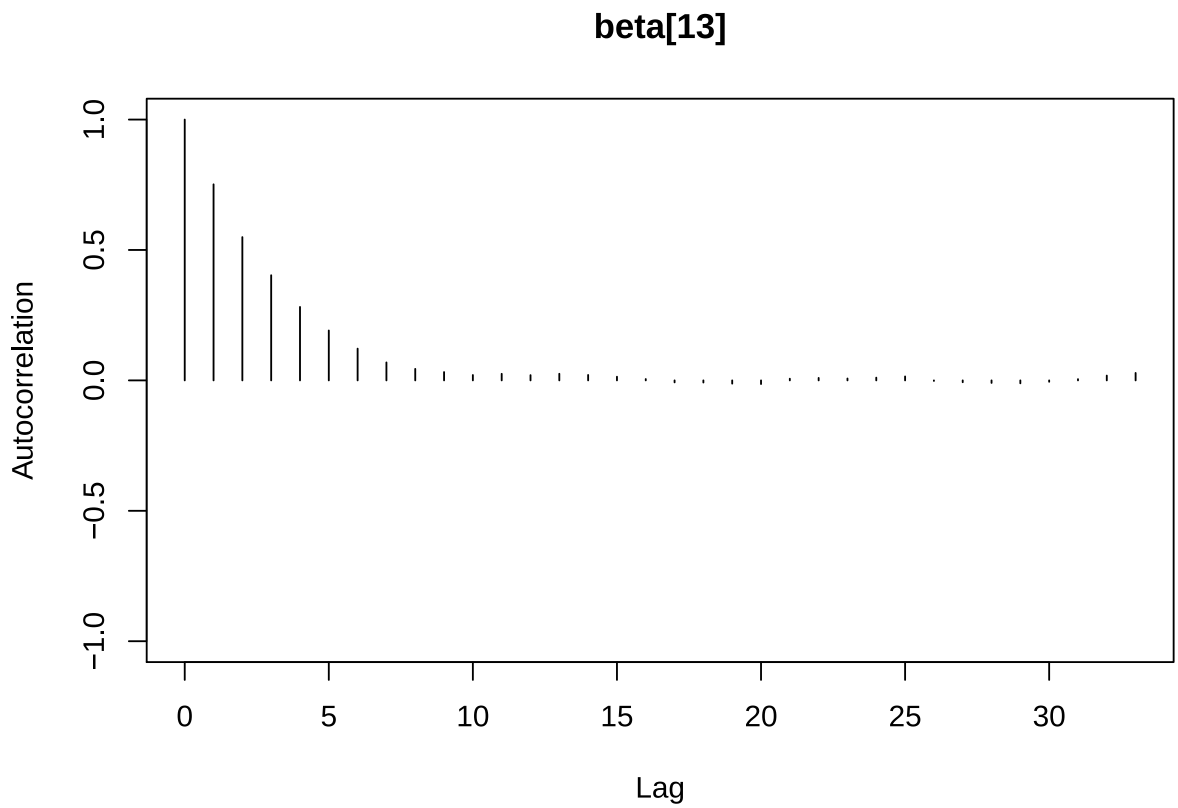

beta[12]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

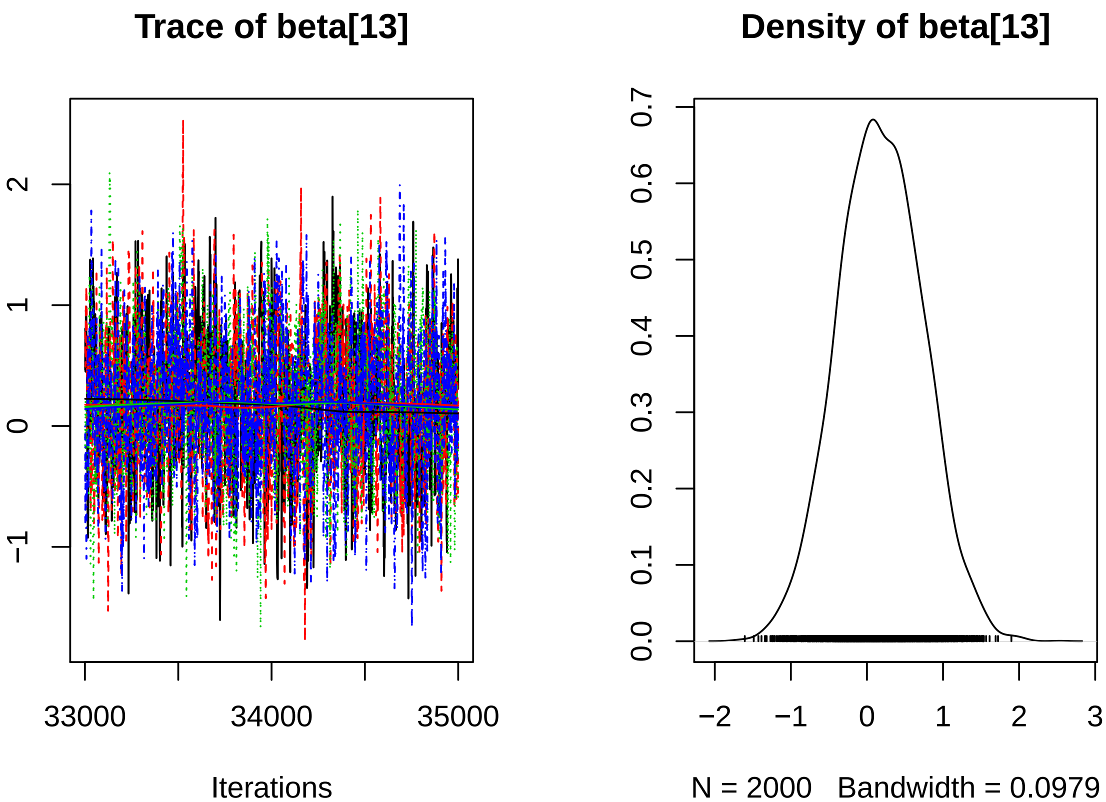

Trace of beta[12]
Density of beta[12]
Iterations
N = 2000 Bandwidth = 0.1009

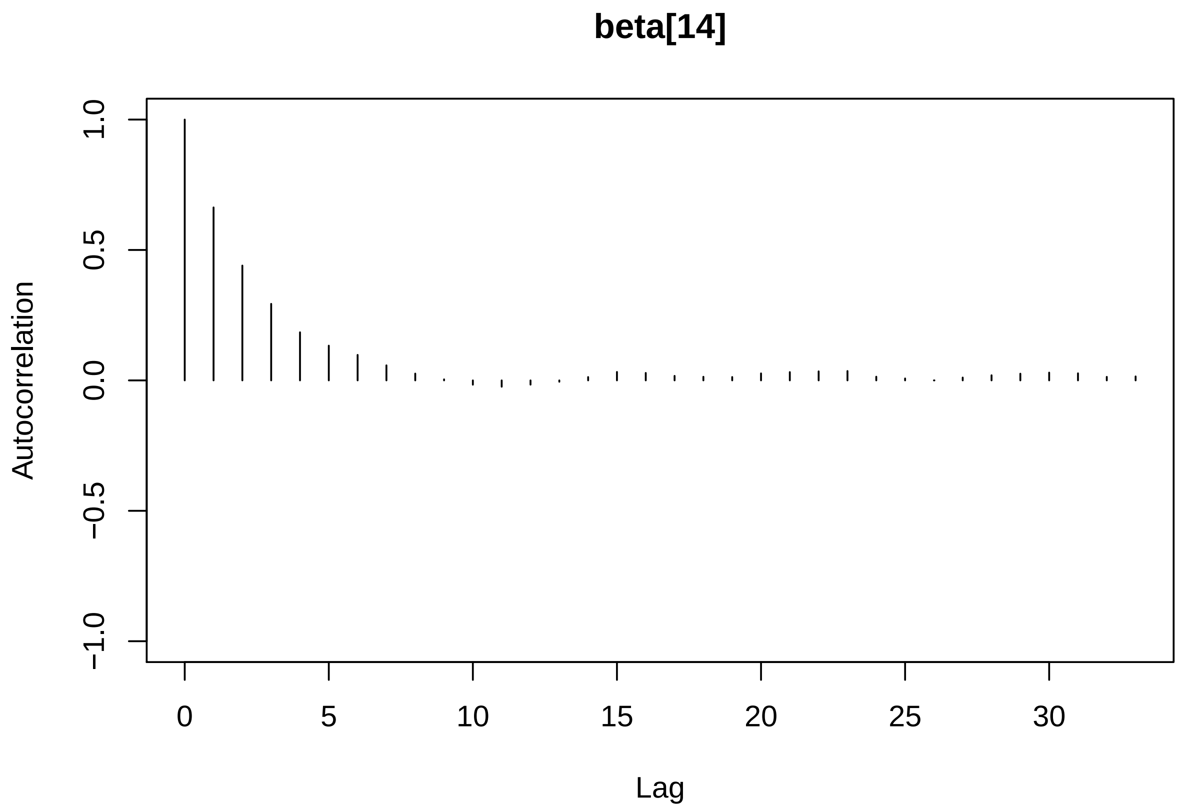

beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

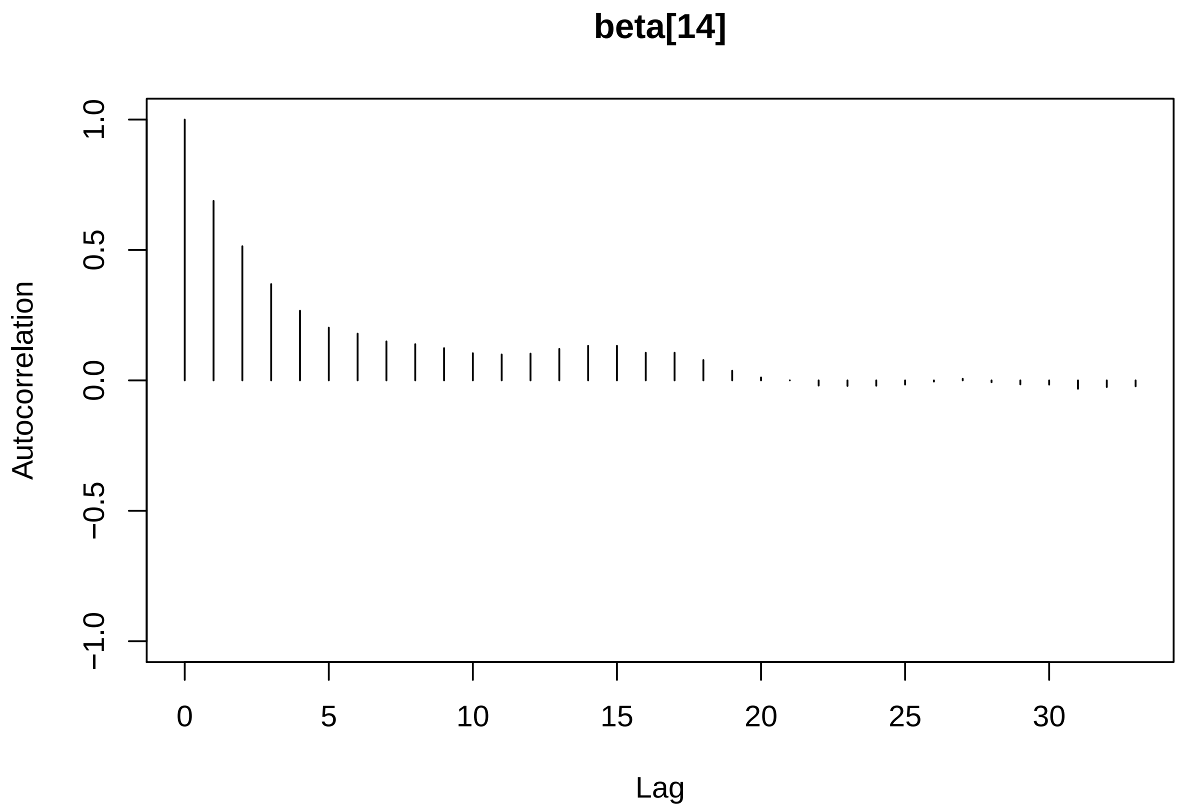

beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

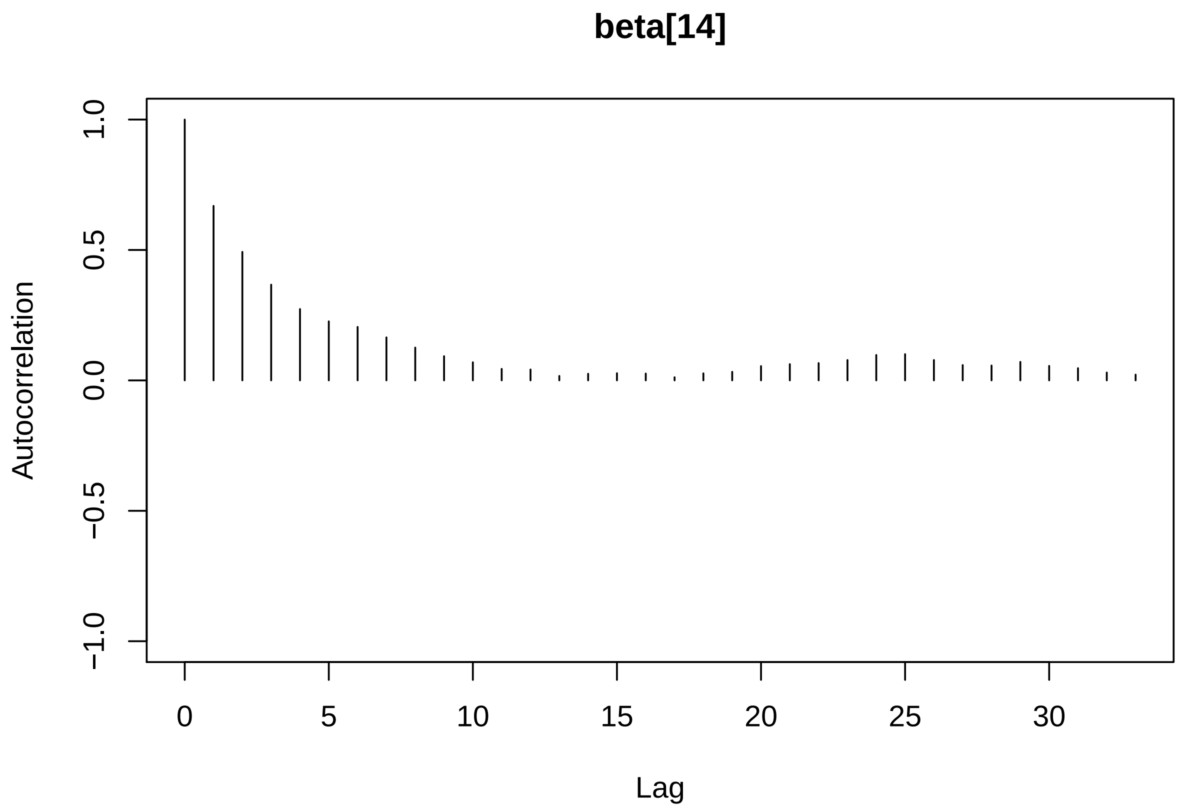

beta[13]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag


beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

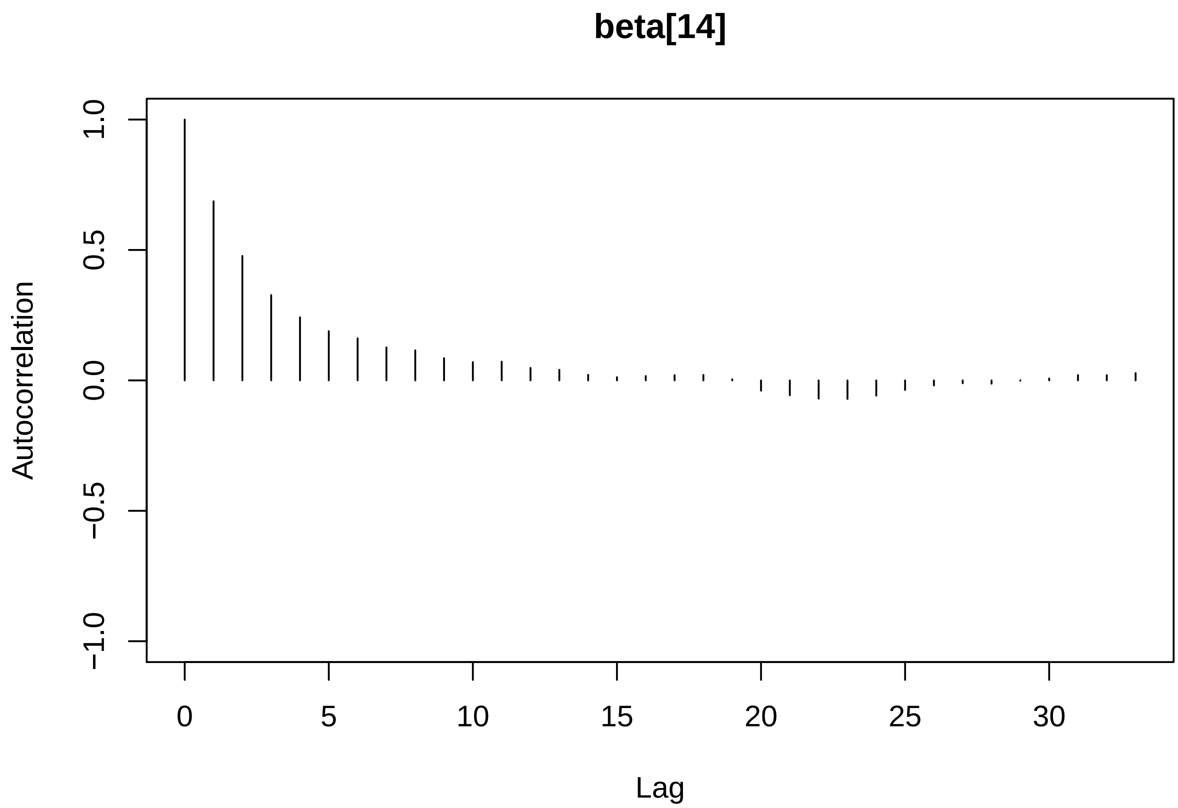

Trace of beta[13]
Iterations
Density of beta[13]
N = 2000 Bandwidth = 0.0979


beta[14]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

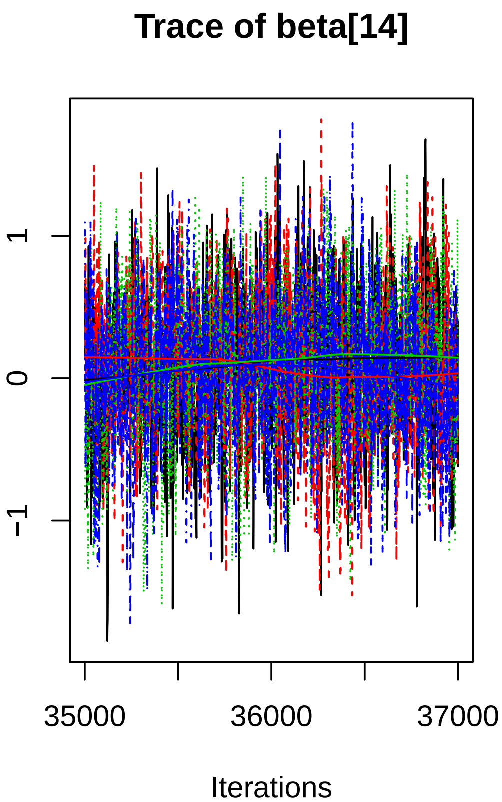

beta[14]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

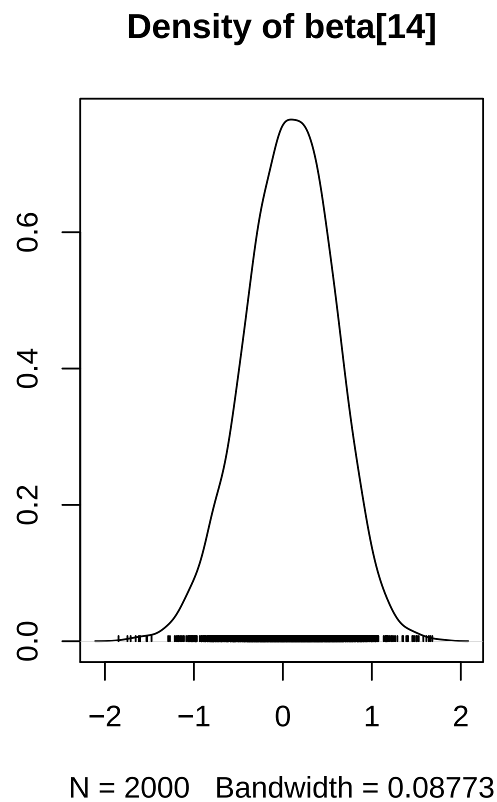


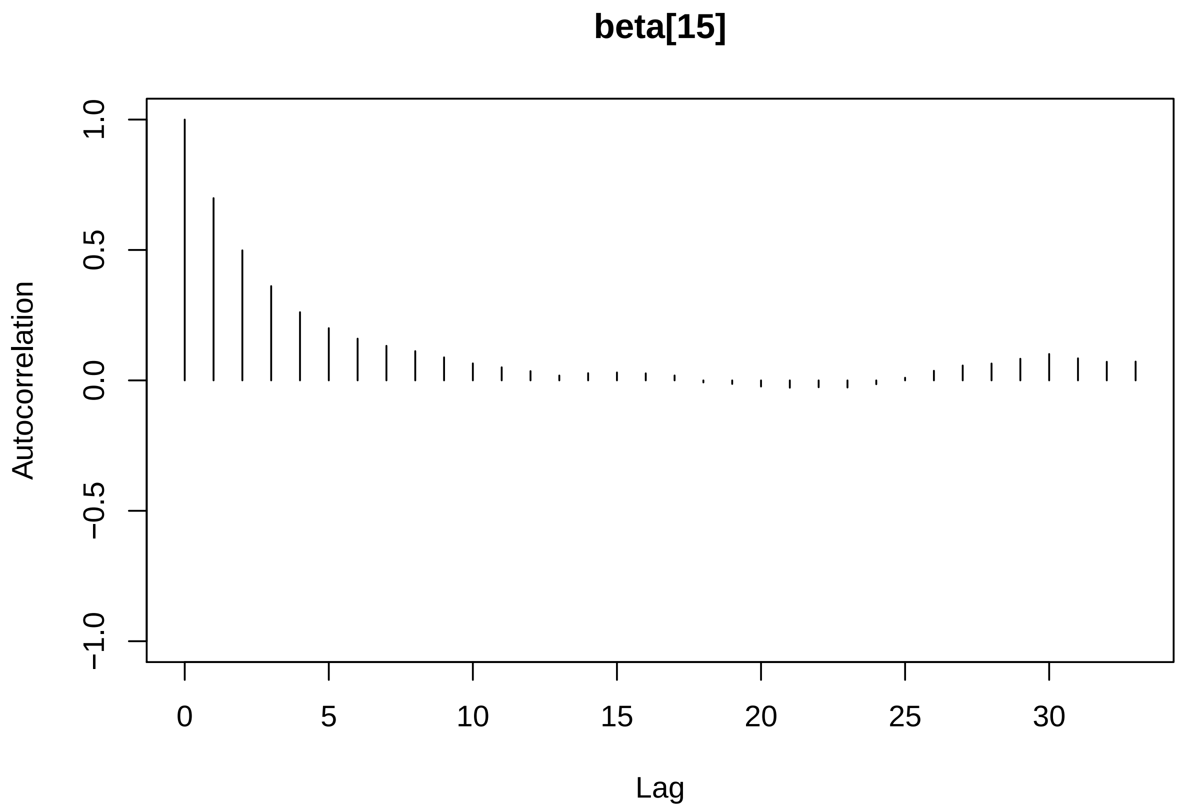
beta[14]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

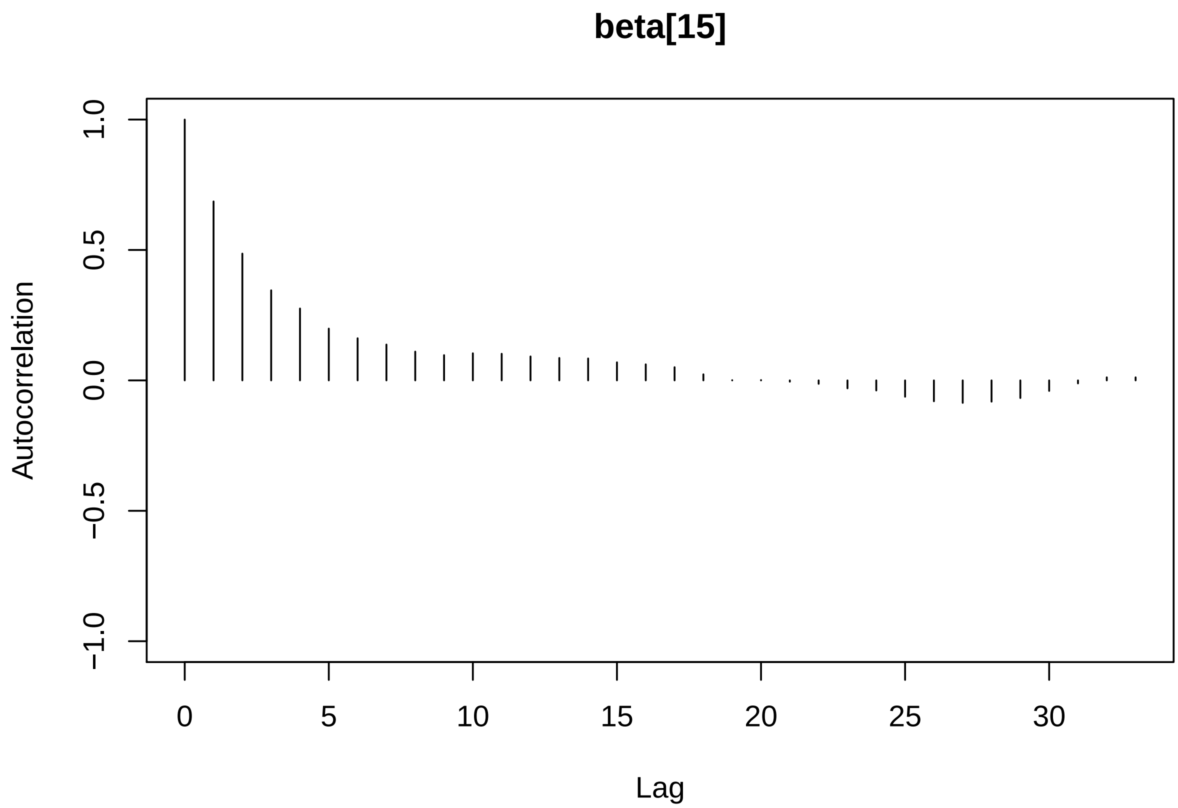


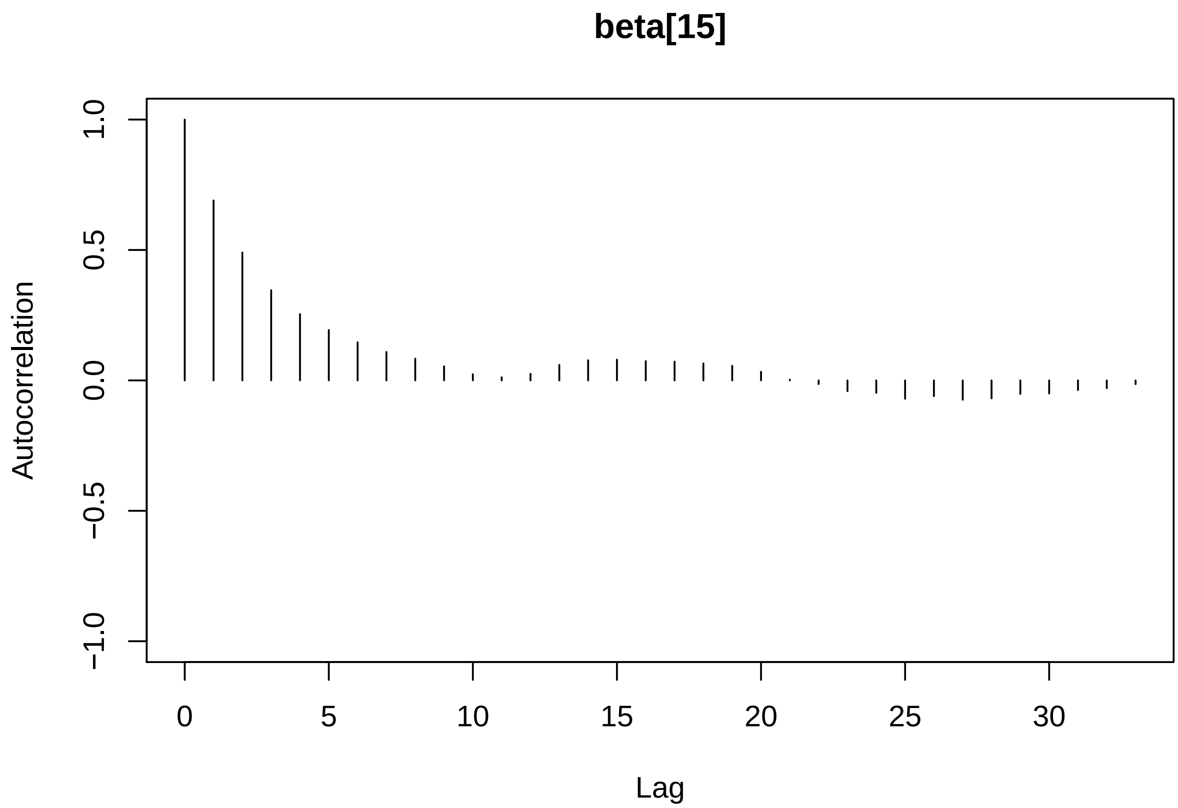
beta[14]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

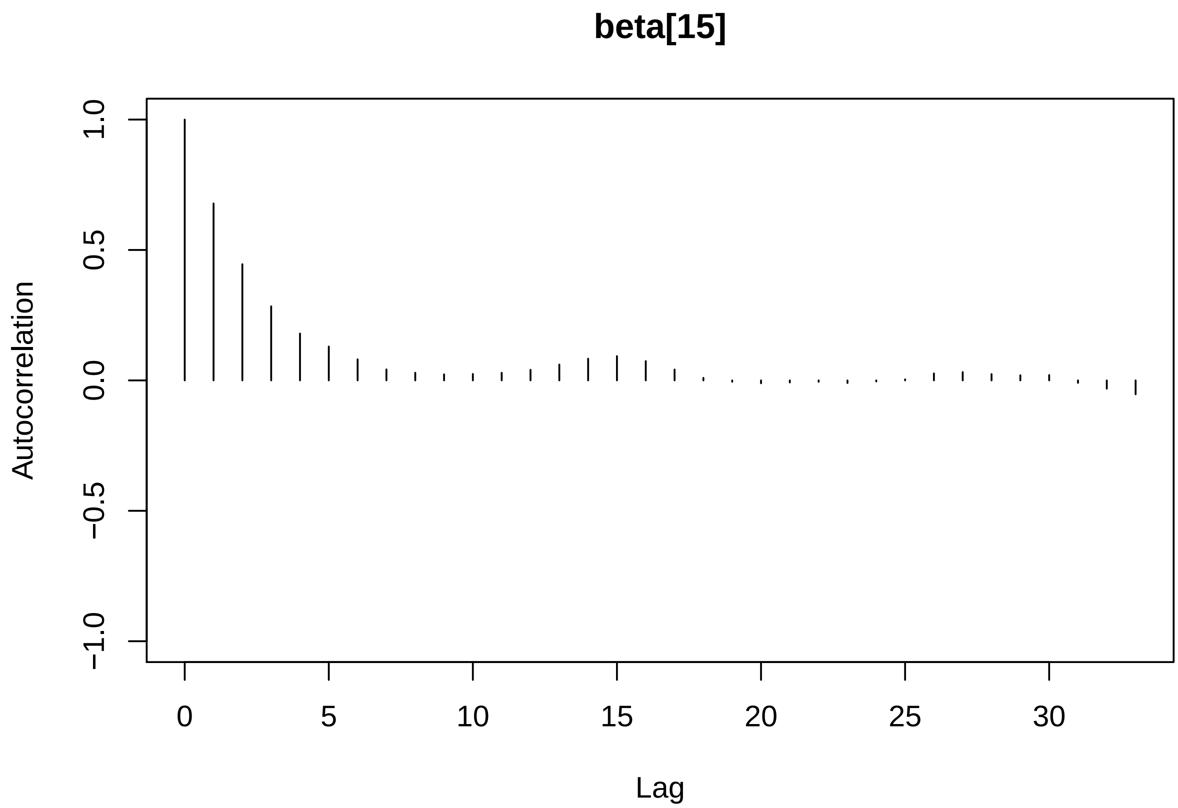

Trace of beta[14]
1
0
−1
35000
36000
37000
Iterations

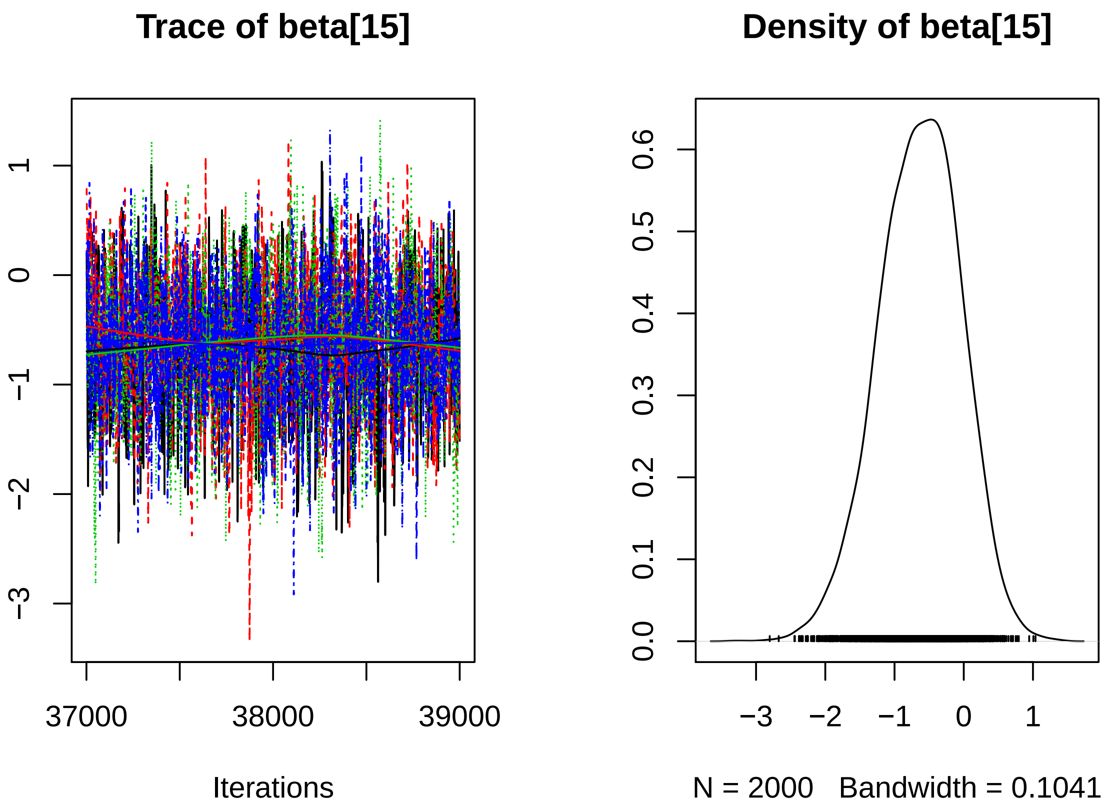

Density of beta[14]
0.6
0.4
0.2
0.0
−2
−1
0
1
2
N = 2000 Bandwidth = 0.08773


beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

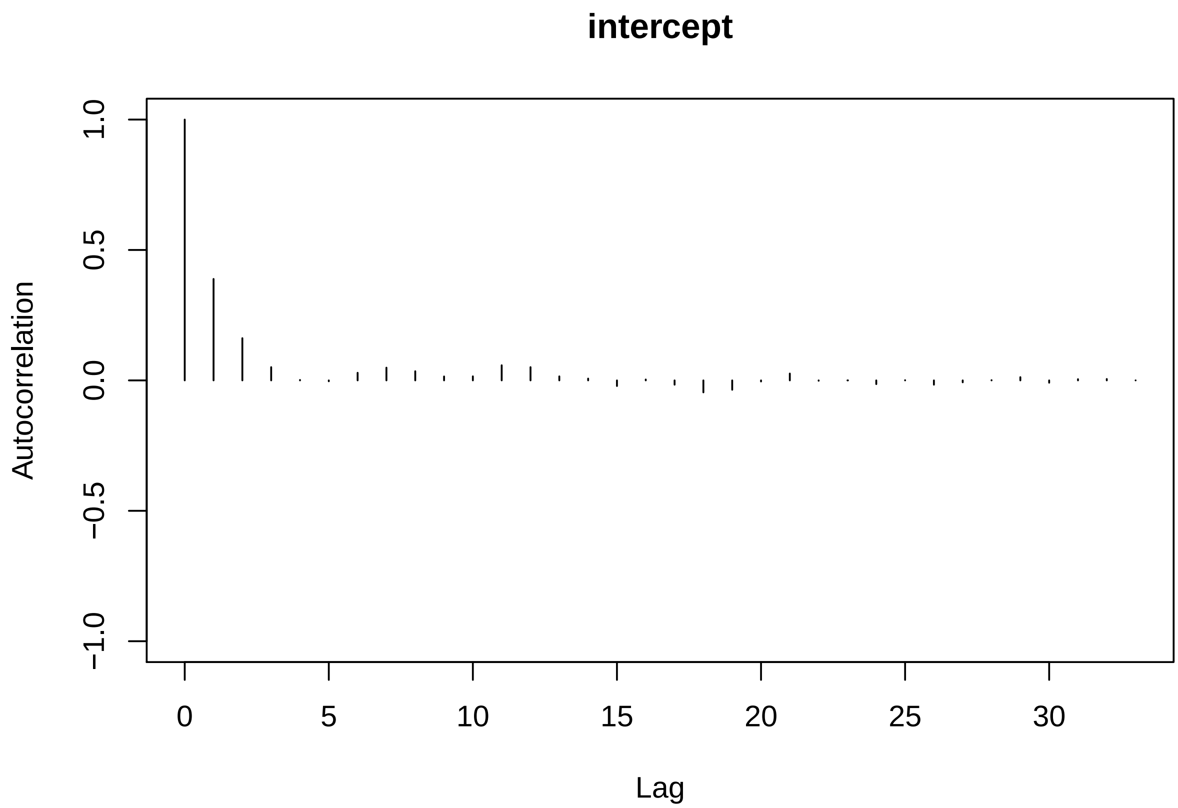

beta[15]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag


beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

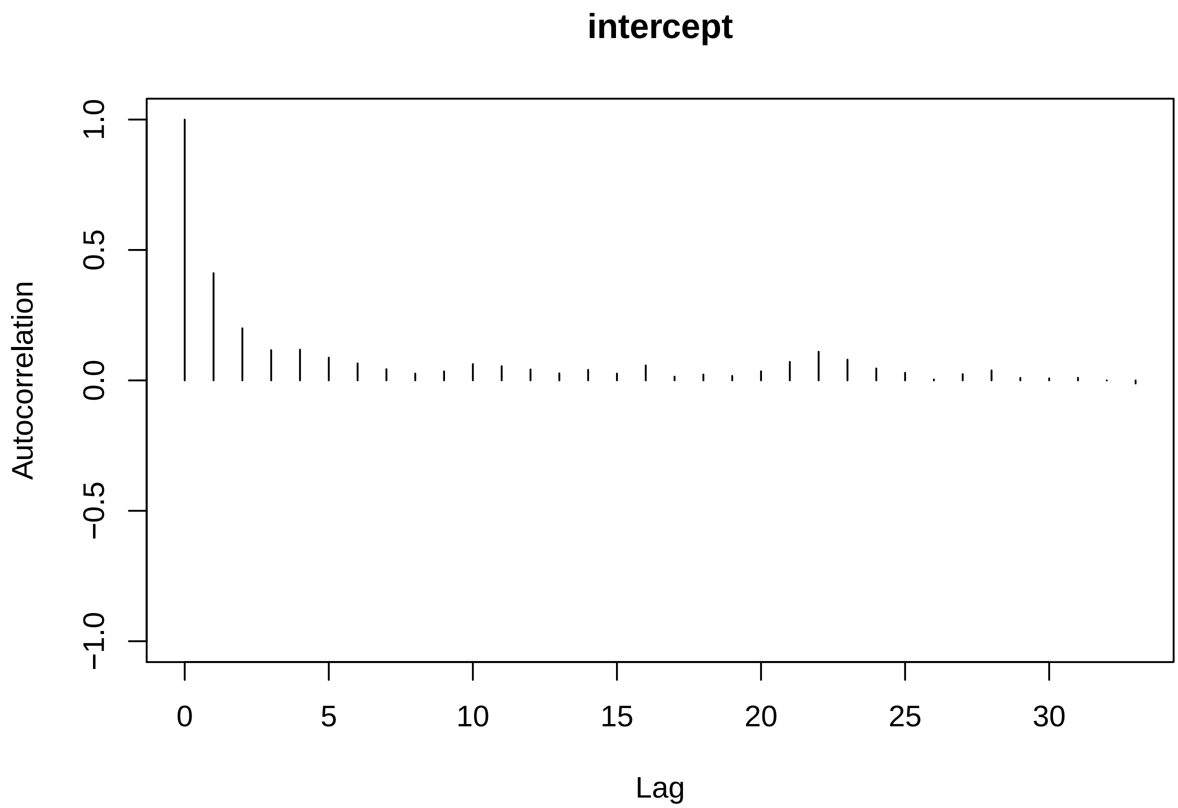

beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

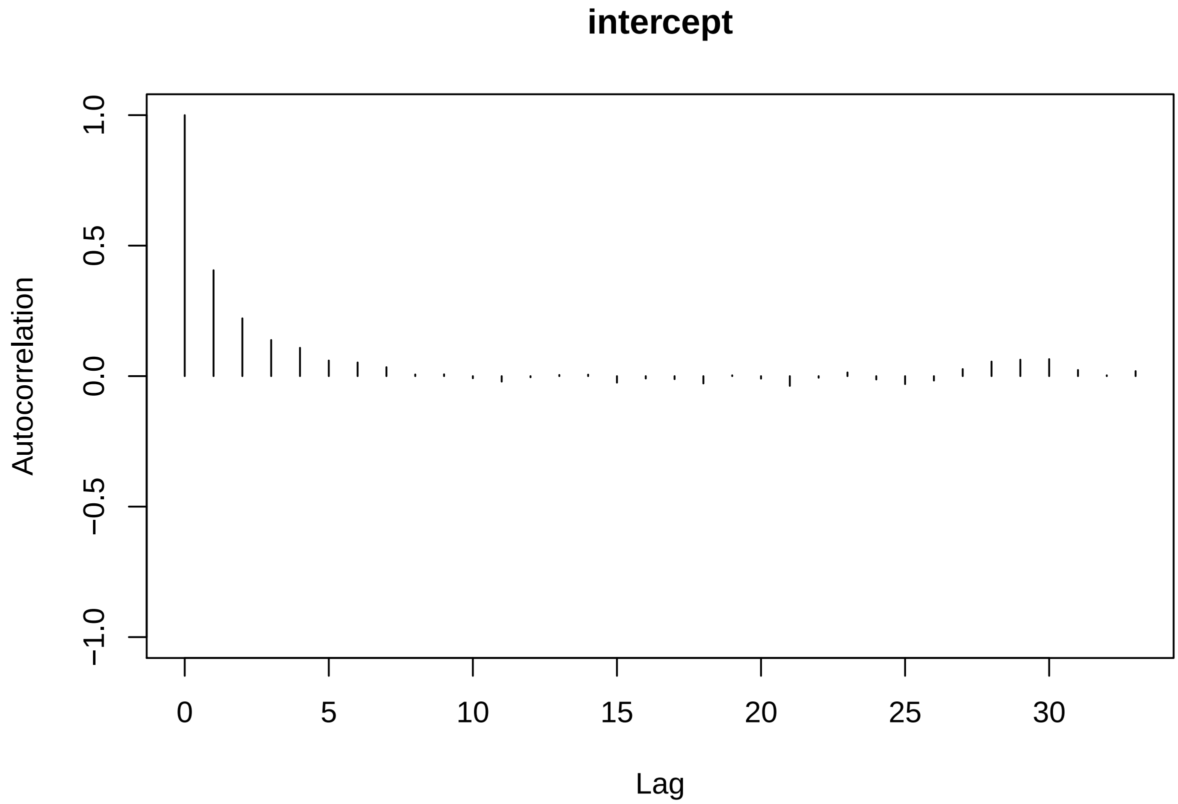


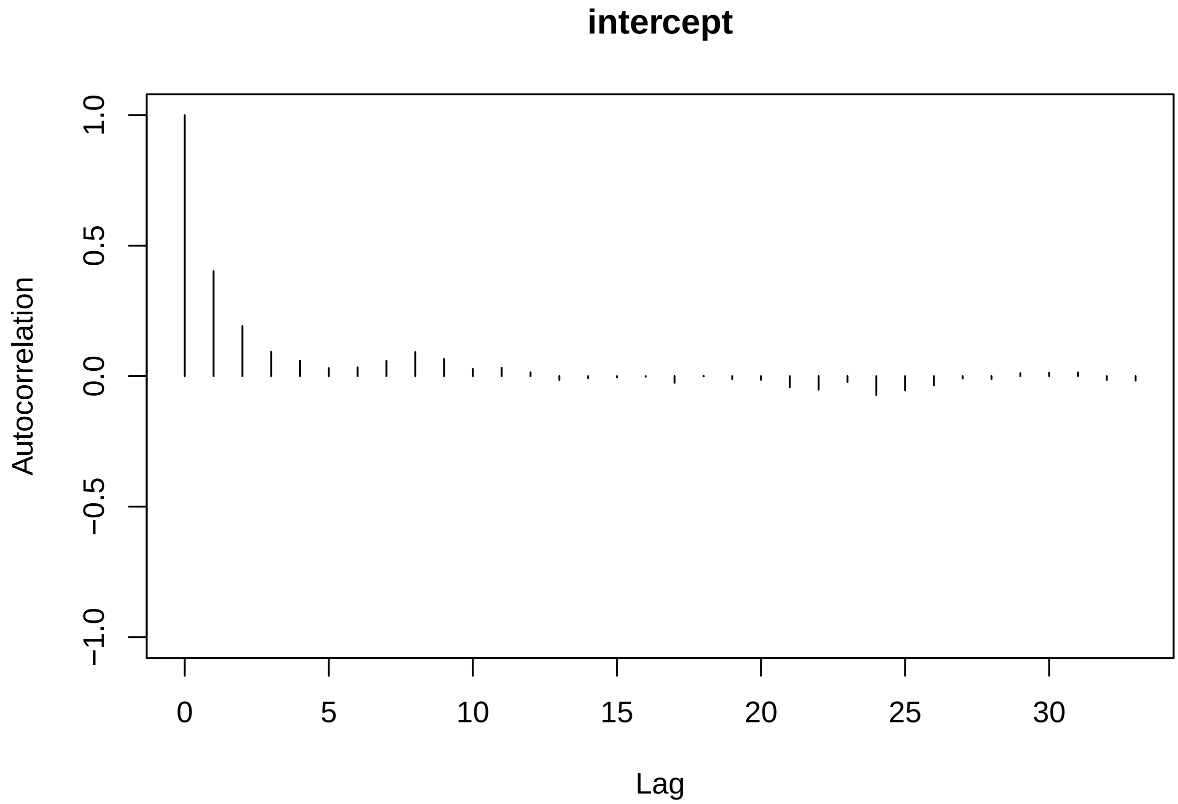
Trace of beta[15]
1
0
−1
−2
−3
37000
38000
39000
Iterations
Density of beta[15]
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−3
−2
−1
0
1
N = 2000 Bandwidth = 0.1041

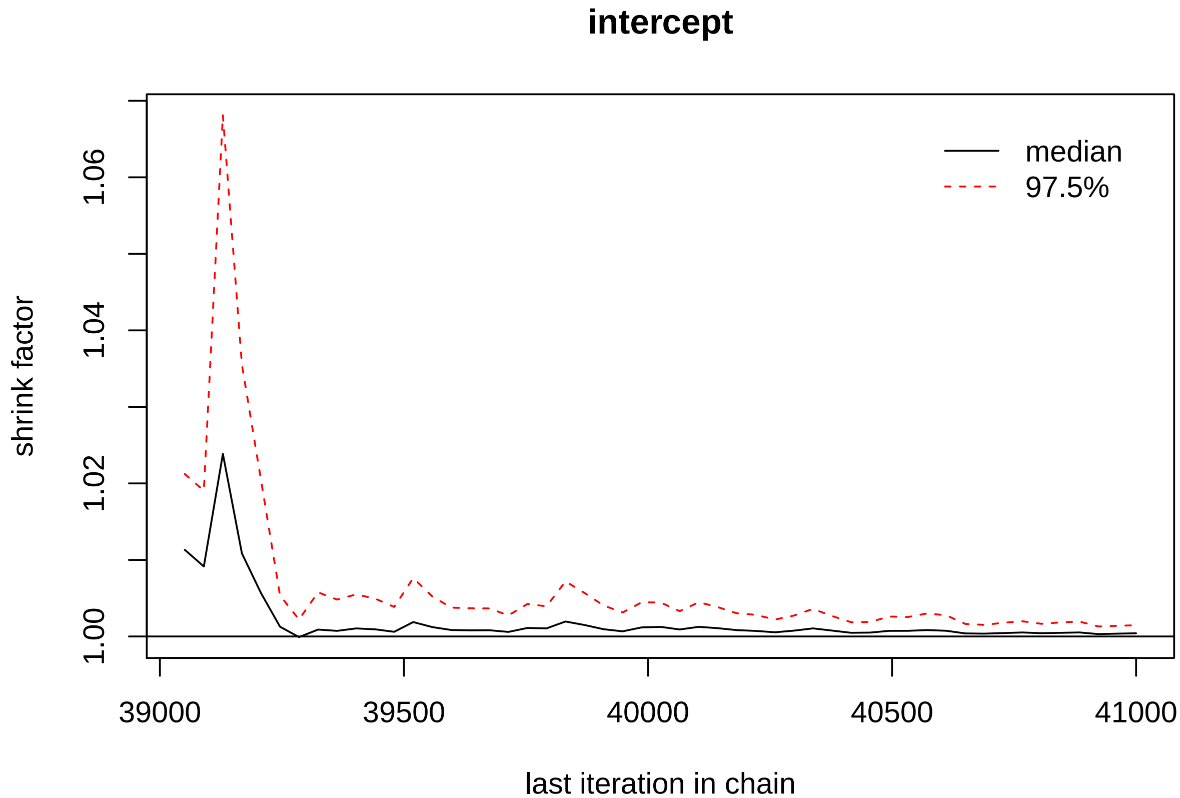


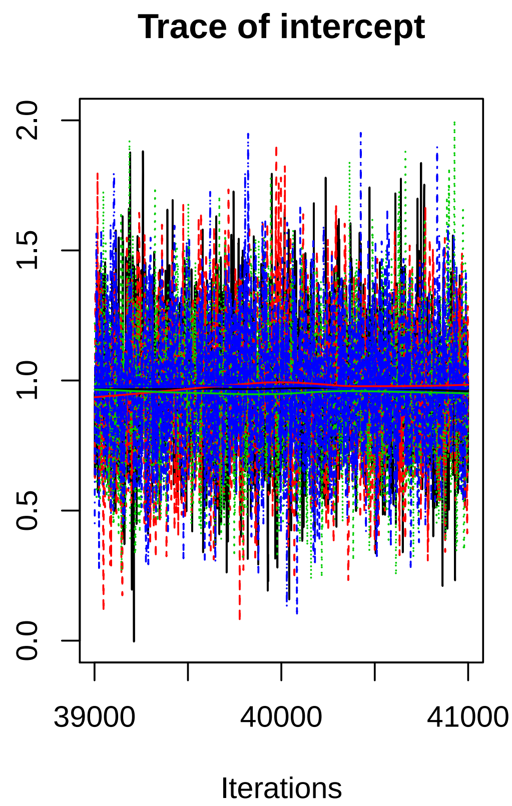
intercept
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

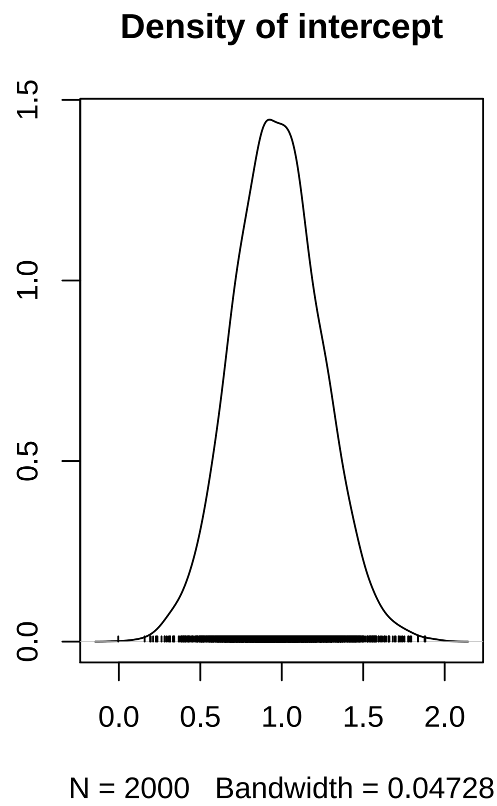

intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
39000
39500
40000
40500
41000
last iteration in chain





$\tau = 100$ **Posterior quantiles**

```
summary(samp.coeff.uninformative)
```

```
##
## Iterations = 3001:7000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 4000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean       SD  Naive SE Time-series SE
## beta[1]     3.477e-04 0.009949 7.866e-05      1.016e-04
## beta[2]     1.018e-03 0.009898 7.825e-05      1.004e-04
## beta[3]    -3.218e-03 0.010167 8.038e-05      1.058e-04
## beta[4]    -4.978e-04 0.010007 7.912e-05      1.024e-04
## beta[5]    -3.660e-04 0.010027 7.927e-05      1.024e-04
## beta[6]    -1.588e-03 0.010030 7.929e-05      1.015e-04
## beta[7]     3.218e-03 0.010081 7.970e-05      1.008e-04
```

```
## beta[8]     1.071e-04 0.009971 7.883e-05      9.957e-05
## beta[9]     4.737e-05 0.010035 7.934e-05      1.019e-04
## beta[10]   -1.604e-03 0.009927 7.848e-05      9.617e-05
## beta[11]   -1.177e-03 0.009993 7.900e-05      9.735e-05
## beta[12]   -1.581e-03 0.009817 7.761e-05      9.390e-05
## beta[13]   -1.556e-03 0.010206 8.068e-05      1.059e-04
## beta[14]   -1.382e-03 0.010027 7.927e-05      9.909e-05
## beta[15]   -1.752e-03 0.009956 7.871e-05      1.035e-04
## intercept  2.158e-03 0.009923 7.845e-05      1.002e-04
##
## 2. Quantiles for each variable:
##
##                2.5%       25%        50%      75%   97.5%
## beta[1]   -0.01939 -0.006299  4.187e-04 0.007171 0.01942
## beta[2]   -0.01842 -0.005586  9.599e-04 0.007664 0.02048
## beta[3]   -0.02288 -0.010081 -3.218e-03 0.003456 0.01705
## beta[4]   -0.02018 -0.007300 -5.475e-04 0.006309 0.01911
## beta[5]   -0.02004 -0.007151 -4.247e-04 0.006437 0.01942
## beta[6]   -0.02124 -0.008374 -1.577e-03 0.005303 0.01793
## beta[7]   -0.01657 -0.003549  3.290e-03 0.010033 0.02299
## beta[8]   -0.01969 -0.006536  1.013e-04 0.006750 0.01967
## beta[9]   -0.01958 -0.006783  6.149e-05 0.006923 0.01961
## beta[10]  -0.02154 -0.008227 -1.478e-03 0.005065 0.01759
## beta[11]  -0.02073 -0.007922 -1.220e-03 0.005566 0.01843
## beta[12]  -0.02082 -0.008235 -1.752e-03 0.005148 0.01758
## beta[13]  -0.02141 -0.008502 -1.572e-03 0.005407 0.01843
## beta[14]  -0.02104 -0.008077 -1.469e-03 0.005396 0.01802
## beta[15]  -0.02147 -0.008418 -1.684e-03 0.004987 0.01763
## intercept -0.01722 -0.004669  2.091e-03 0.008895 0.02156
```

$\tau = 100$ **Sample again and estimate the mean and MAP mode of the posterior dostributions.**

```r
samp.coeff.jags.uninformative <- jags.samples(model.carolinas.uninformative,
    variable.names = c("intercept", "beta"),
    n.iter = 2 * nSamples, progress.bar = "none")
posterior_means.uninformative <- lapply(samp.coeff.jags.uninformative,
    apply, 1, "mean")
pander(posterior_means.uninformative, caption = "posterior_means.uninformative")
```

- **beta**: *0.0005006*, *0.001018*, *-0.003278*, *-0.0004608*, *-0.0003554*, *-0.001501*, *0.003236*, *1.027e-05*, *0.0001927*, *-0.001338*, *-0.00126*, *-0.001865*, *-0.001637*, *-0.001249* and *-0.001775*
- **intercept**: *0.002009*

```r
posterior_means.uninformative <- lapply(samp.coeff.jags.uninformative,
    apply, 1, "mlv")
posterior_means.uninformative
```

```
## $beta
## $beta[[1]]
## Mode (most likely value): 0.0001343827
## Bickel's modal skewness: 0.0235
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[2]]
## Mode (most likely value): 0.001732364
## Bickel's modal skewness: -0.059125
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[3]]
## Mode (most likely value): -0.004063409
## Bickel's modal skewness: 0.063625
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[4]]
## Mode (most likely value): -5.255803e-06
## Bickel's modal skewness: -0.031125
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[5]]
## Mode (most likely value): -1.397581e-05
## Bickel's modal skewness: -0.02775
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[6]]
## Mode (most likely value): -0.001500696
## Bickel's modal skewness: -0.00075
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[7]]
## Mode (most likely value): 0.003568656
## Bickel's modal skewness: -0.02875
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[8]]
## Mode (most likely value): 0.0004487259
## Bickel's modal skewness: -0.034125
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[9]]
## Mode (most likely value): 3.926732e-05
## Bickel's modal skewness: 0.010625
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[10]]
## Mode (most likely value): -0.0012739
```

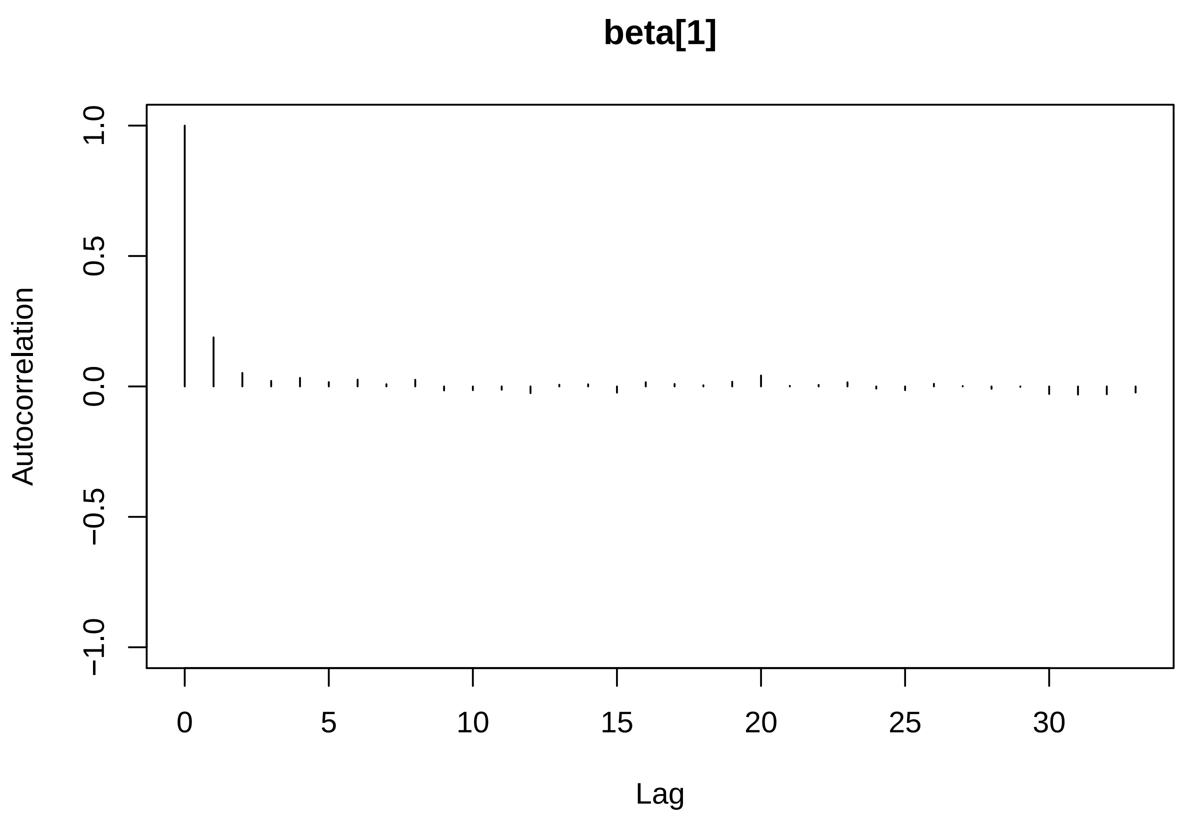
```
## Bickel's modal skewness: 0.000625
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[11]]
## Mode (most likely value): -0.001732697
## Bickel's modal skewness: 0.037125
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[12]]
## Mode (most likely value): -0.001645683
## Bickel's modal skewness: -0.015625
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[13]]
## Mode (most likely value): -0.001663144
## Bickel's modal skewness: -0.004375
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[14]]
## Mode (most likely value): -0.001974153
## Bickel's modal skewness: 0.051375
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
## $beta[[15]]
## Mode (most likely value): -0.001117388
## Bickel's modal skewness: -0.048125
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
##
##
## $intercept
## $intercept[[1]]
## Mode (most likely value): 0.001545672
## Bickel's modal skewness: 0.036
## Call: mlv.default(x = array(newX[, i], d.call, dn.call))
```

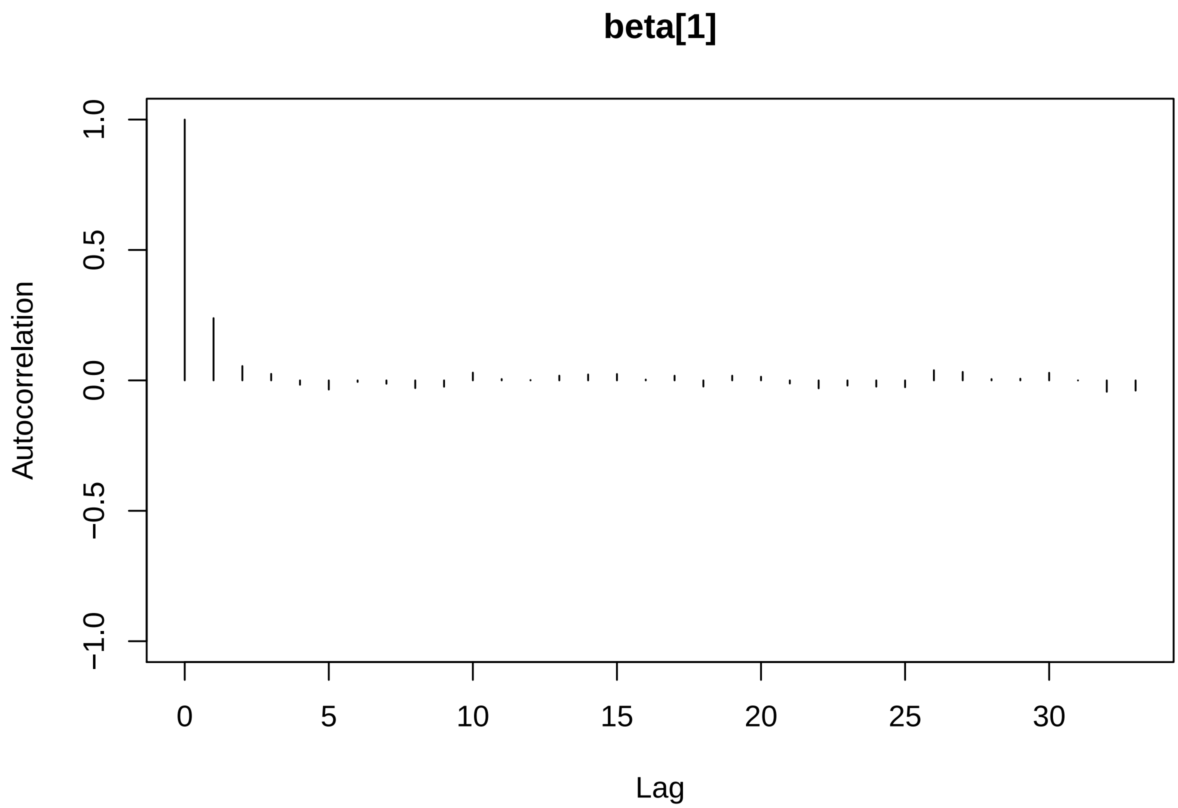
$\tau = 100$ Plot the time series, empirical posterior distribution, and the autocoerrelation fucntion for the coefficients

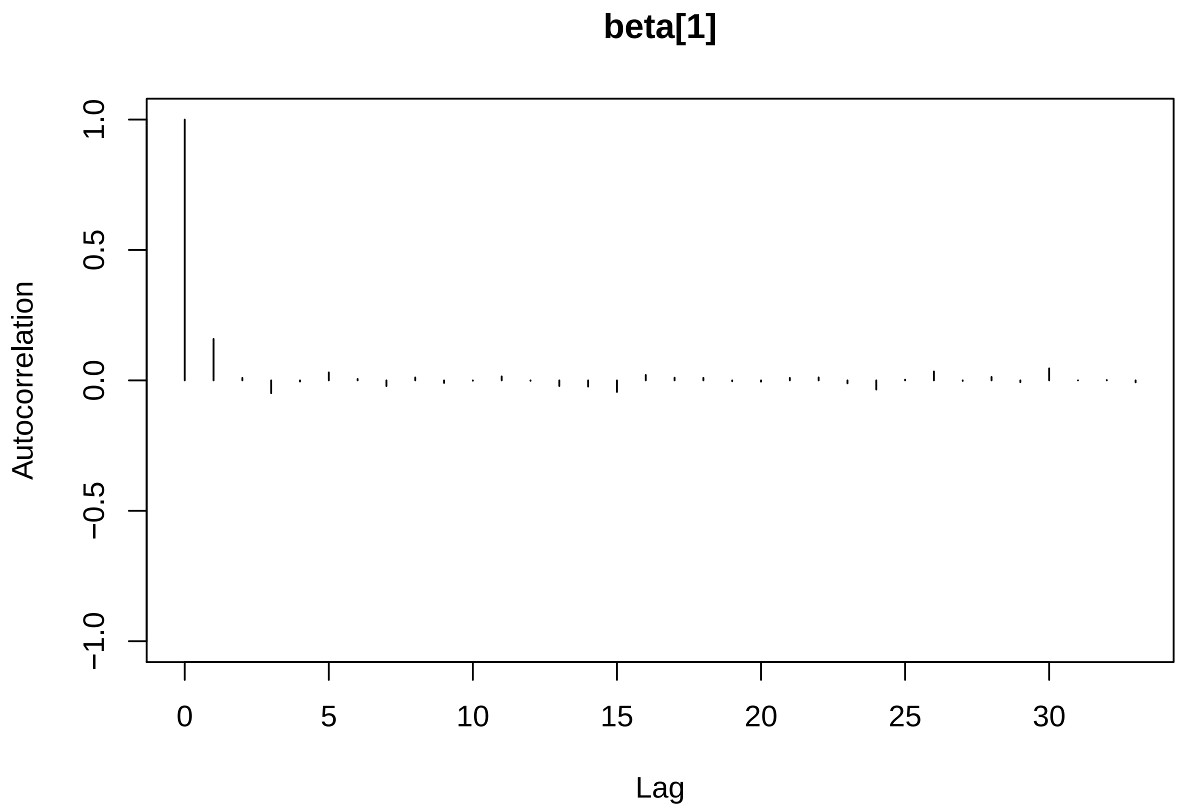
We only plot the intercept for the final report. Set the DEBUG flag to TRUE in order to include all of the coefficients.

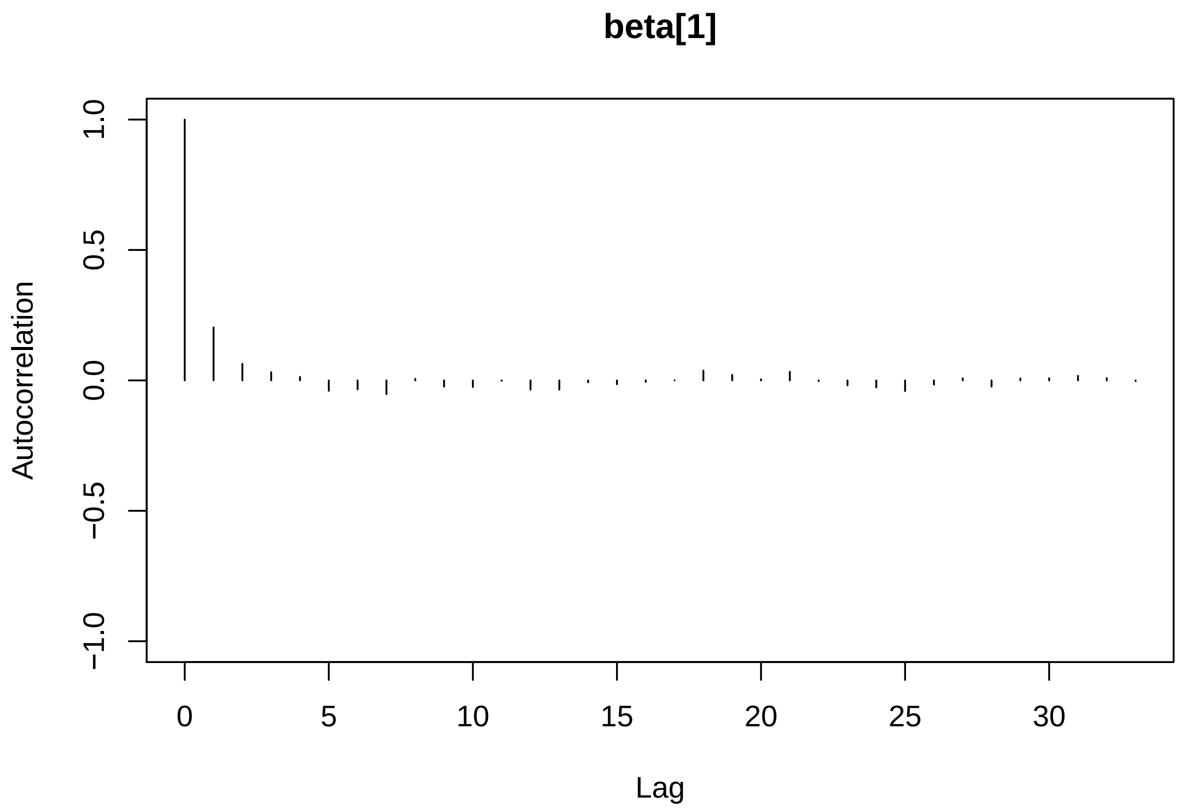
```r
if (DEBUG) {
    for (i in 1:p) {
        samp.coeff.uninformative <- coda.samples(model.carolinas.uninformative,
            variable.names = c(paste("beta[",
                i, "]", sep = "")), n.iter = nSamples,
            progress.bar = "none")
        autocorr.plot(samp.coeff.uninformative)
        gelman.plot(samp.coeff.uninformative)
```

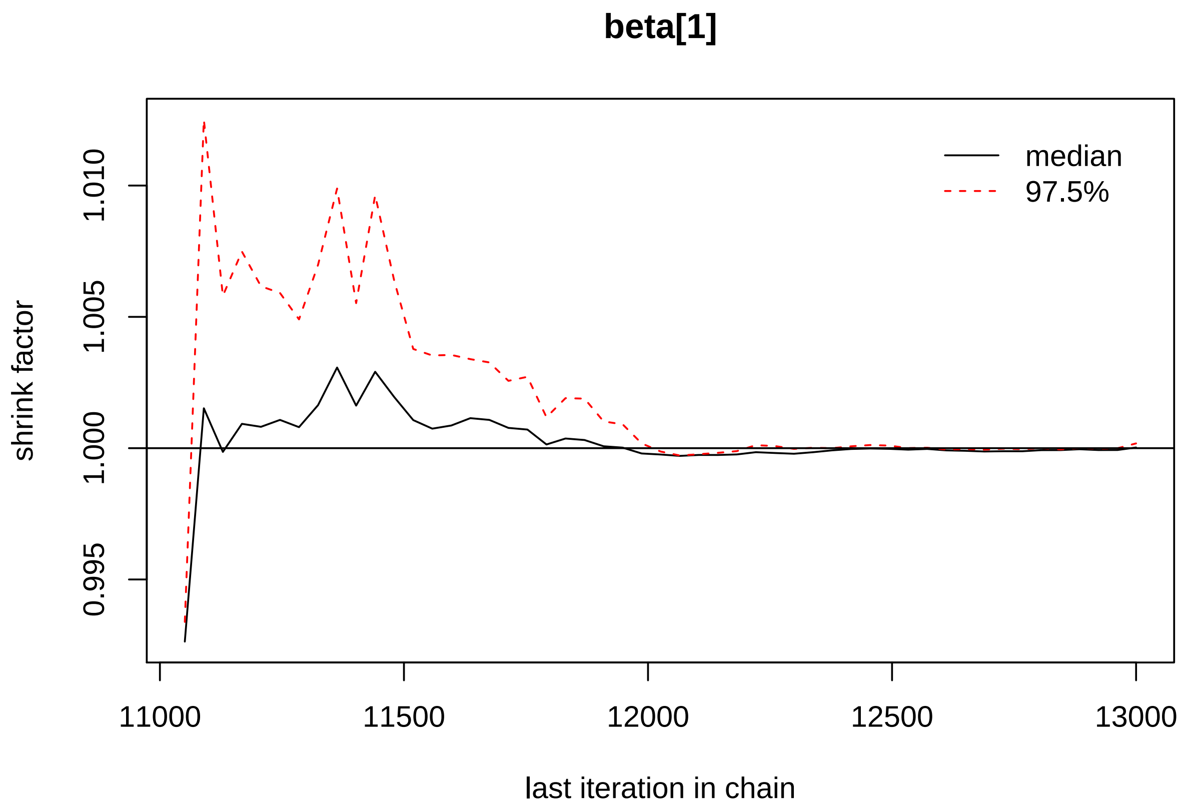
```
        plot(samp.coeff.uninformative)
    }
    samp.coeff.uninformative <- coda.samples(model.carolinas.uninformative,
        variable.names = "intercept", n.iter = nSamples,
        progress.bar = "none")
    autocorr.plot(samp.coeff.uninformative)
    gelman.plot(samp.coeff.uninformative)
    plot(samp.coeff.uninformative)
} else {
    samp.coeff.uninformative <- coda.samples(model.carolinas.uninformative,
        variable.names = "intercept", n.iter = nSamples,
        progress.bar = "none")
    autocorr.plot(samp.coeff.uninformative)
    gelman.plot(samp.coeff.uninformative)
    plot(samp.coeff.uninformative)
}
```

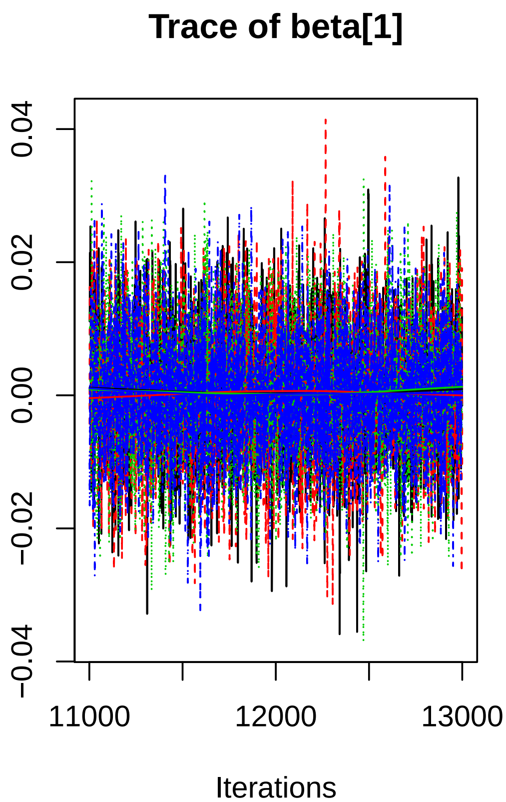



beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

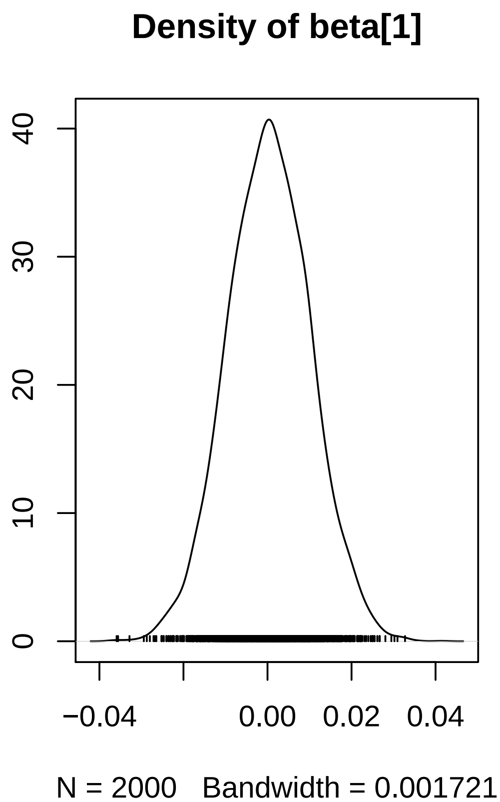

beta[1]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[1]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

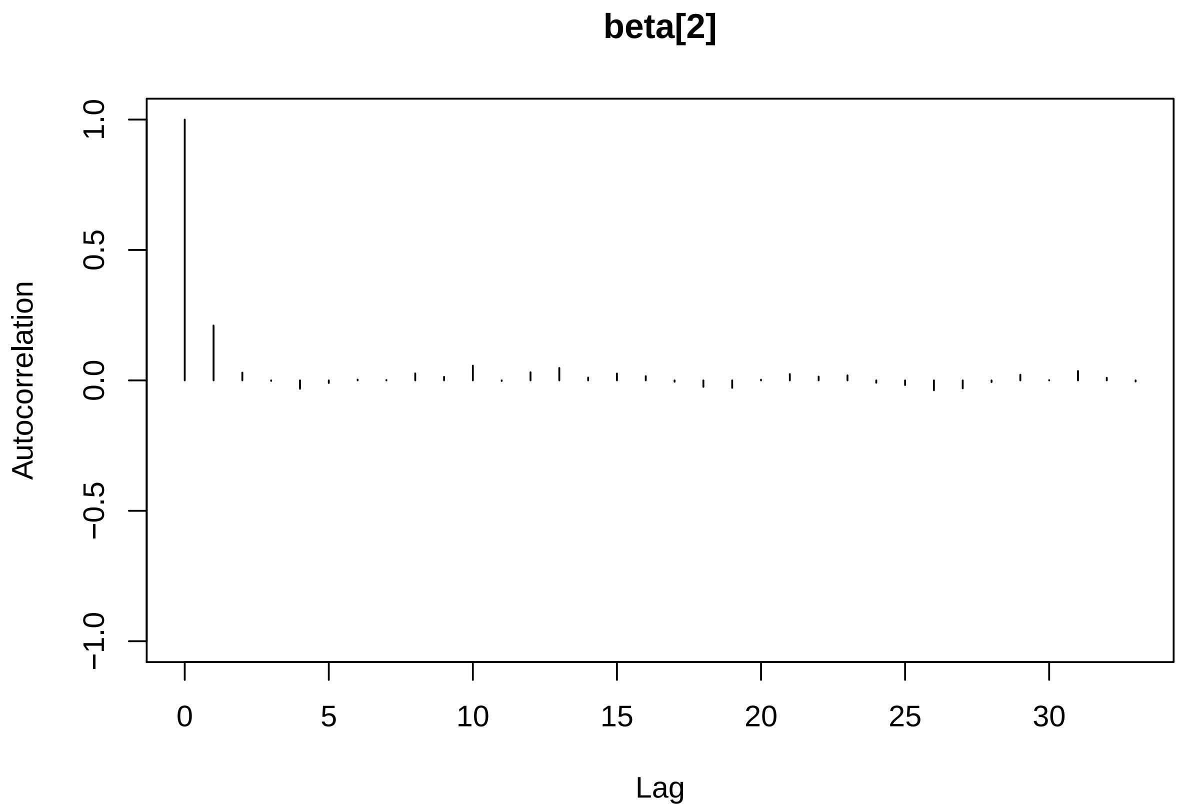

beta[1]
median
97.5%
shrink factor
1.010
1.005
1.000
0.995
11000
11500
12000
12500
13000
last iteration in chain

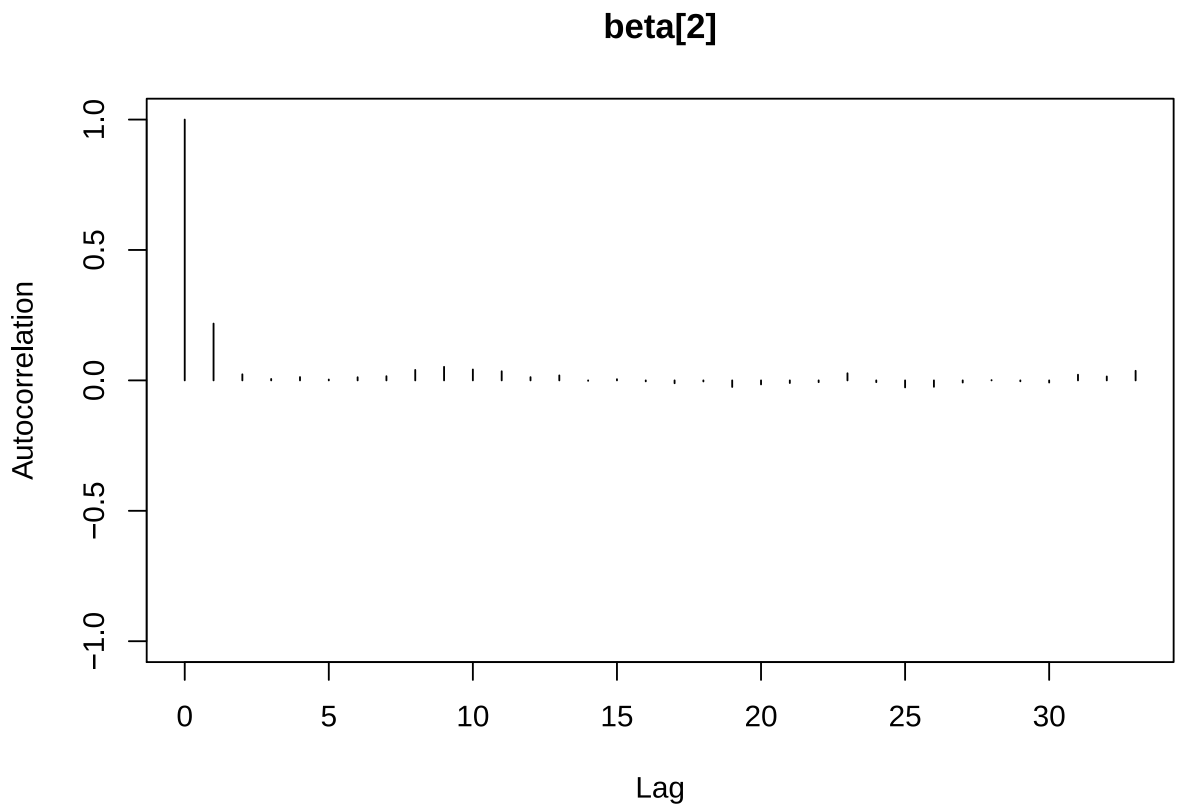


Trace of beta[1]
0.04
0.02
0.00
−0.02
−0.04
11000
12000
13000
Iterations

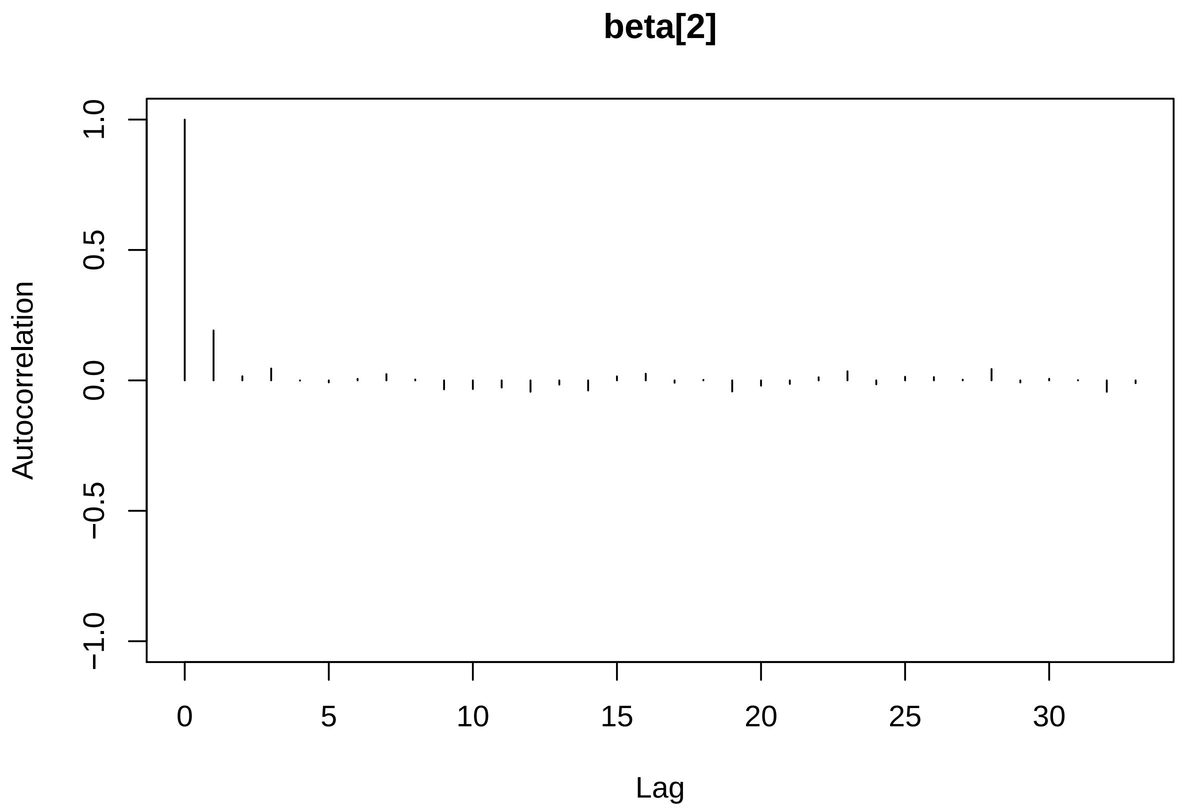


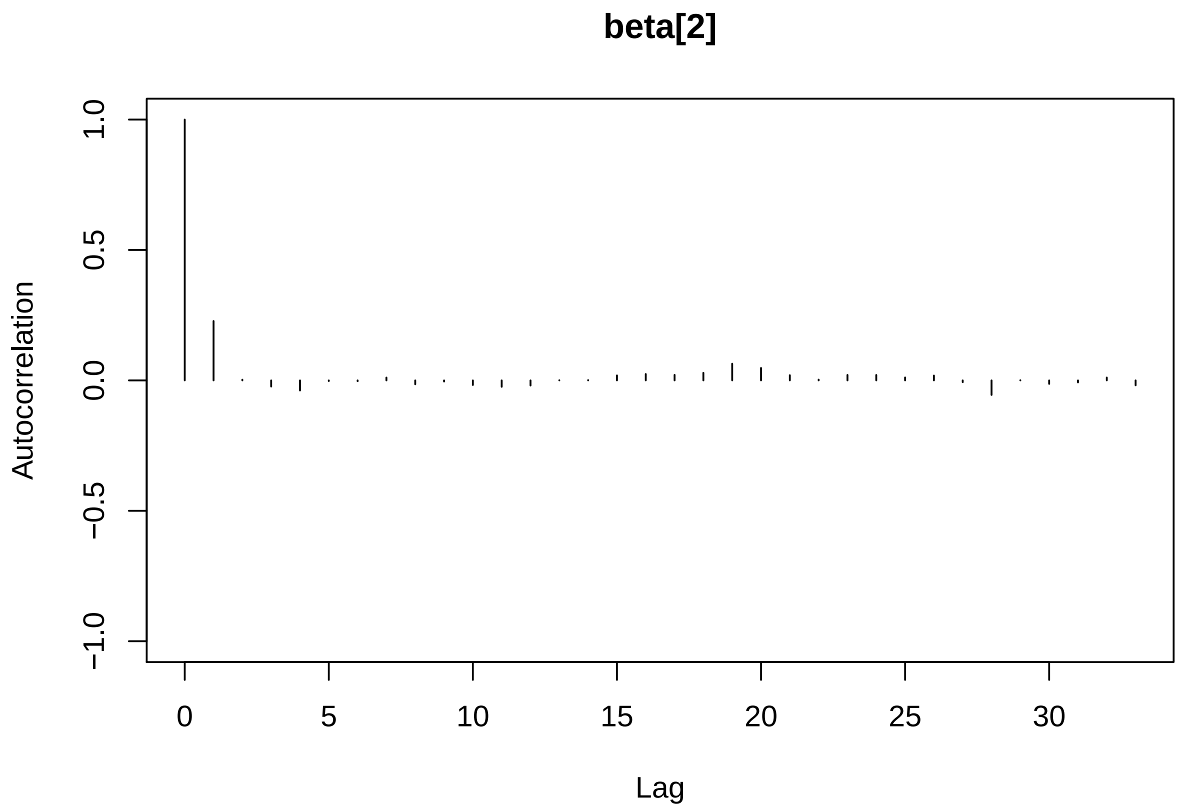
Density of beta[1]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001721


beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag



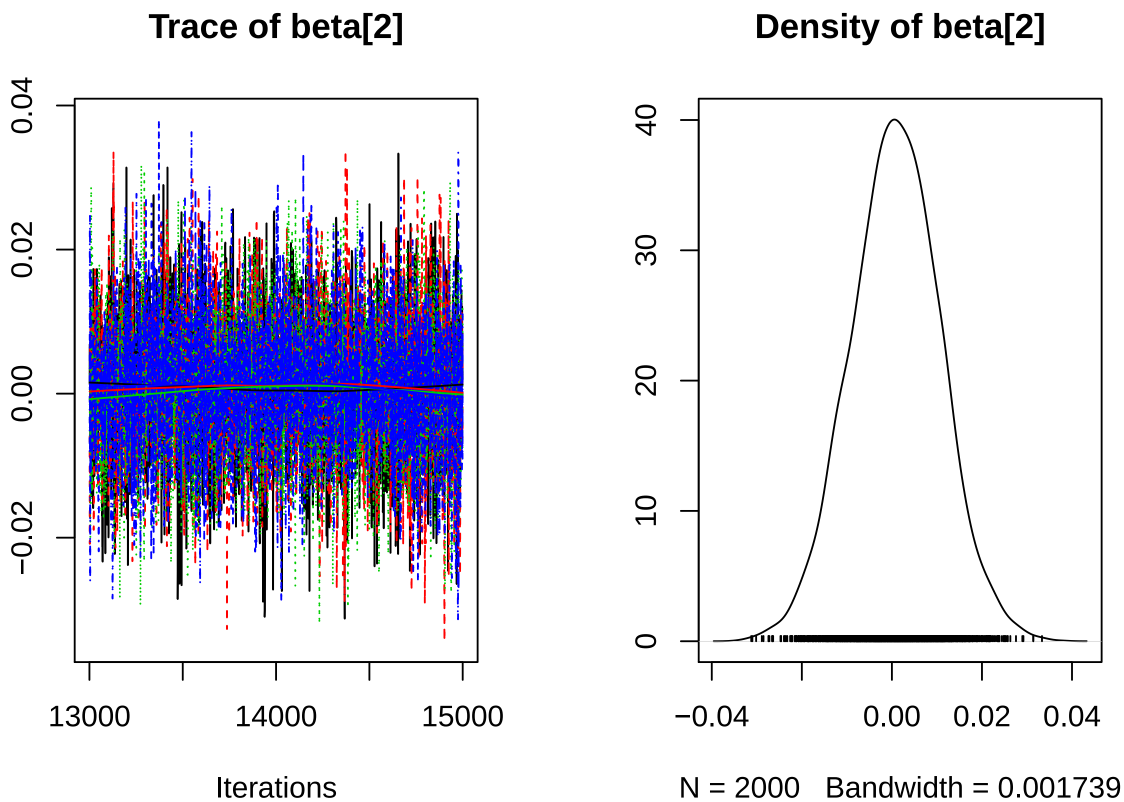
beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

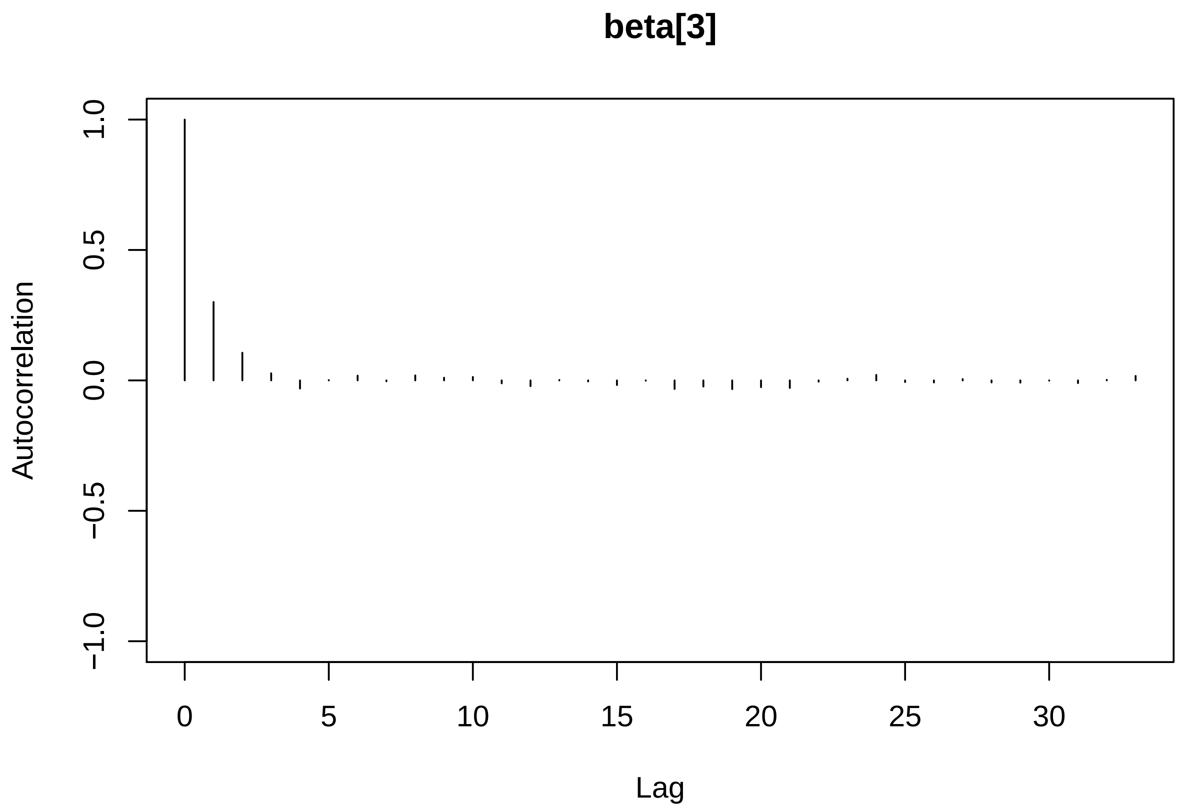

beta[2]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

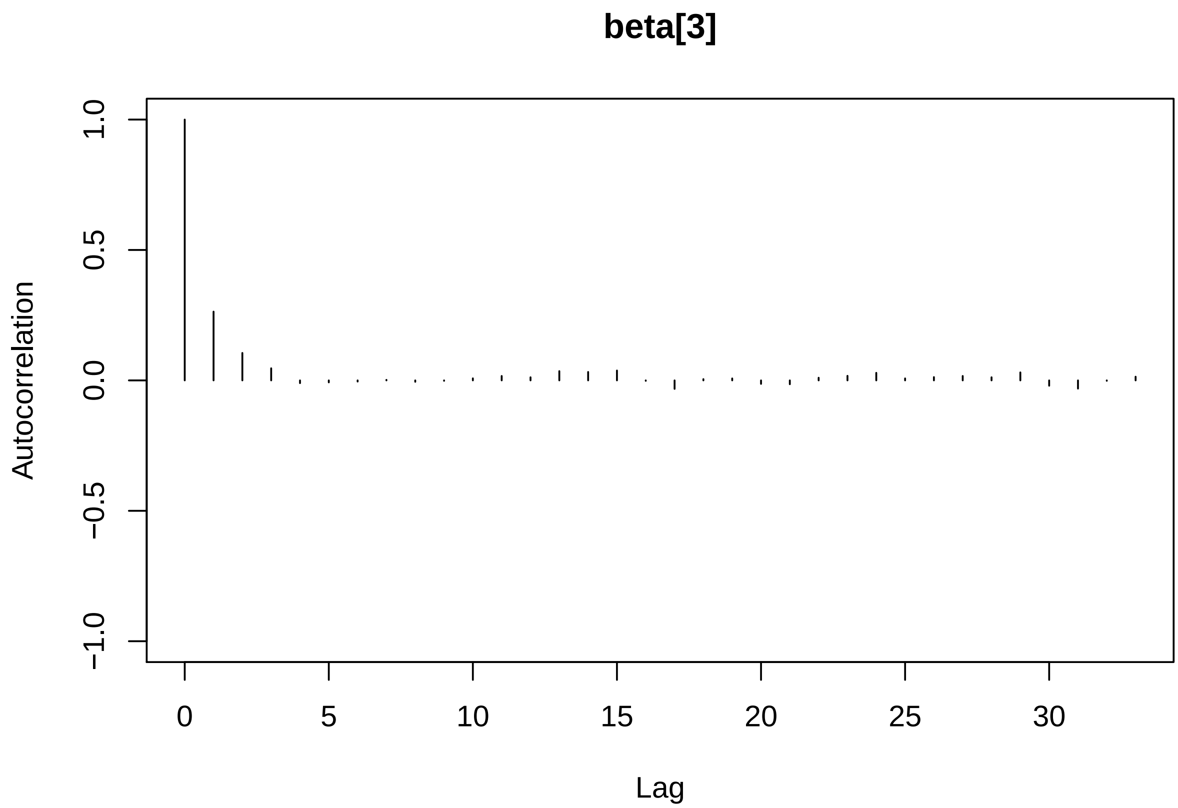

beta[2]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

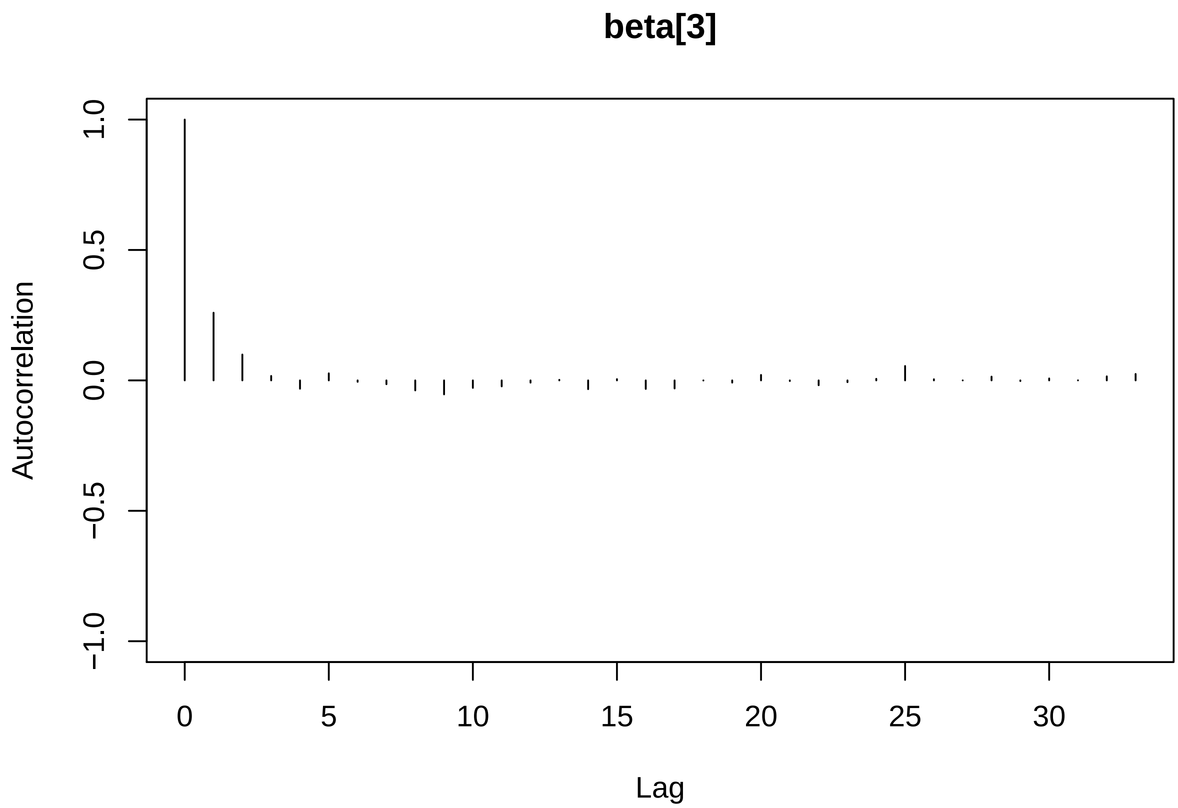
beta[2]

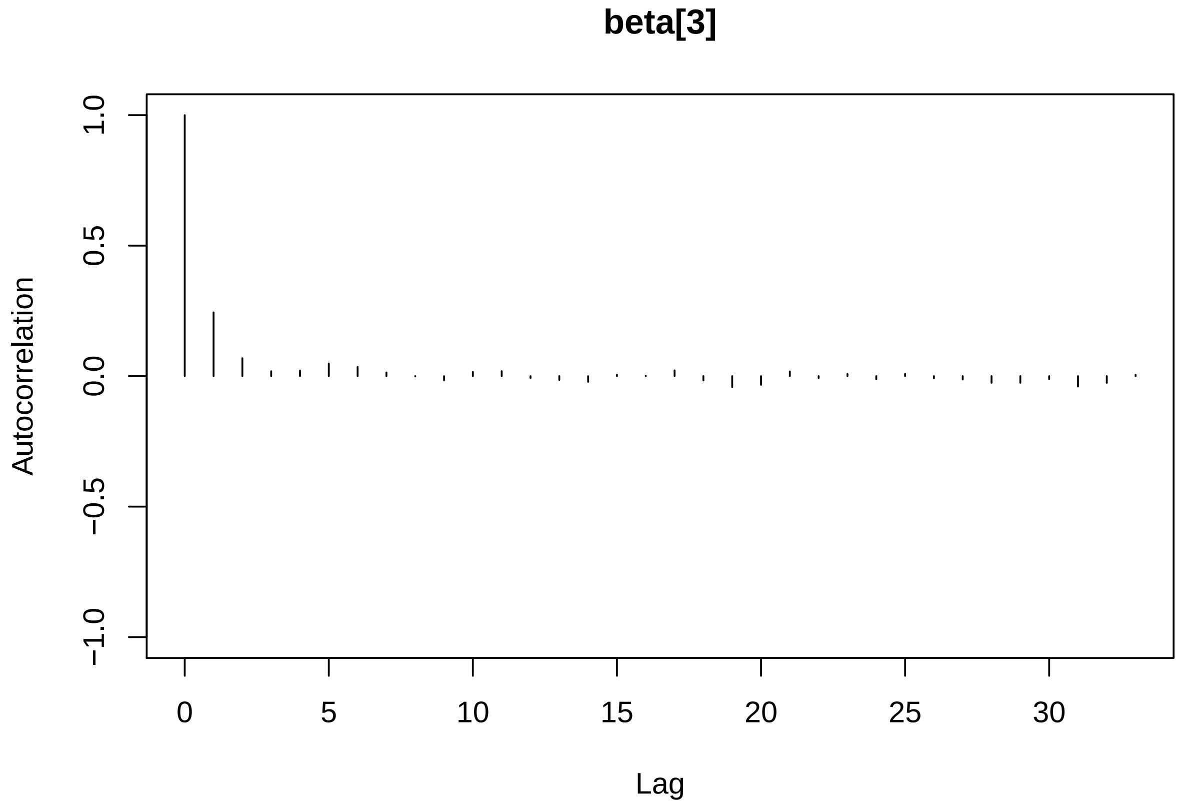
median
97.5%
shrink factor
1.000
1.005
1.010
1.015
13000
13500
14000
14500
15000
last iteration in chain

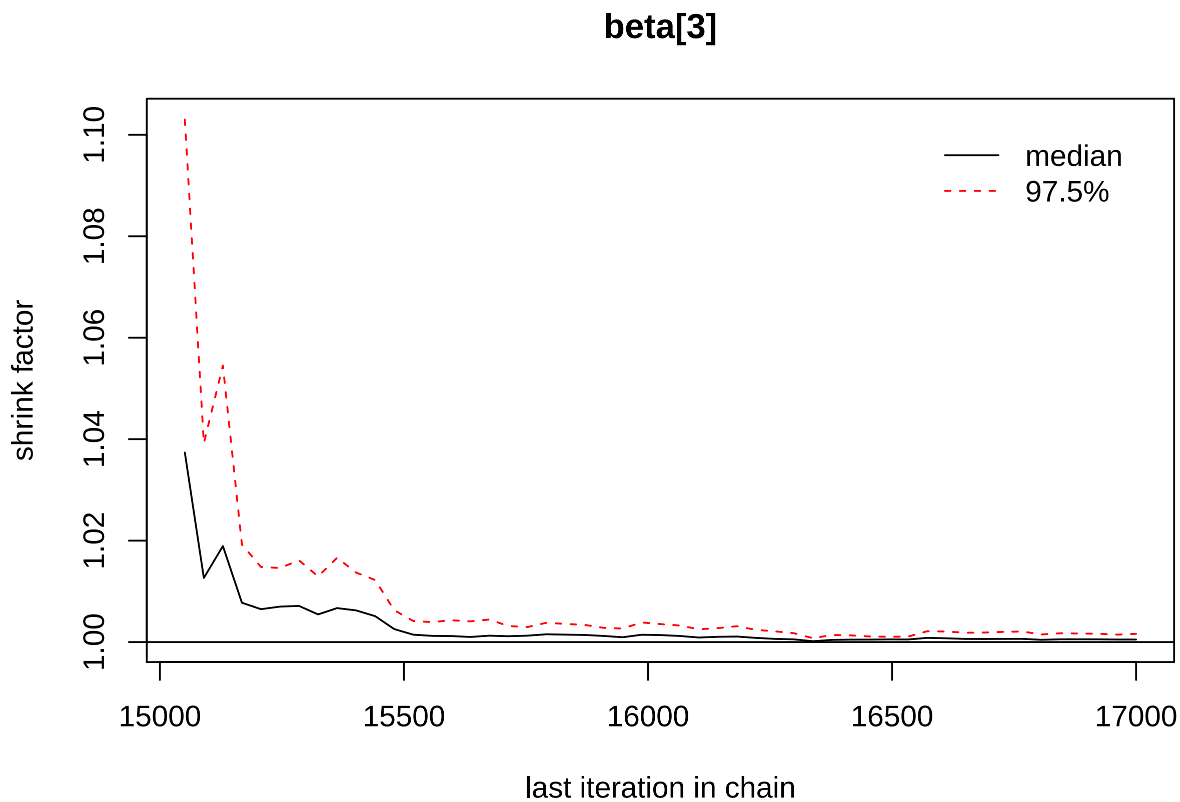

Trace of beta[2]
0.04
0.02
0.00
−0.02
13000
14000
15000
Iterations
Density of beta[2]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001739

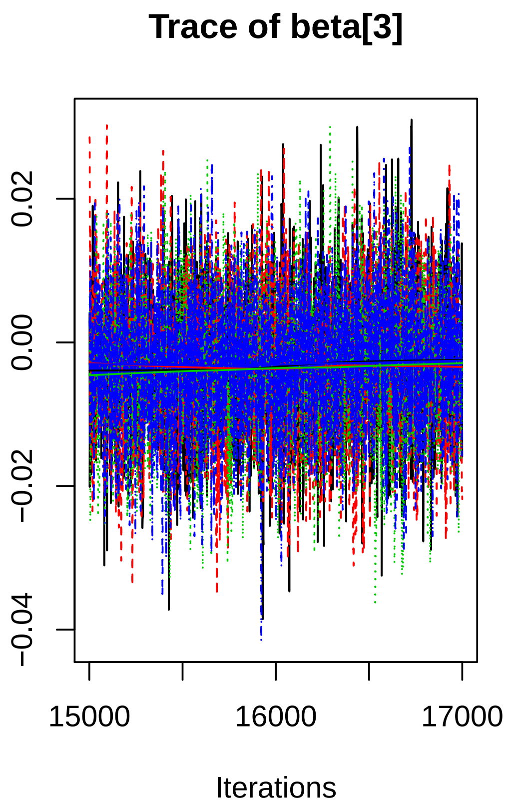

beta[3]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[3]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

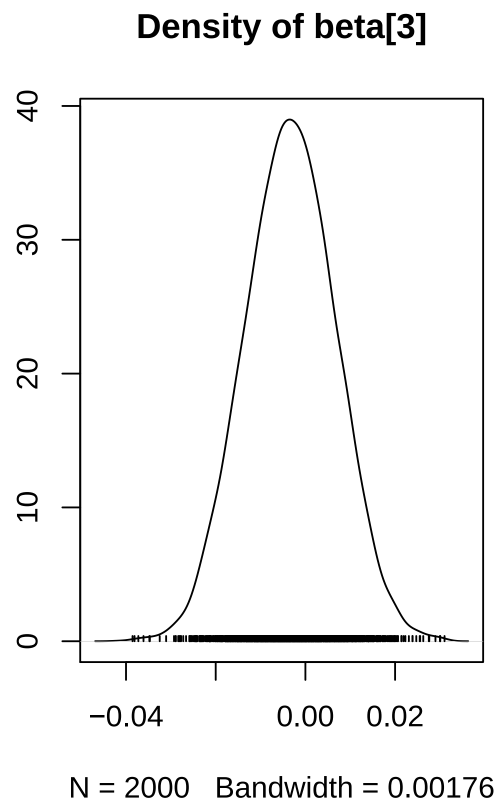

beta[3]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[3]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

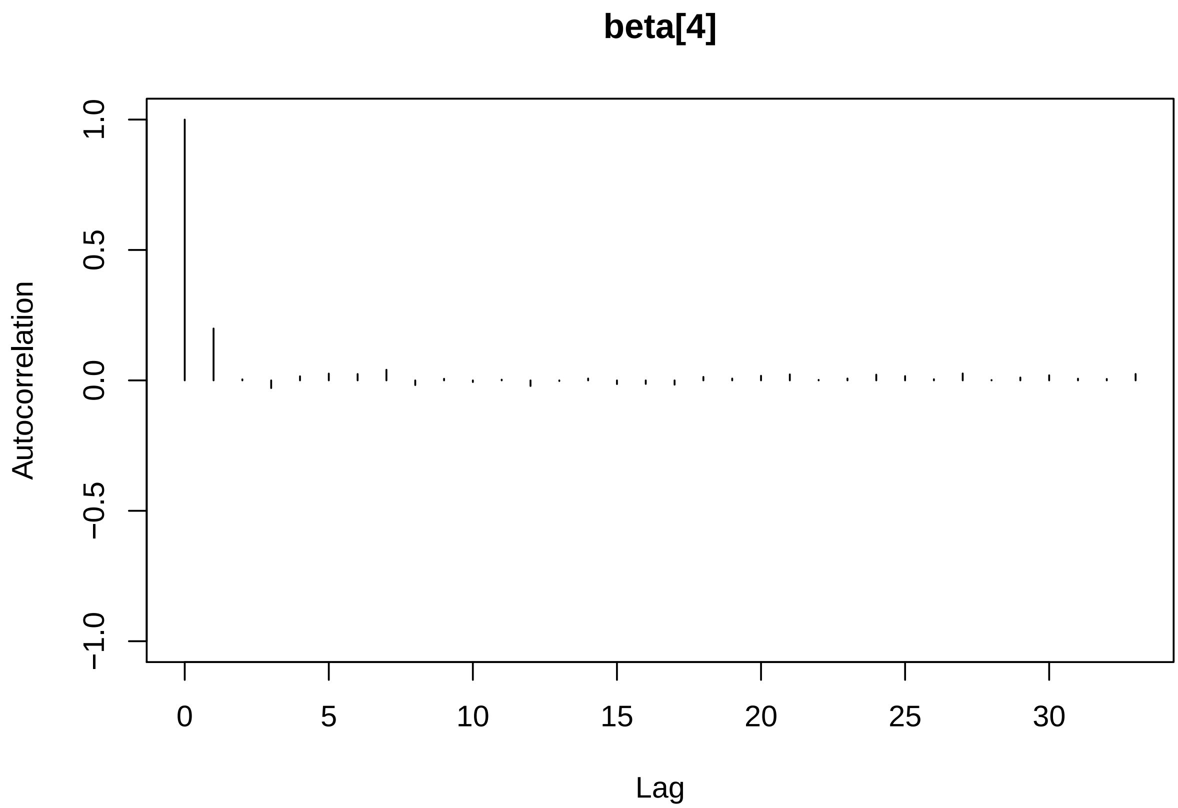

beta[3]
shrink factor
1.00
1.02
1.04
1.06
1.08
1.10
median
97.5%
15000
15500
16000
16500
17000
last iteration in chain

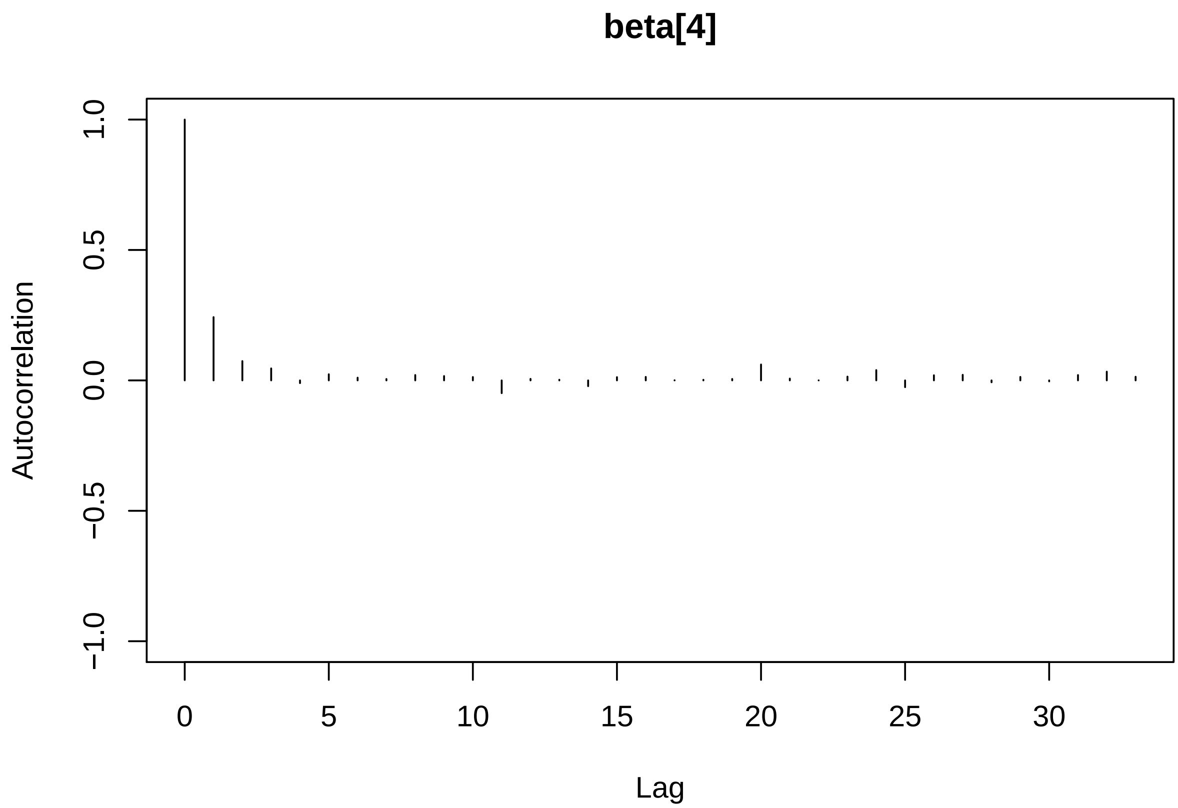


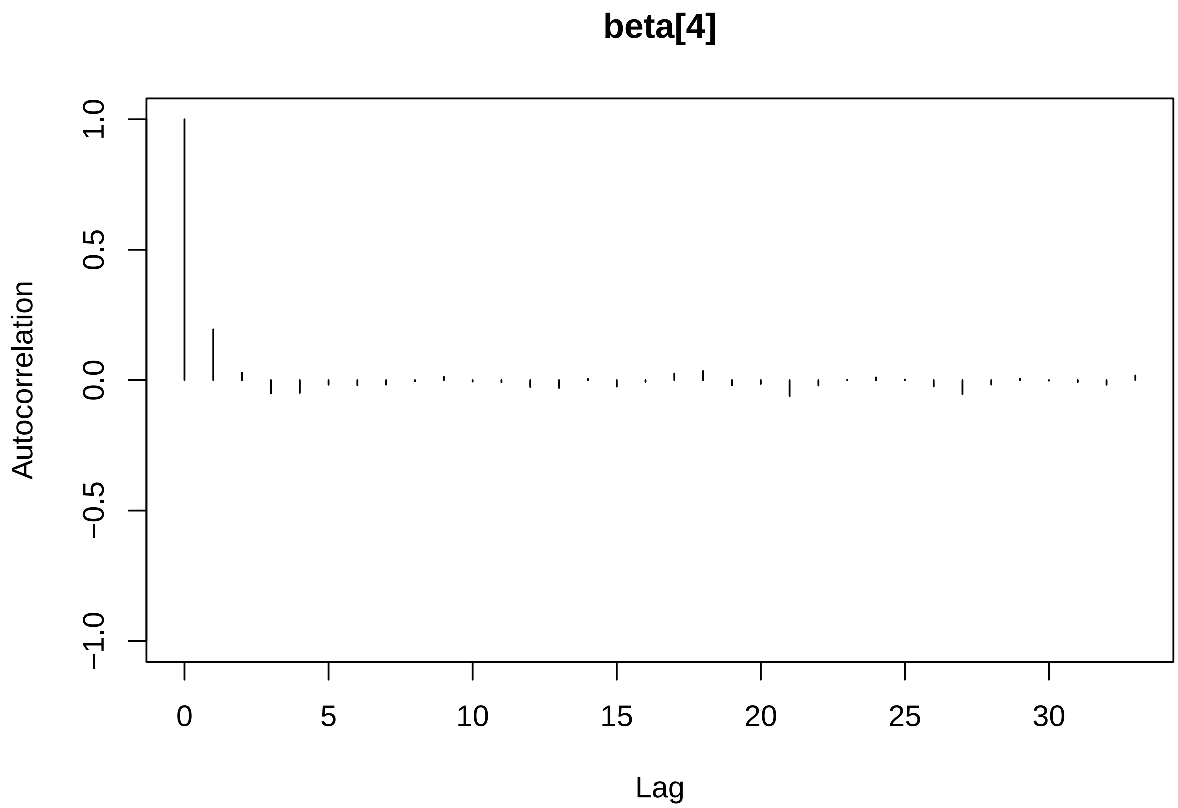
Trace of beta[3]
0.02
0.00
−0.02
−0.04
15000
16000
17000
Iterations

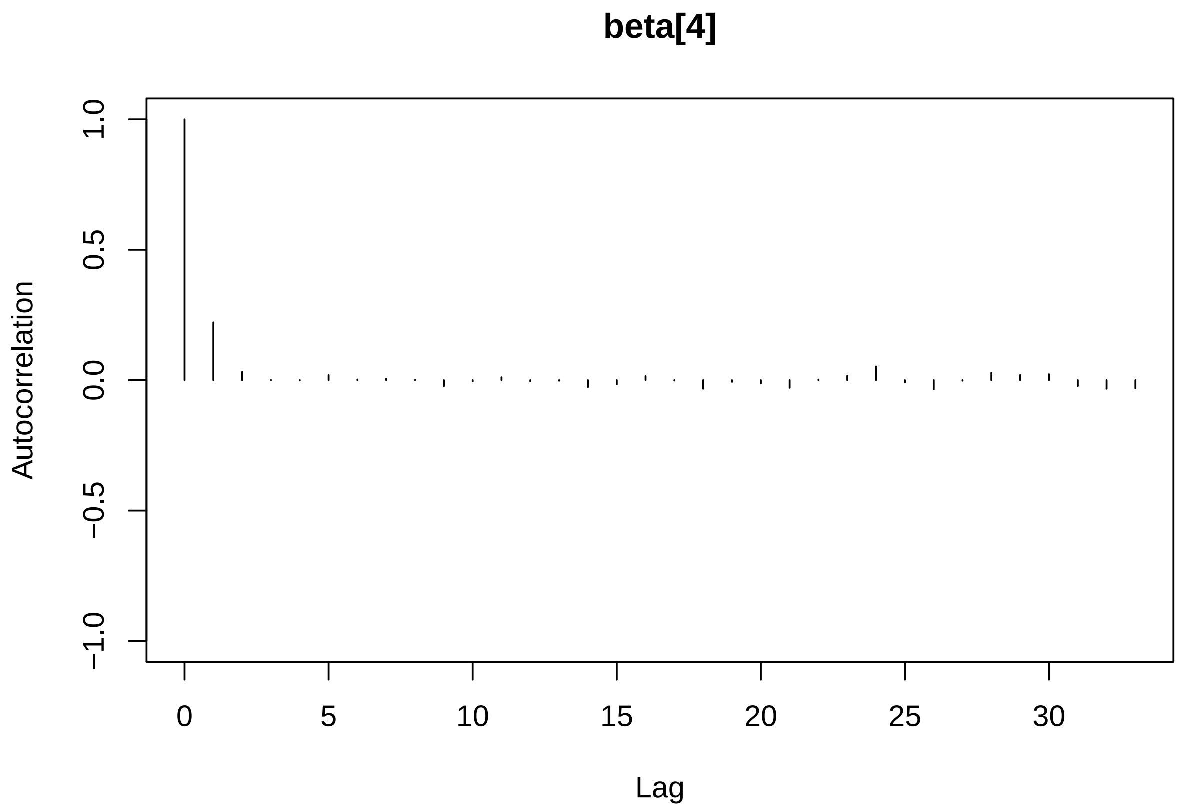


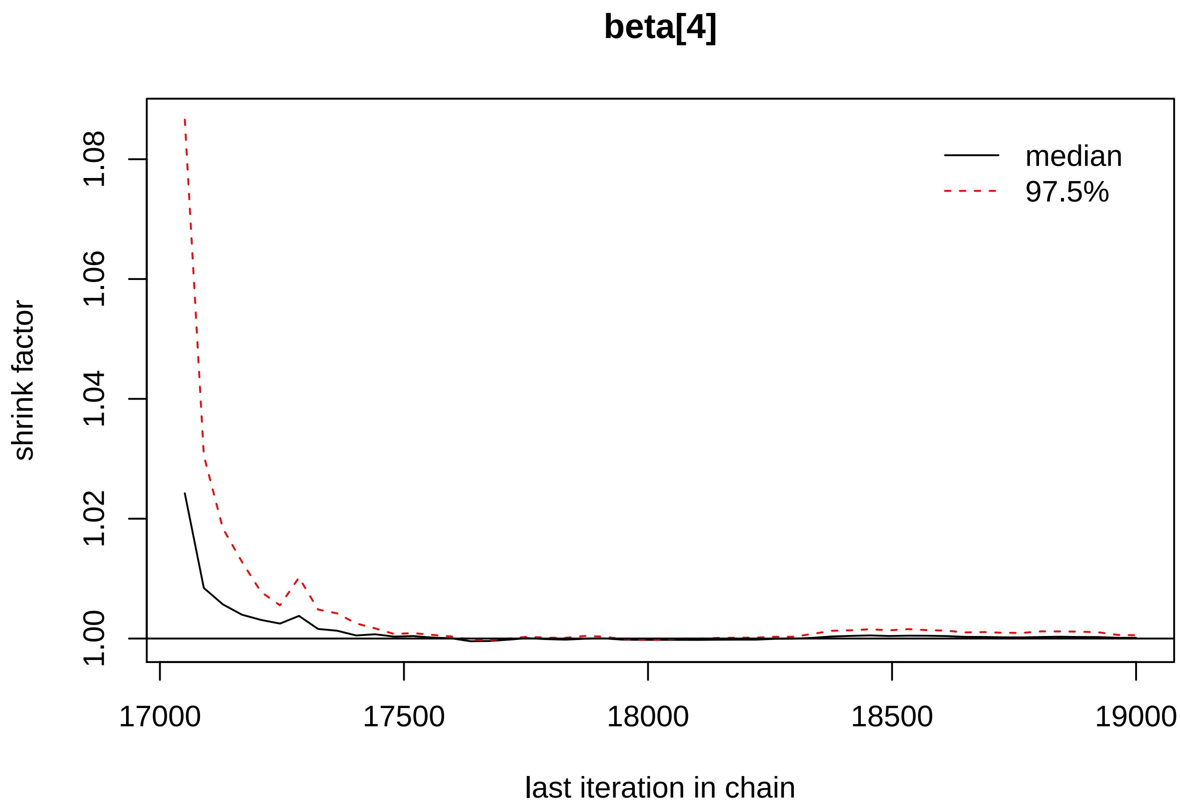
Density of beta[3]
40
30
20
10
0
−0.04
0.00
0.02
N = 2000 Bandwidth = 0.00176

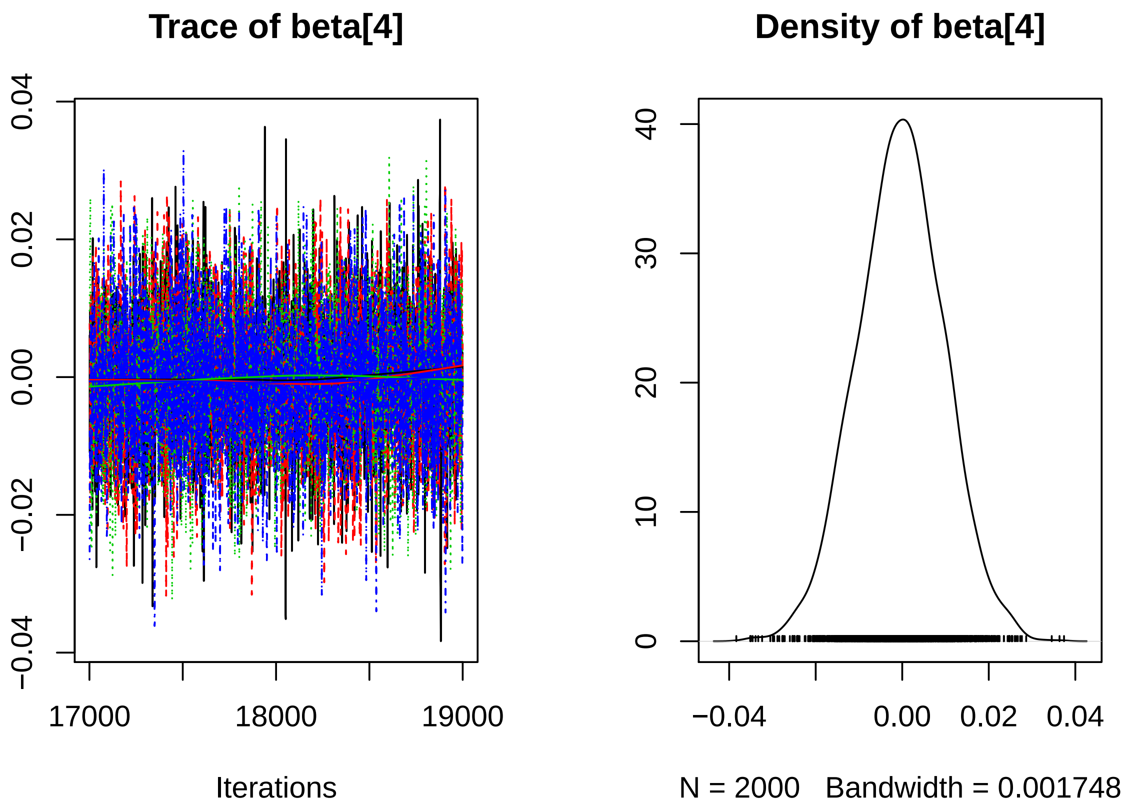

beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

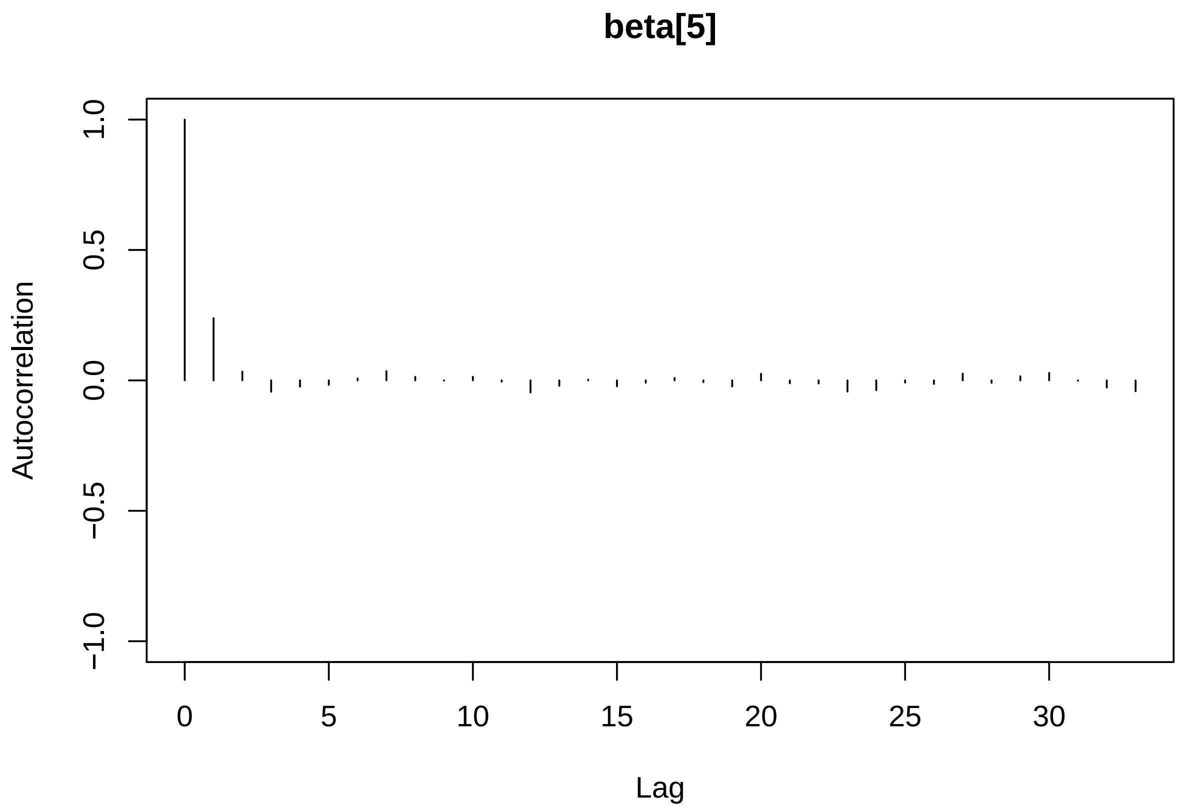

beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

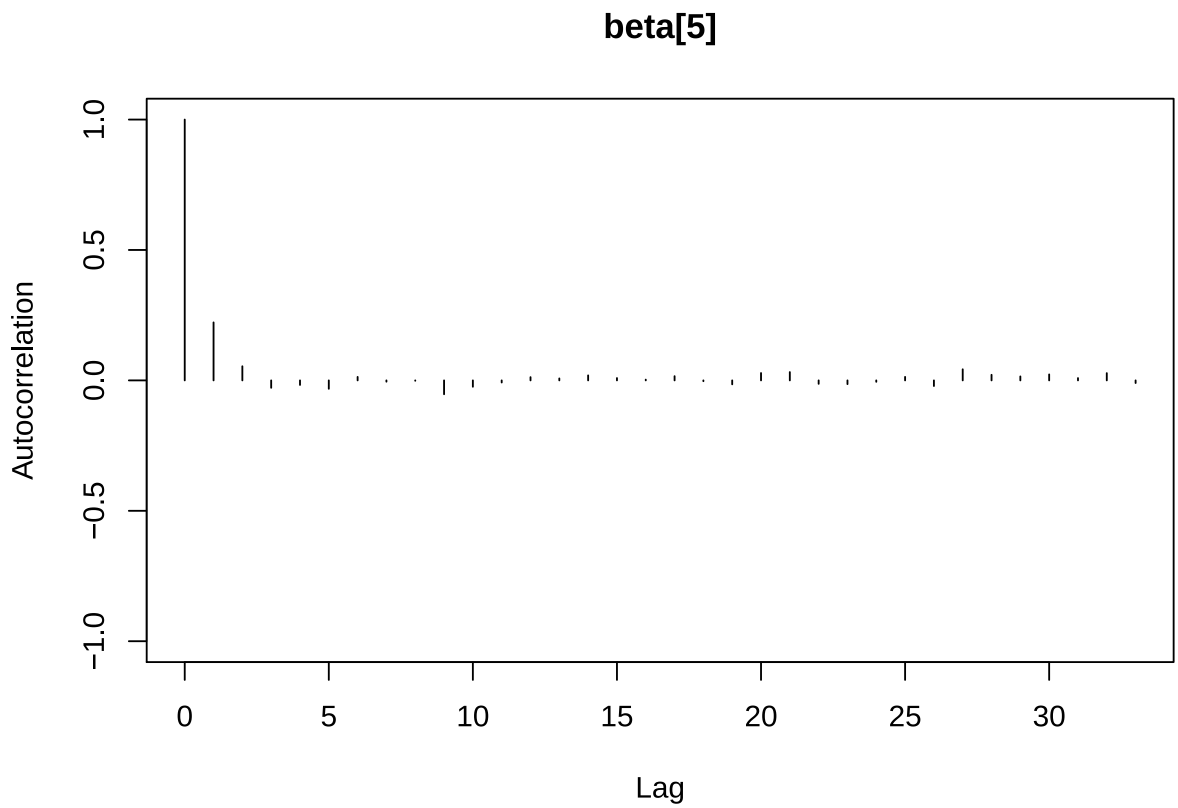

beta[4]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

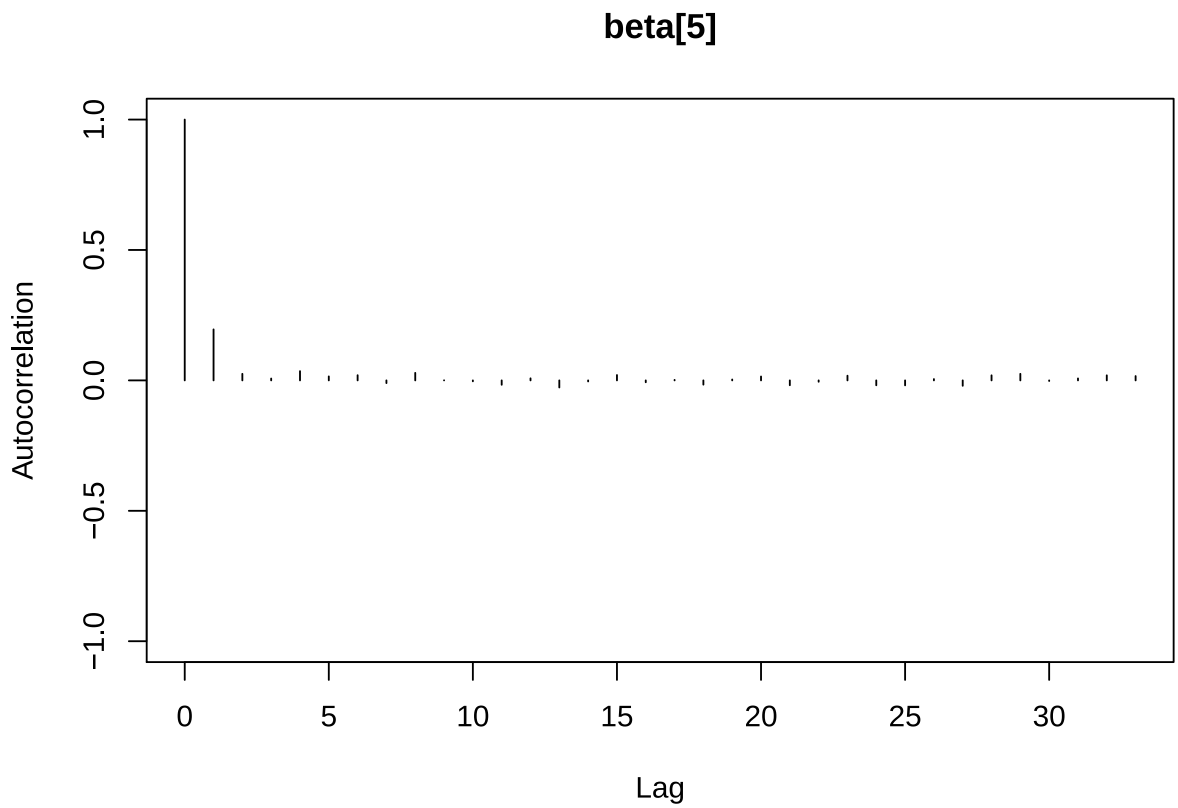

beta[4]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
17000
17500
18000
18500
19000
last iteration in chain

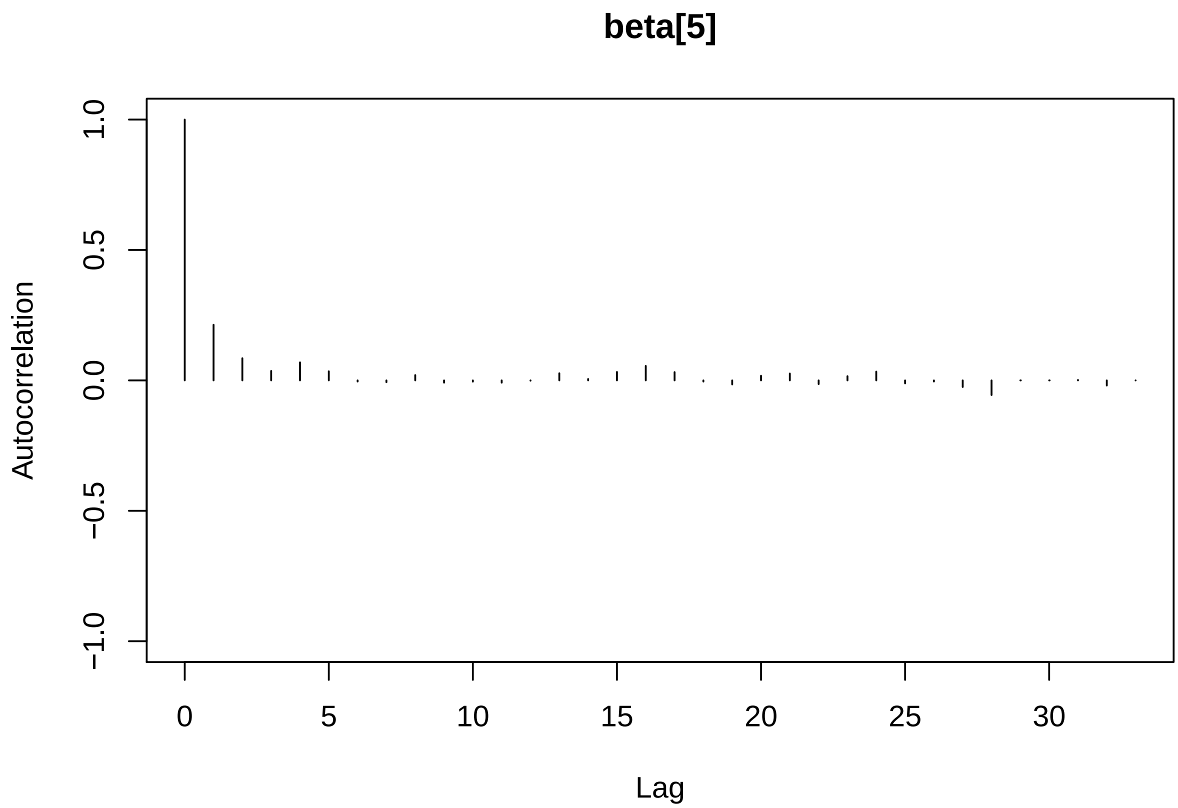

Trace of beta[4]
0.04
0.02
0.00
−0.02
−0.04
17000
18000
19000
Iterations
Density of beta[4]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000   Bandwidth = 0.001748

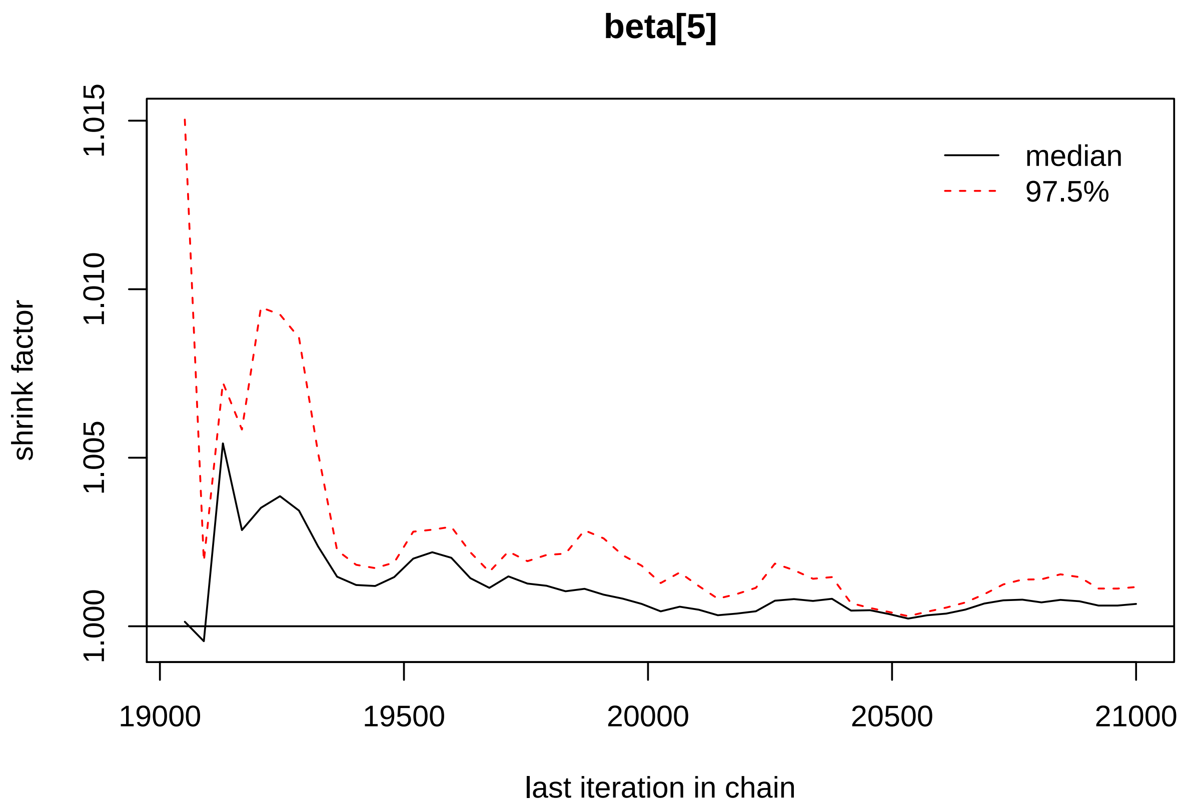


beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

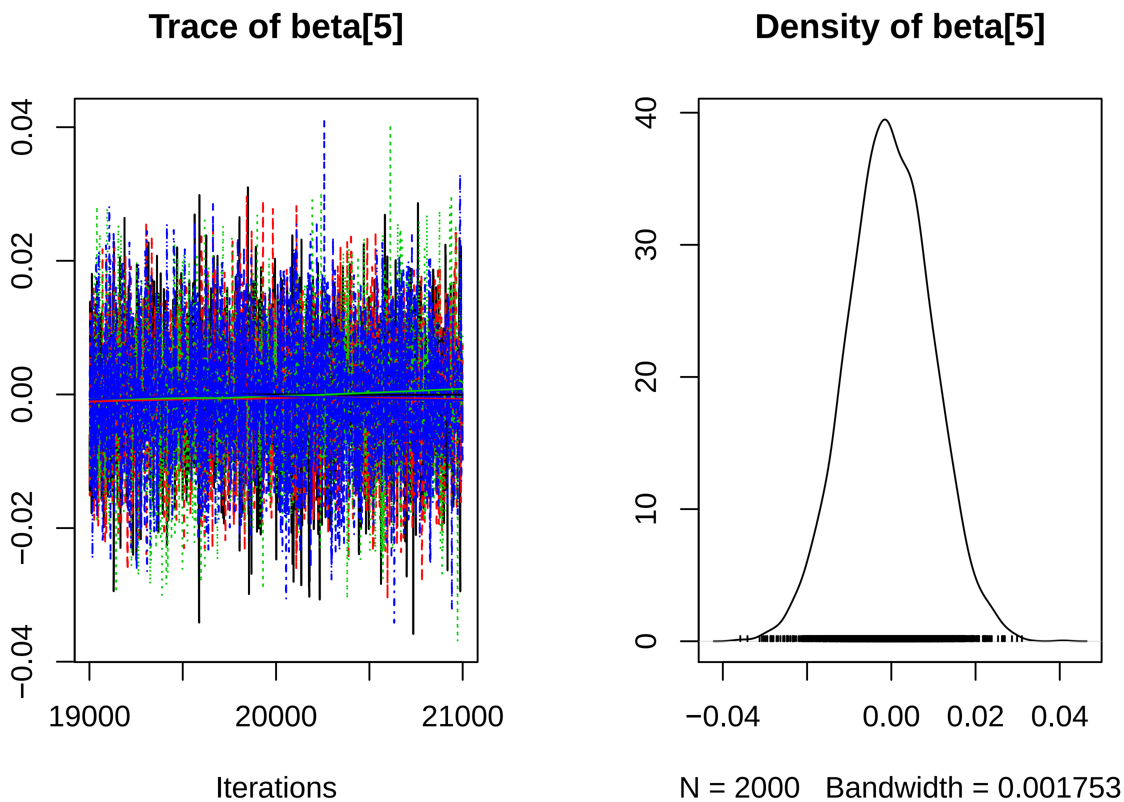

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

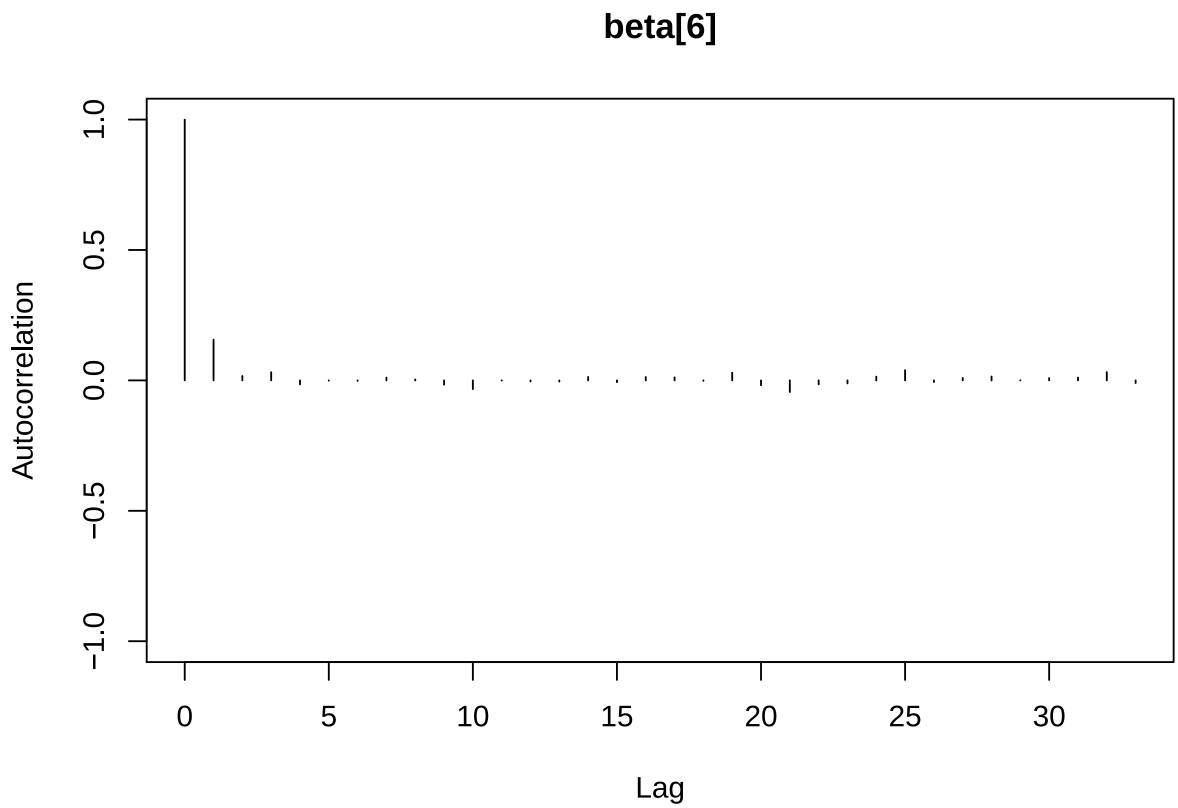

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

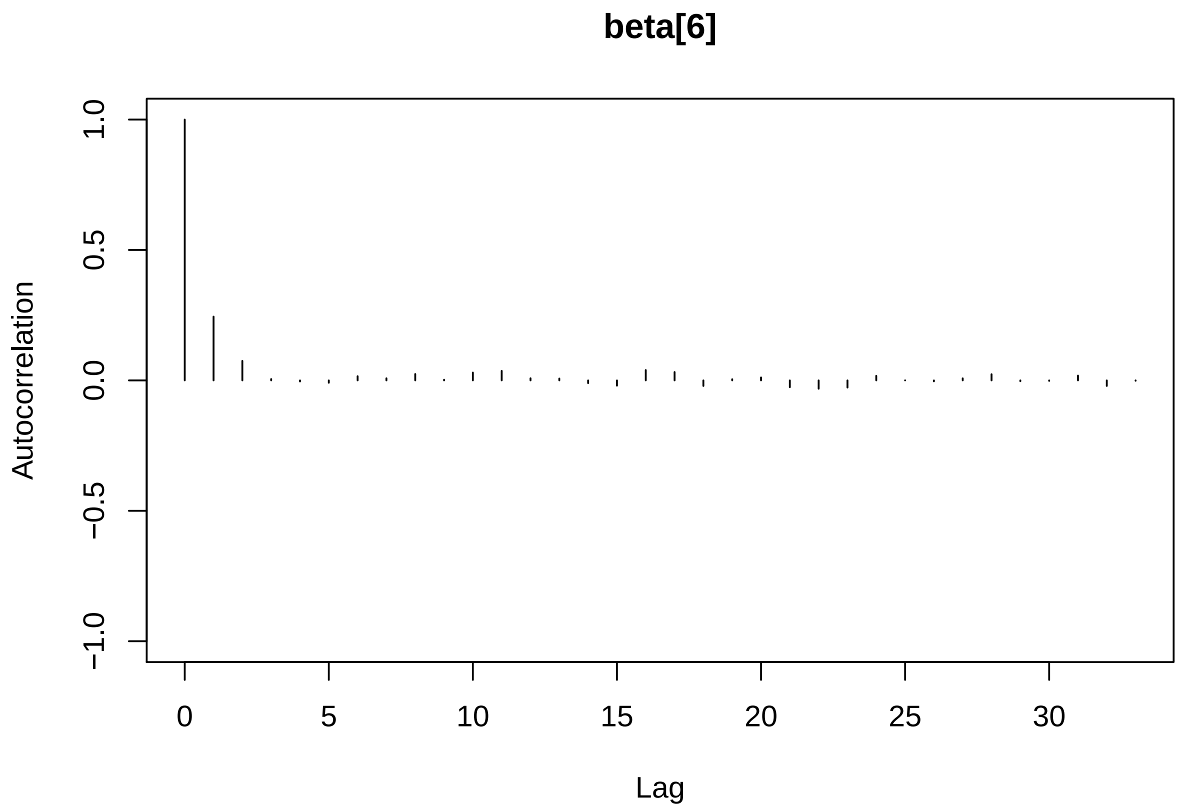

beta[5]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

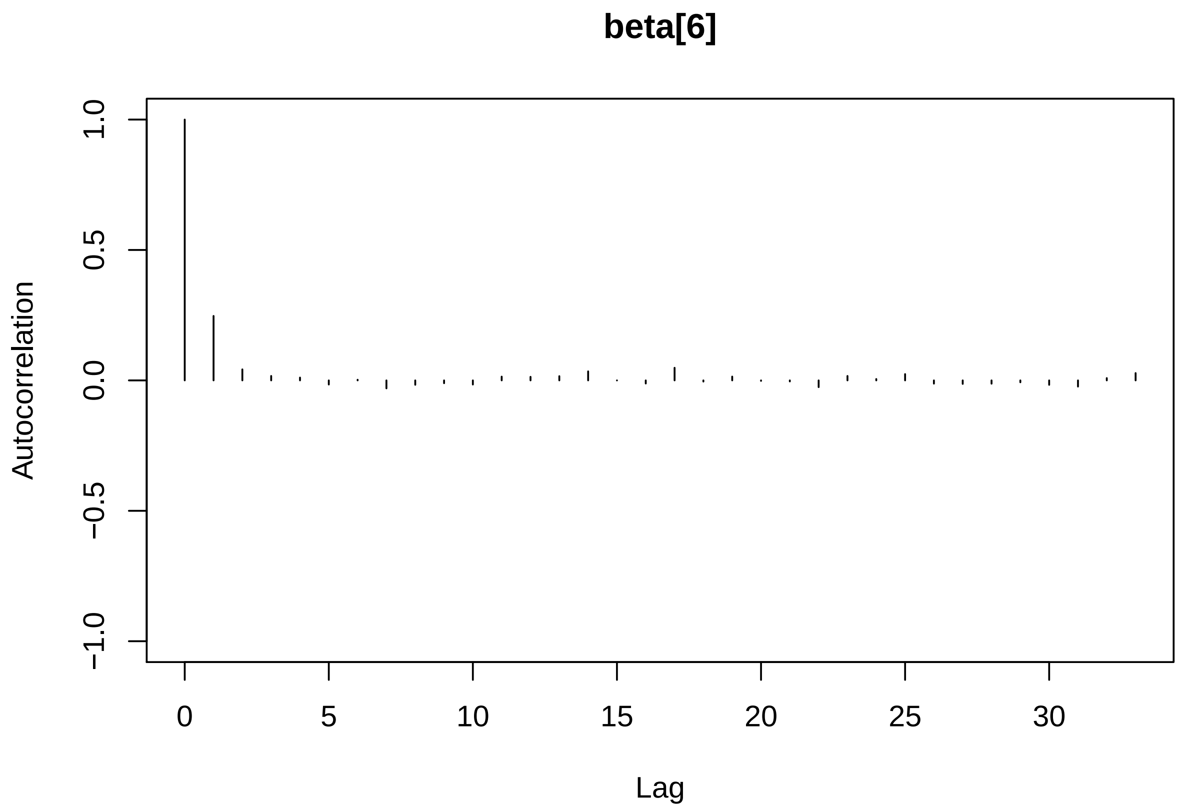


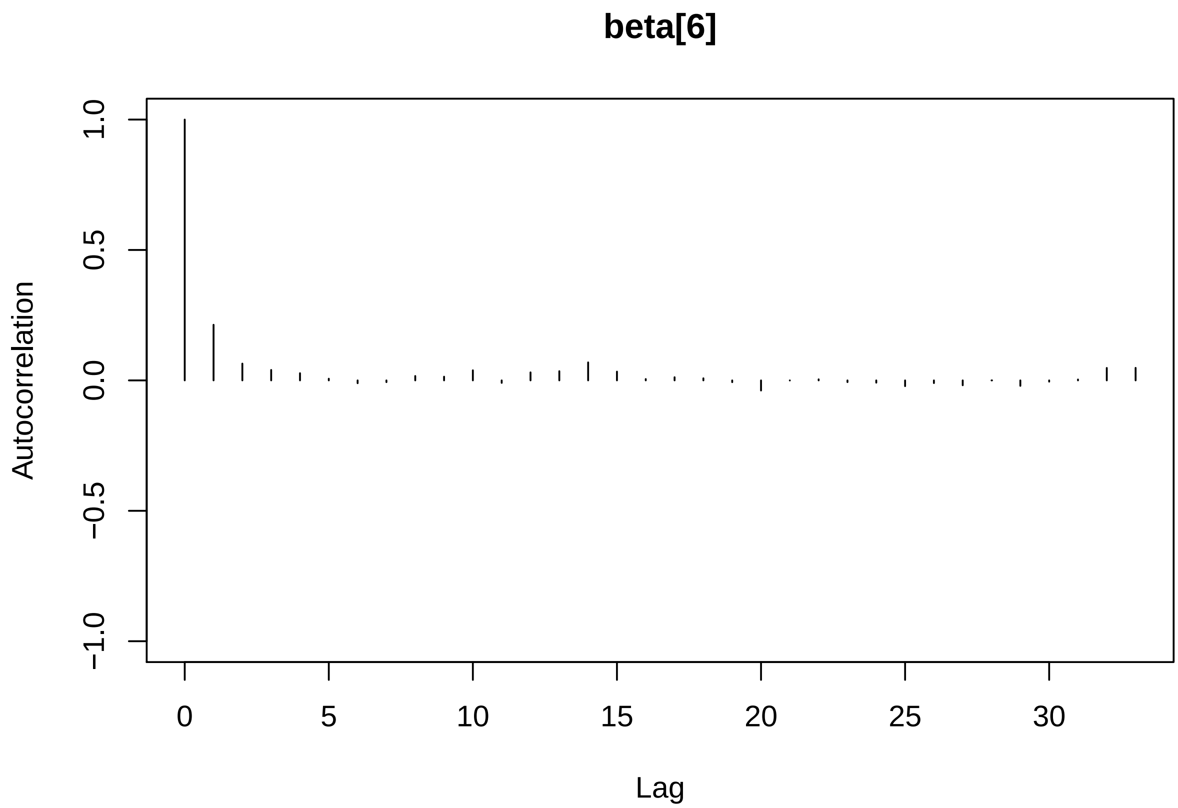
beta[5]
shrink factor
1.000
1.005
1.010
1.015
19000
19500
20000
20500
21000
last iteration in chain
median
97.5%

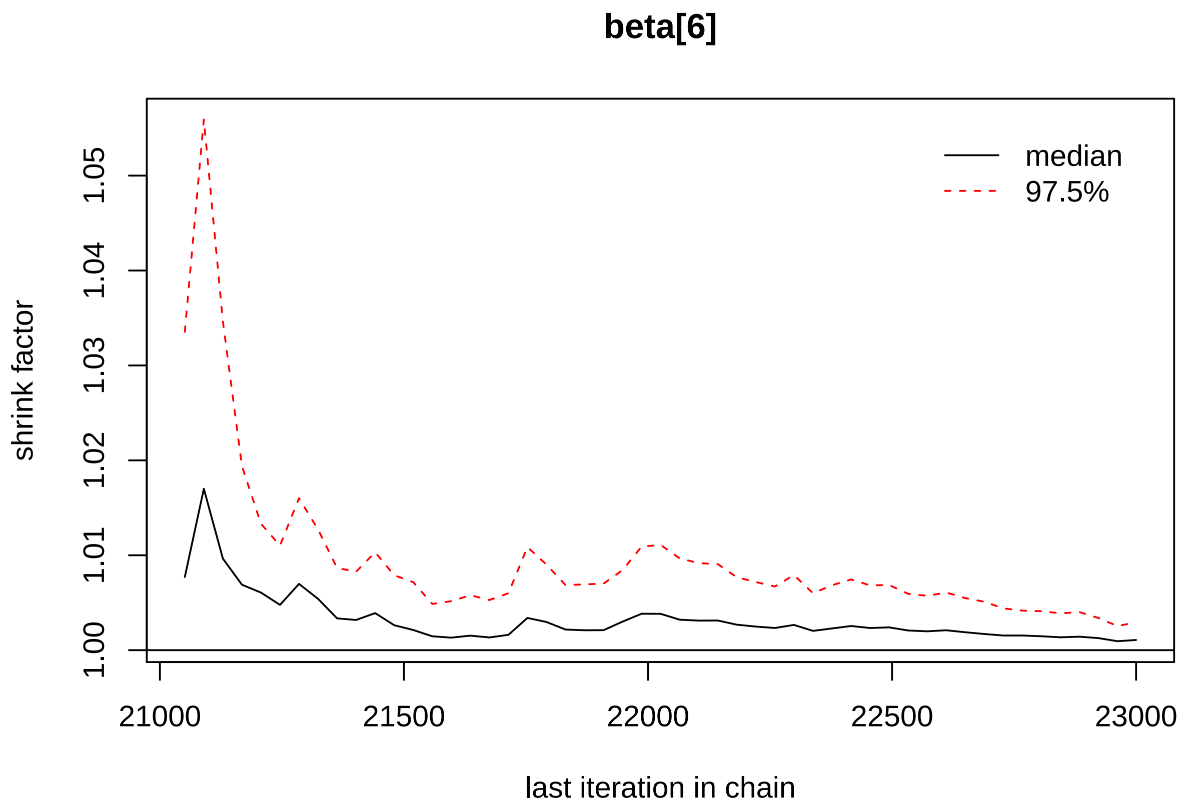

Trace of beta[5]
0.04
0.02
0.00
−0.02
−0.04
19000
20000
21000
Iterations
Density of beta[5]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001753


beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

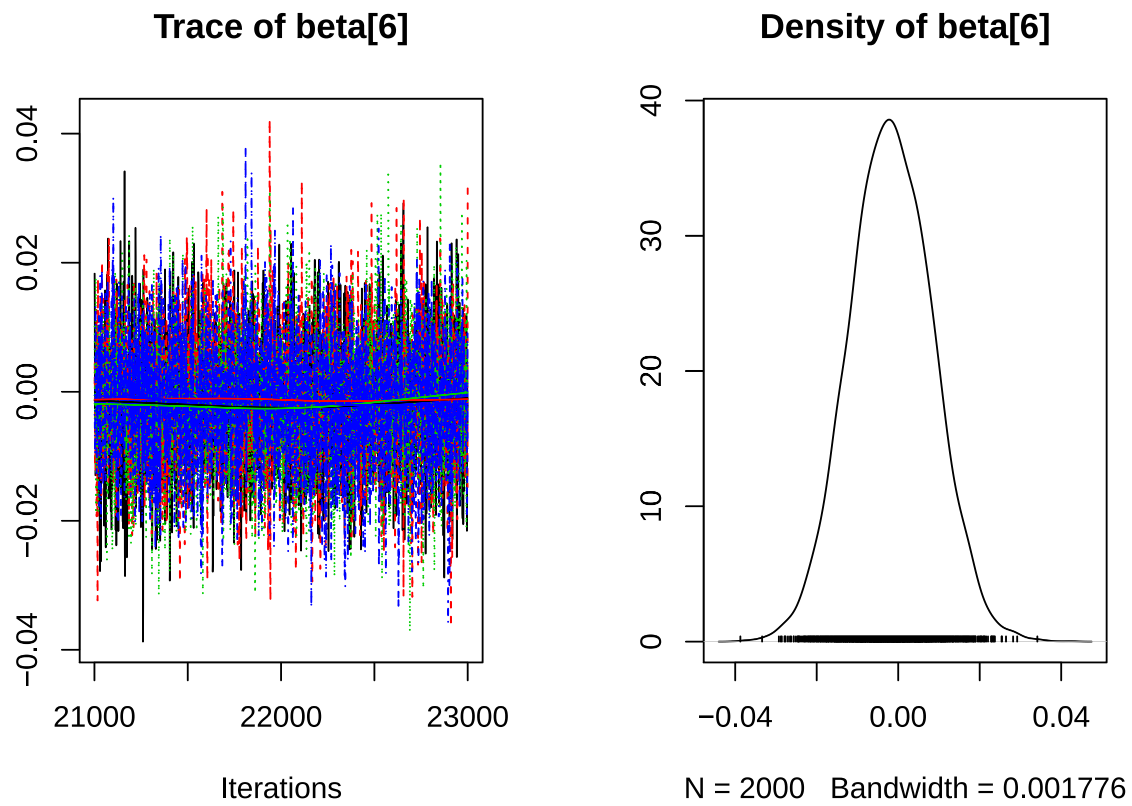

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

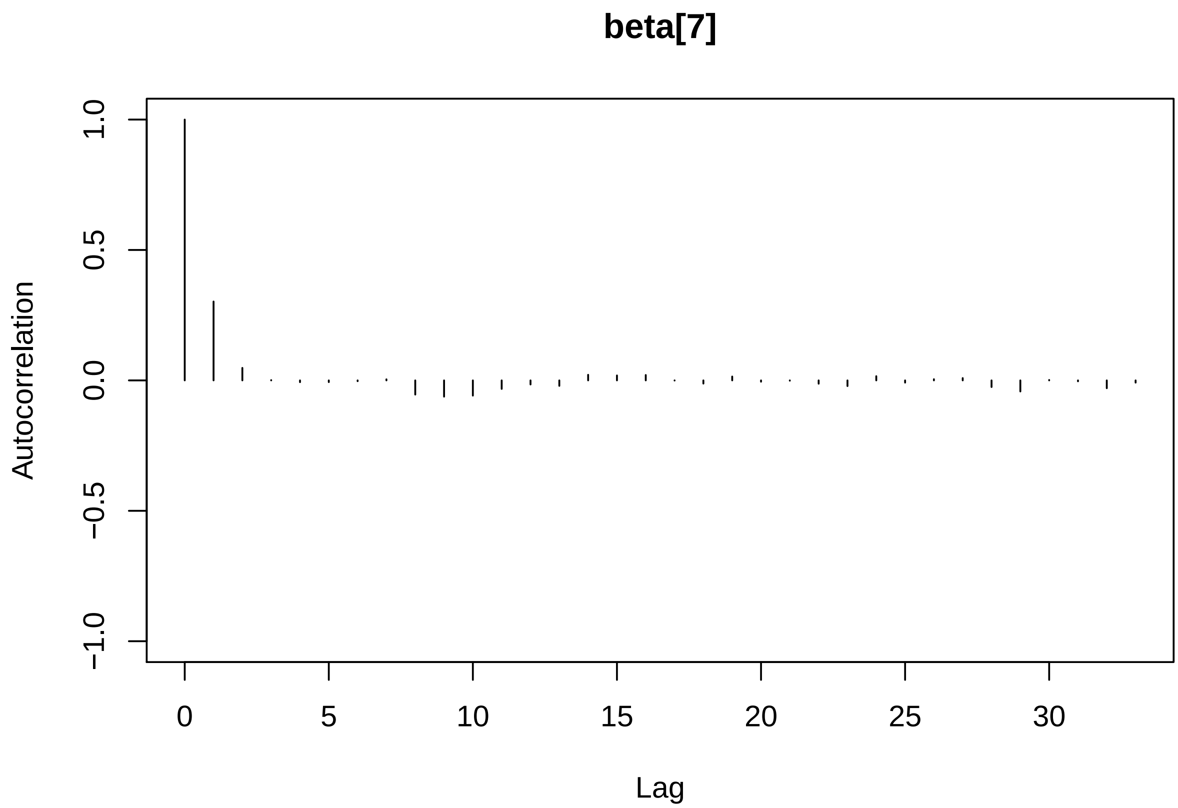

beta[6]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

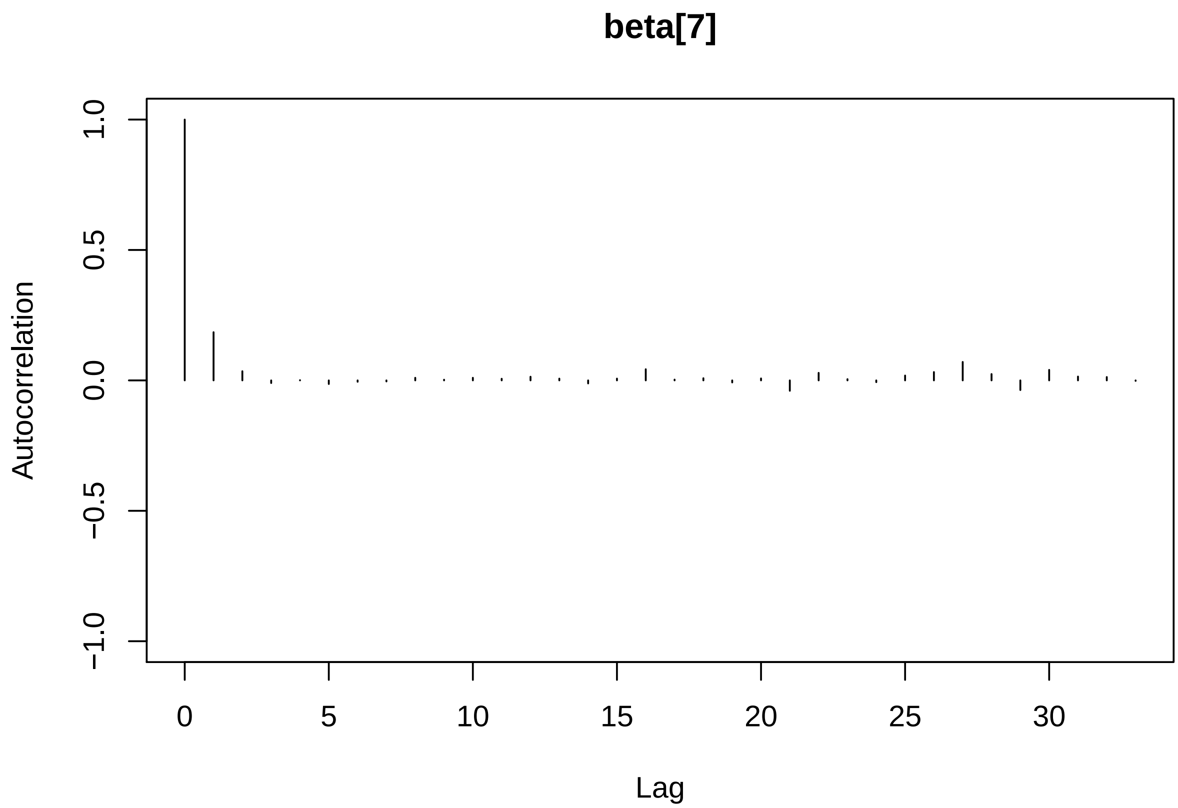


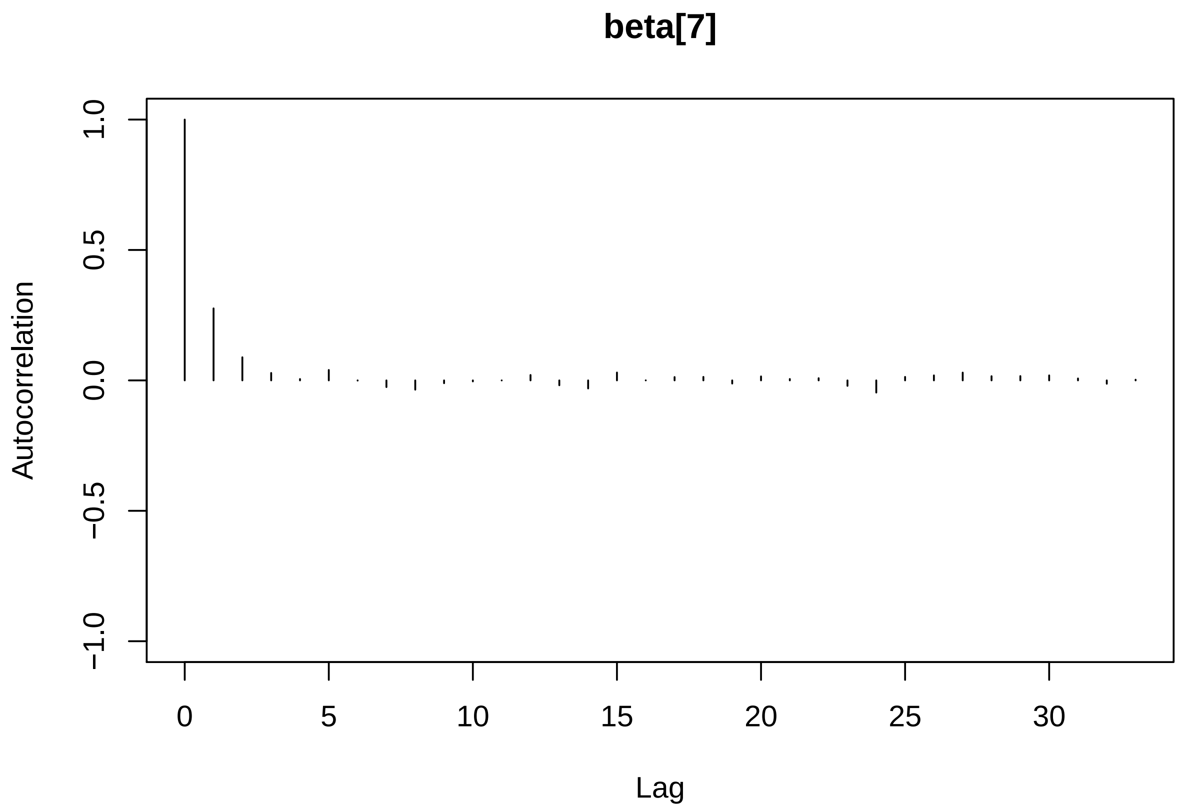
beta[6]
median
97.5%
shrink factor
1.00
1.01
1.02
1.03
1.04
1.05
21000
21500
22000
22500
23000
last iteration in chain

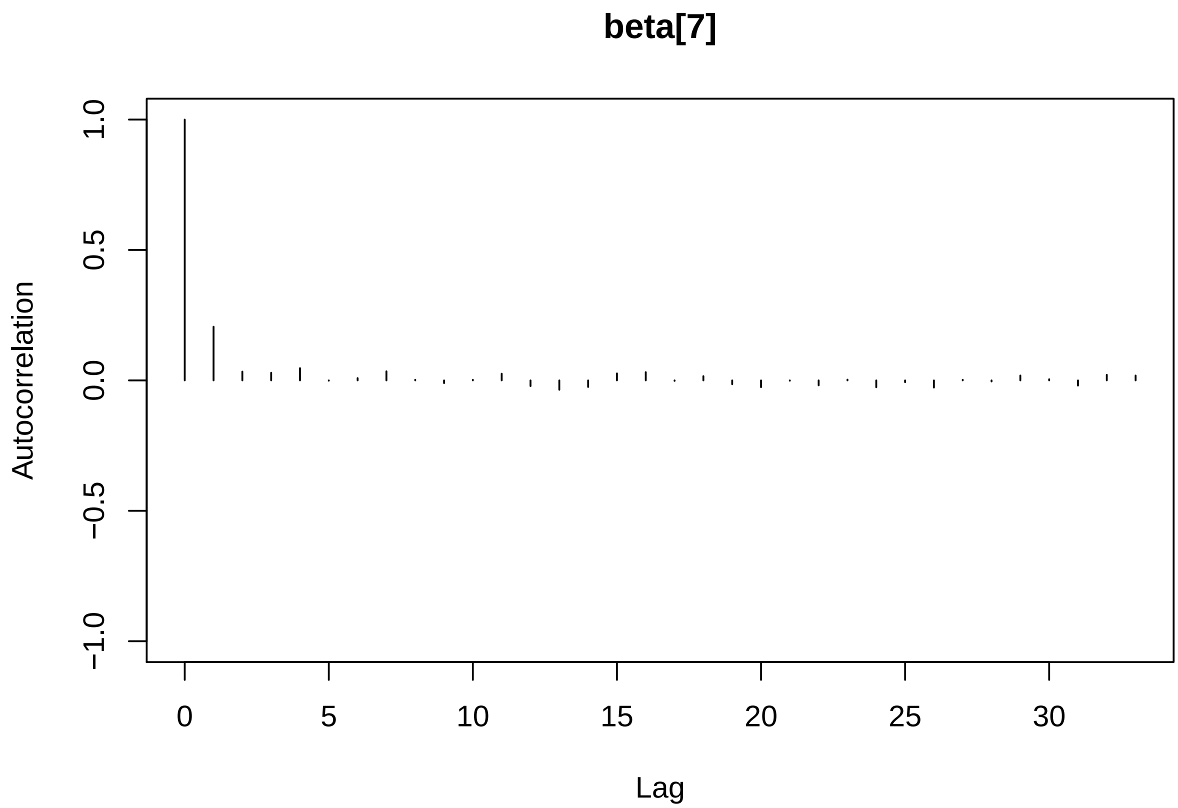


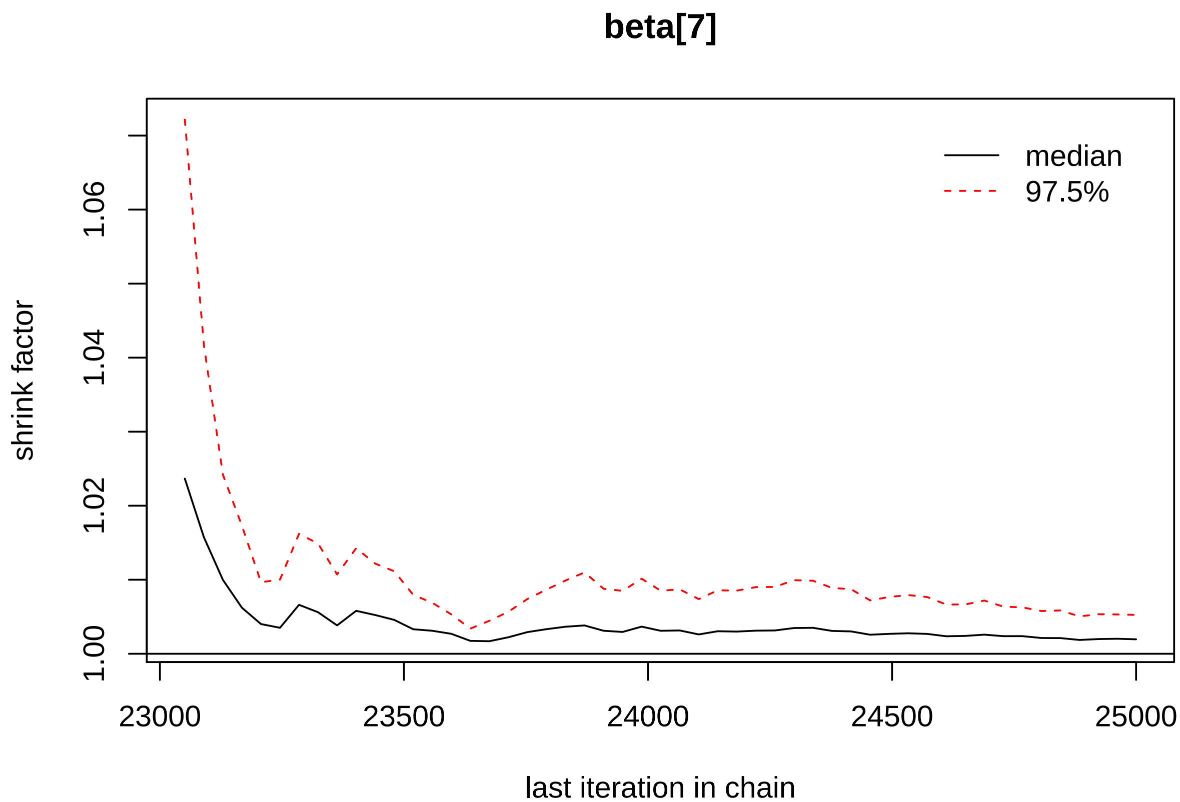
Trace of beta[6]
0.04
0.02
0.00
−0.02
−0.04
21000
22000
23000
Iterations
Density of beta[6]
40
30
20
10
0
−0.04
0.00
0.04
N = 2000 Bandwidth = 0.001776

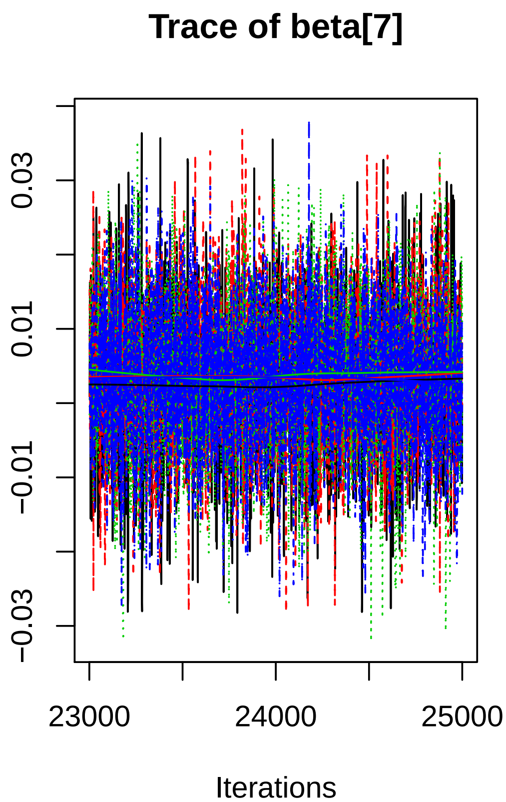

beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

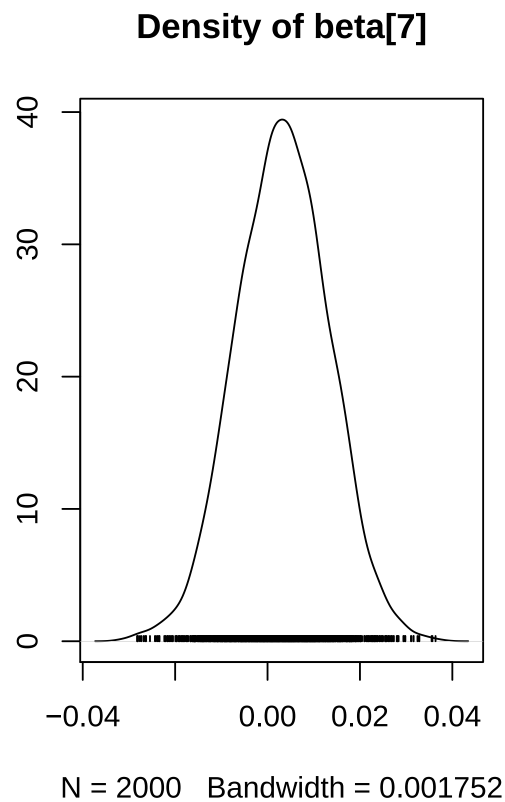

beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

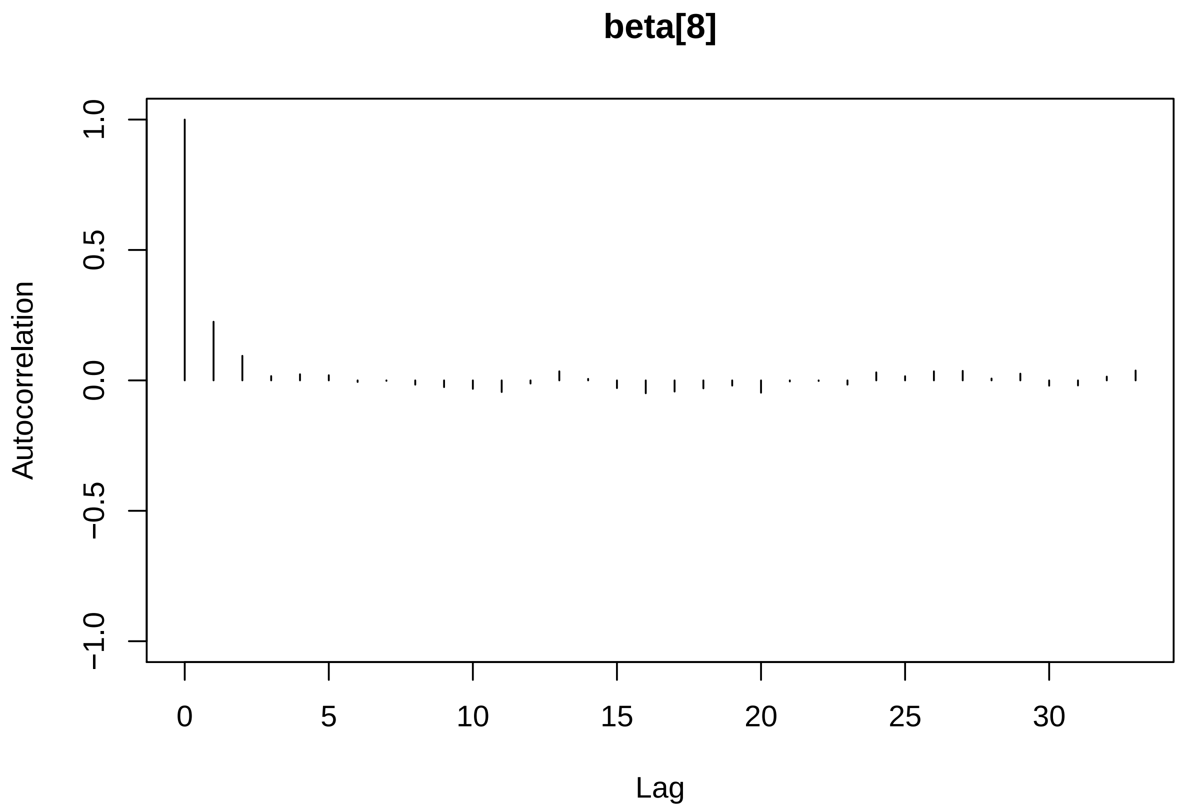

beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[7]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

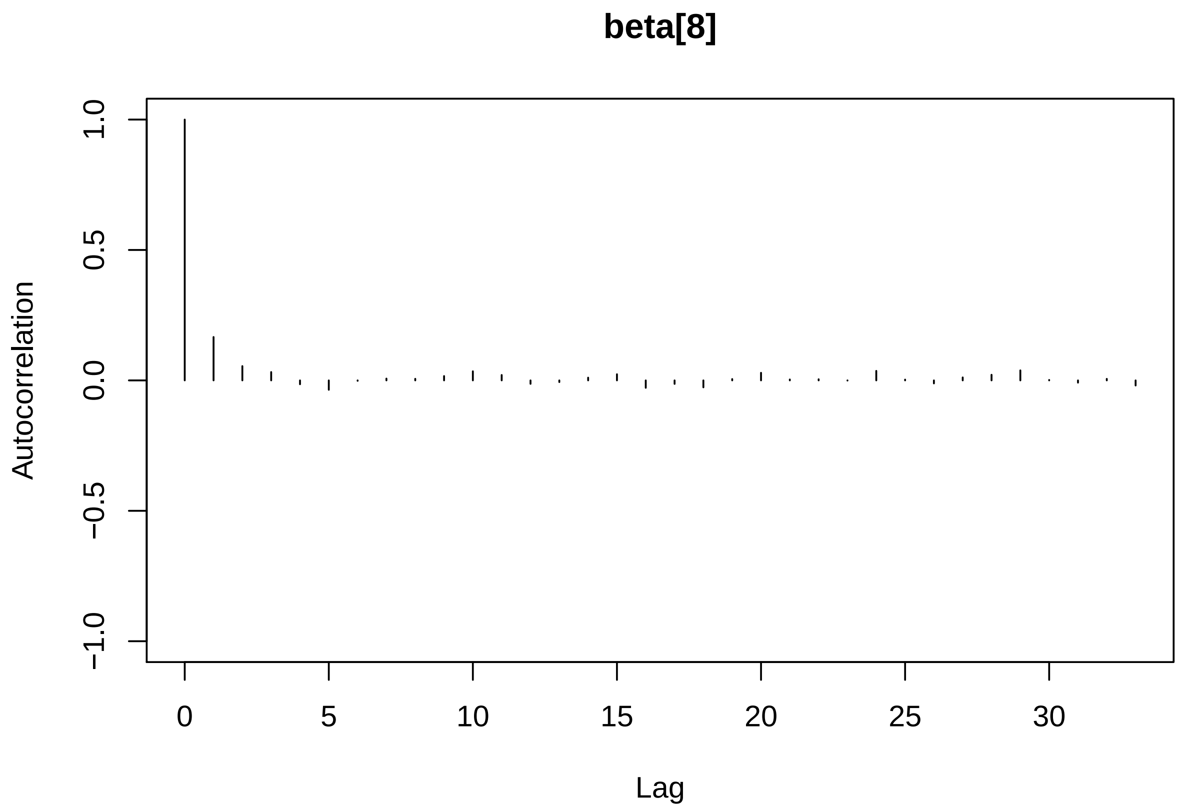

beta[7]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
23000
23500
24000
24500
25000
last iteration in chain

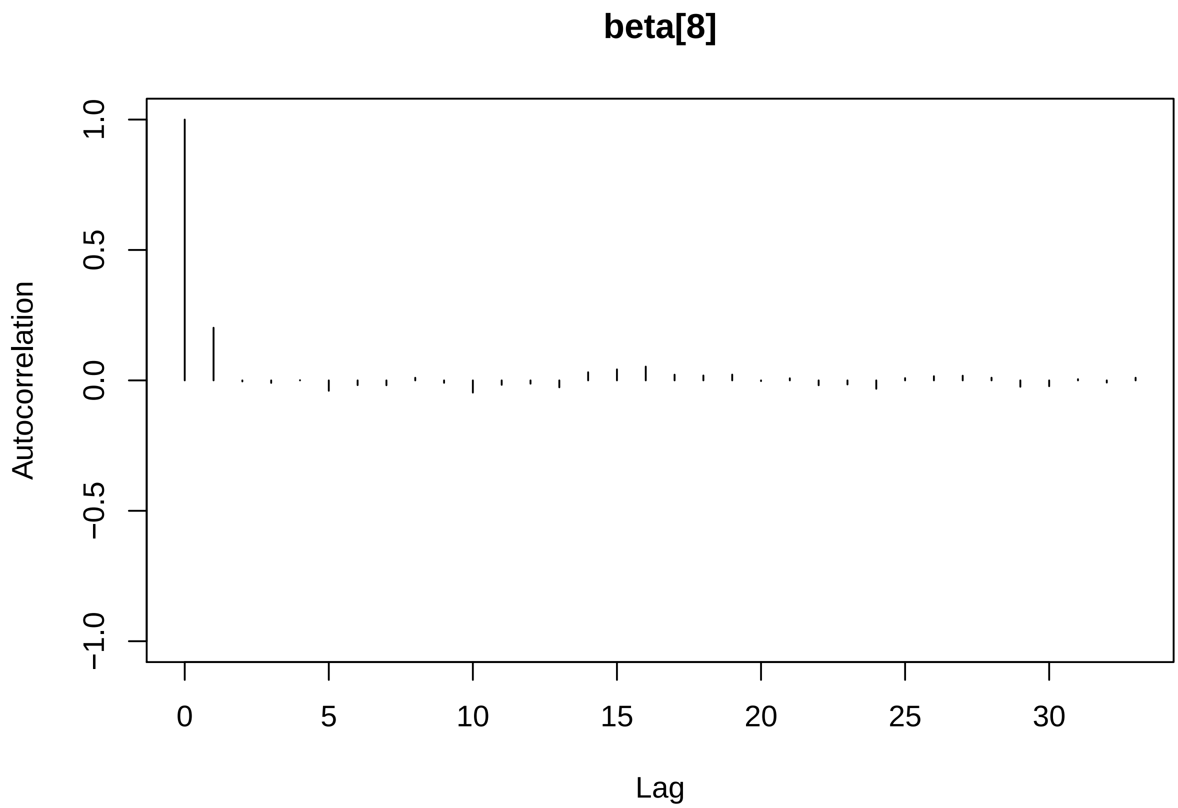

Trace of beta[7]
0.03
0.01
−0.01
−0.03
23000
24000
25000
Iterations

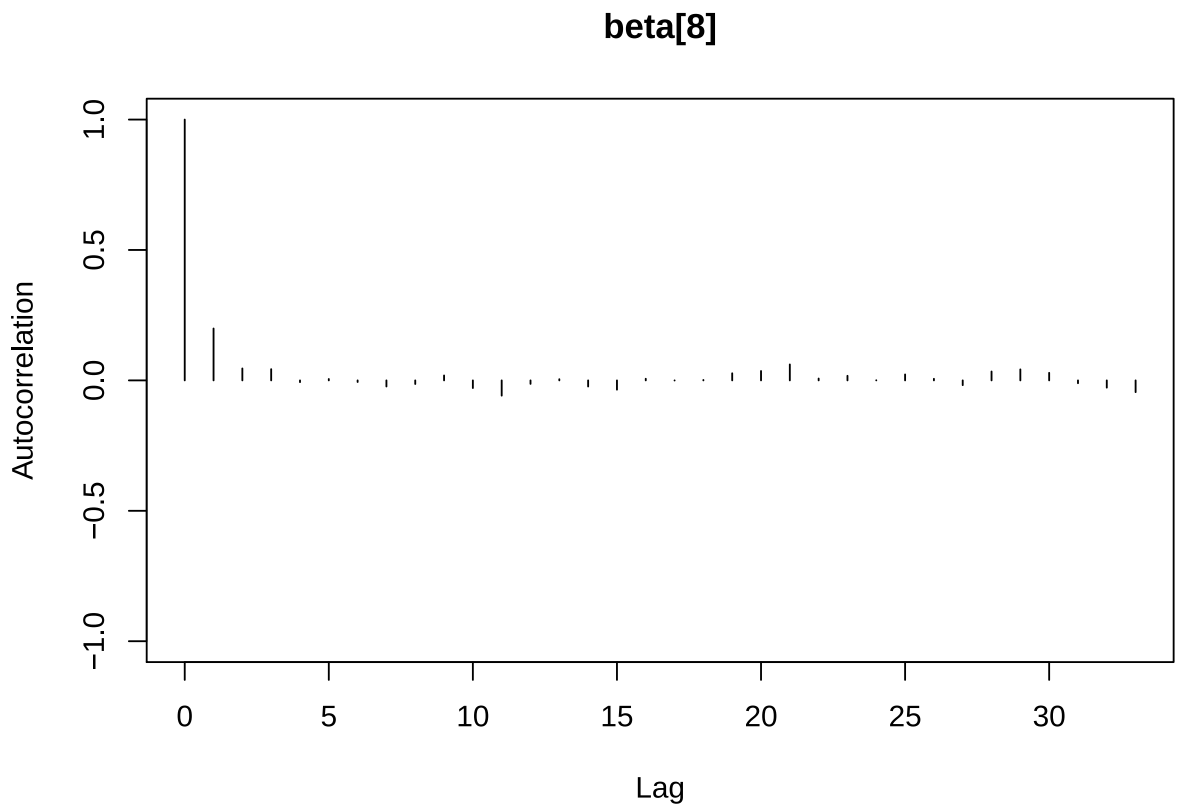

Density of beta[7]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001752

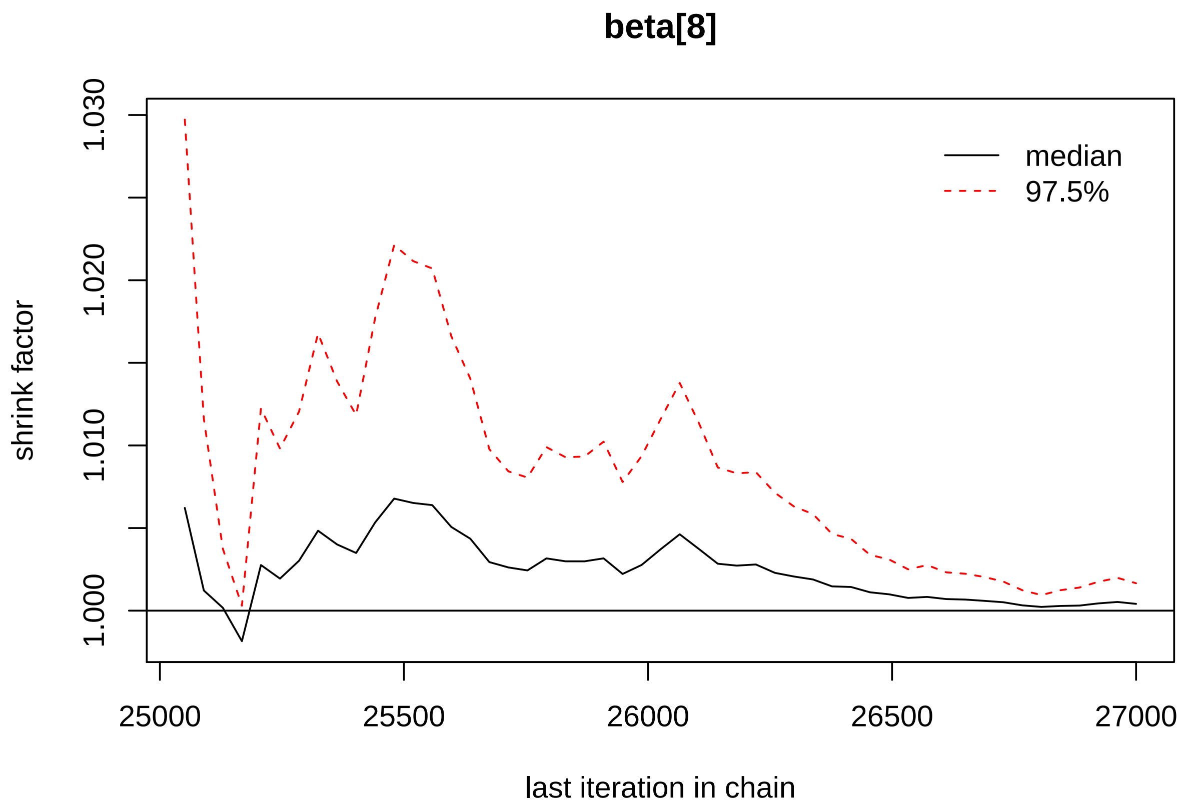


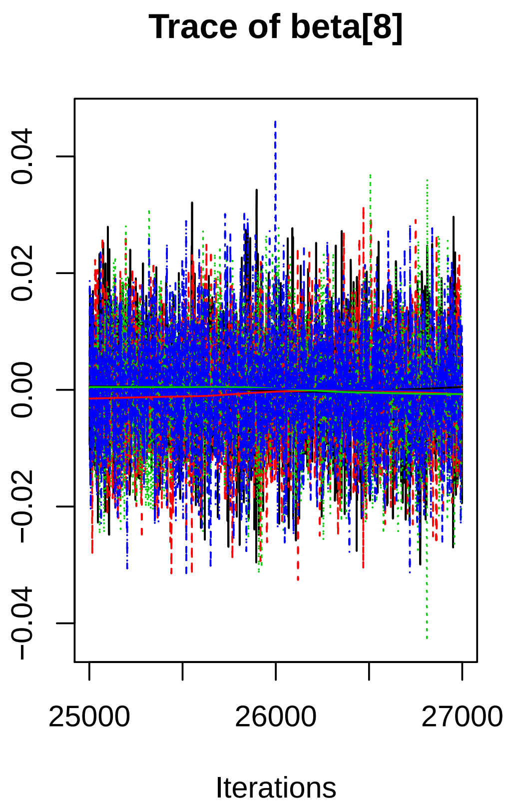
beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

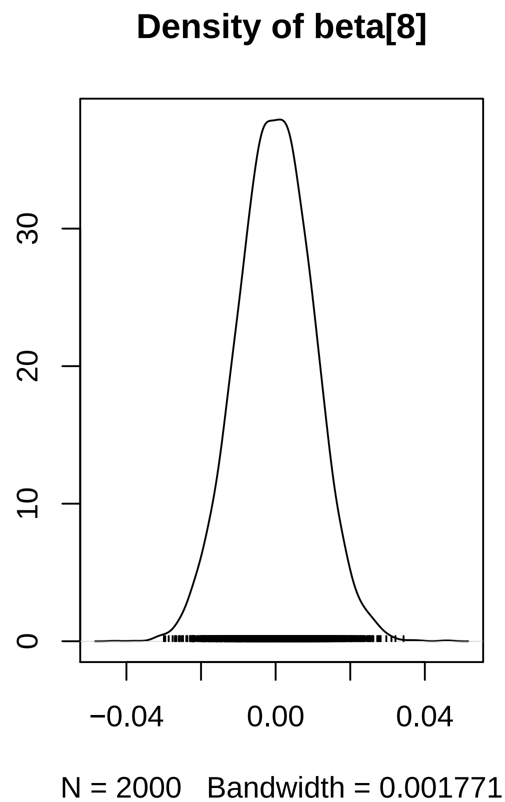

beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

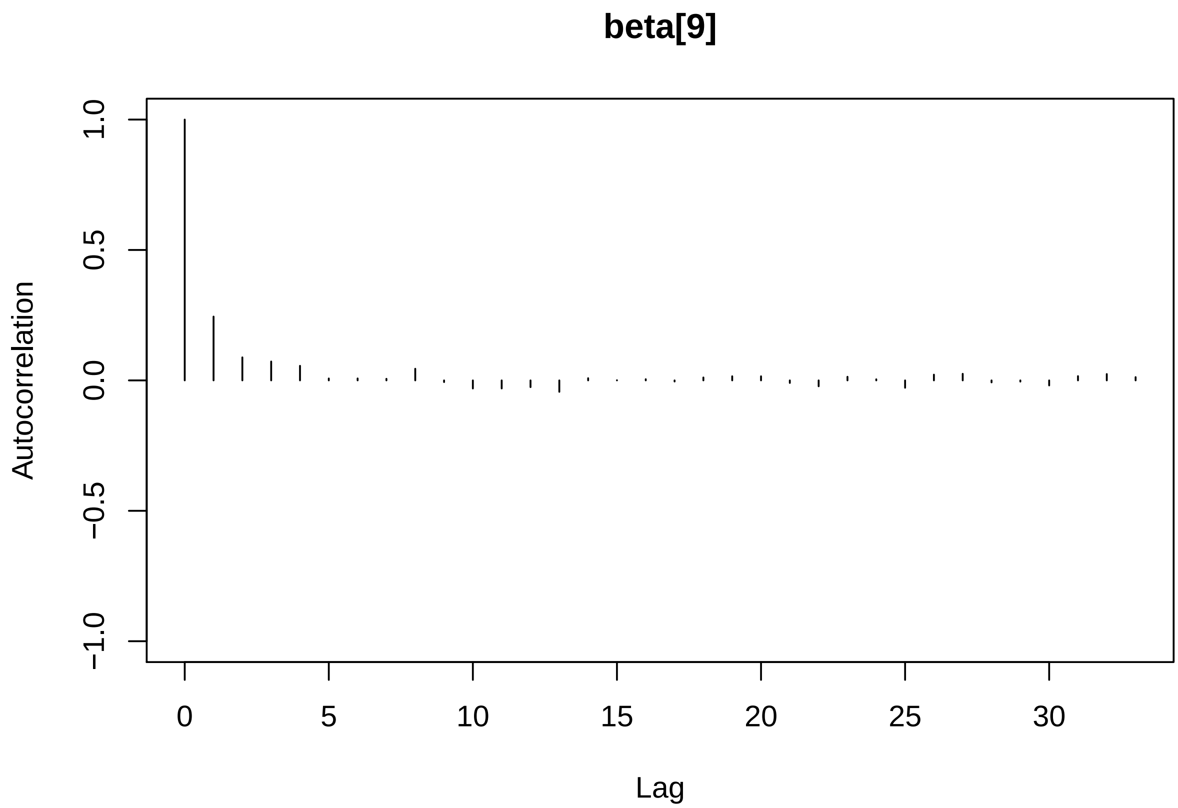

beta[8]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

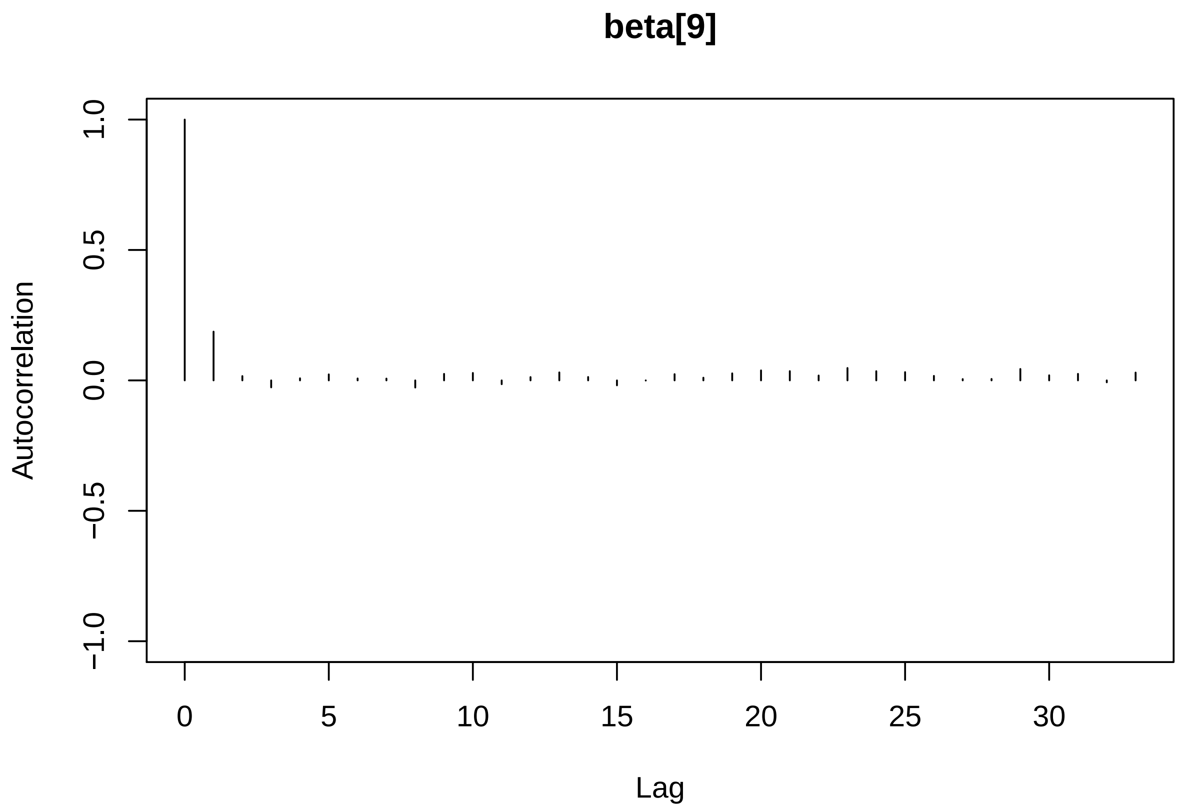

beta[8]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[8]
shrink factor
1.000
1.010
1.020
1.030
median
97.5%
25000
25500
26000
26500
27000
last iteration in chain

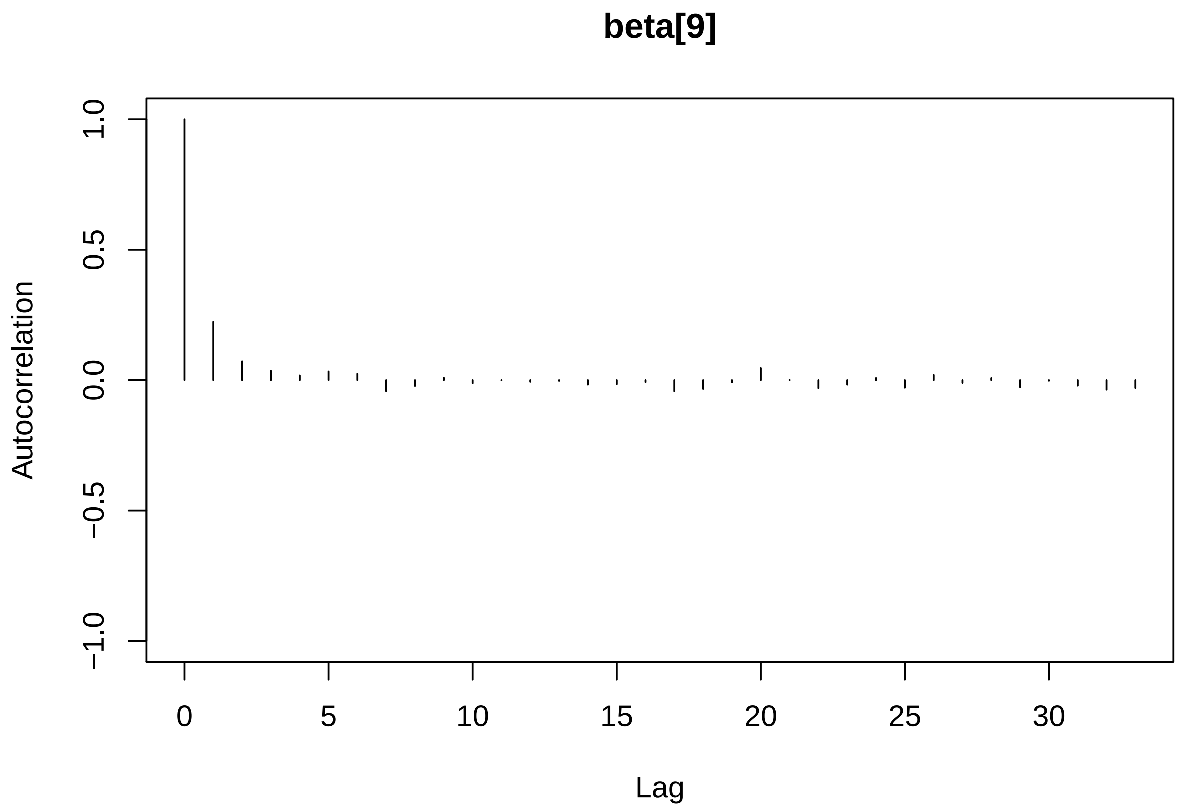

Trace of beta[8]
0.04
0.02
0.00
−0.02
−0.04
25000
26000
27000
Iterations

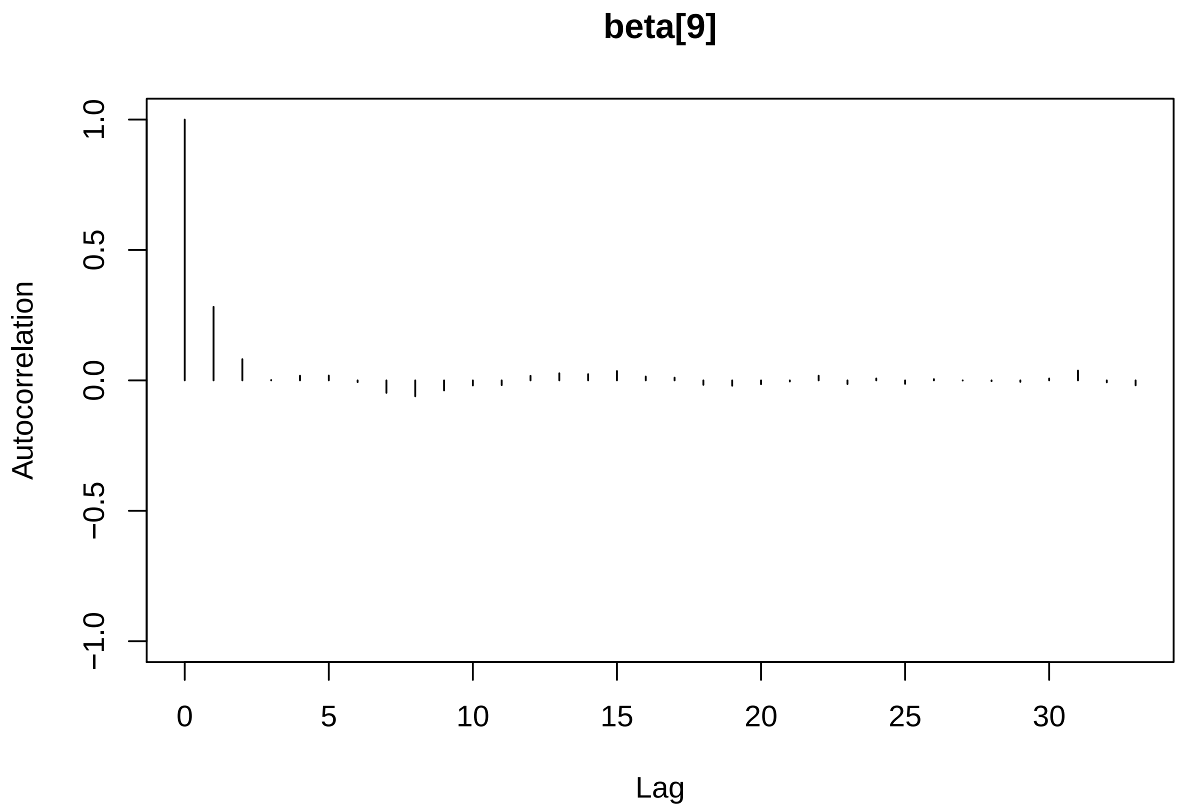

Density of beta[8]
30
20
10
0
−0.04
0.00
0.04
N = 2000   Bandwidth = 0.001771

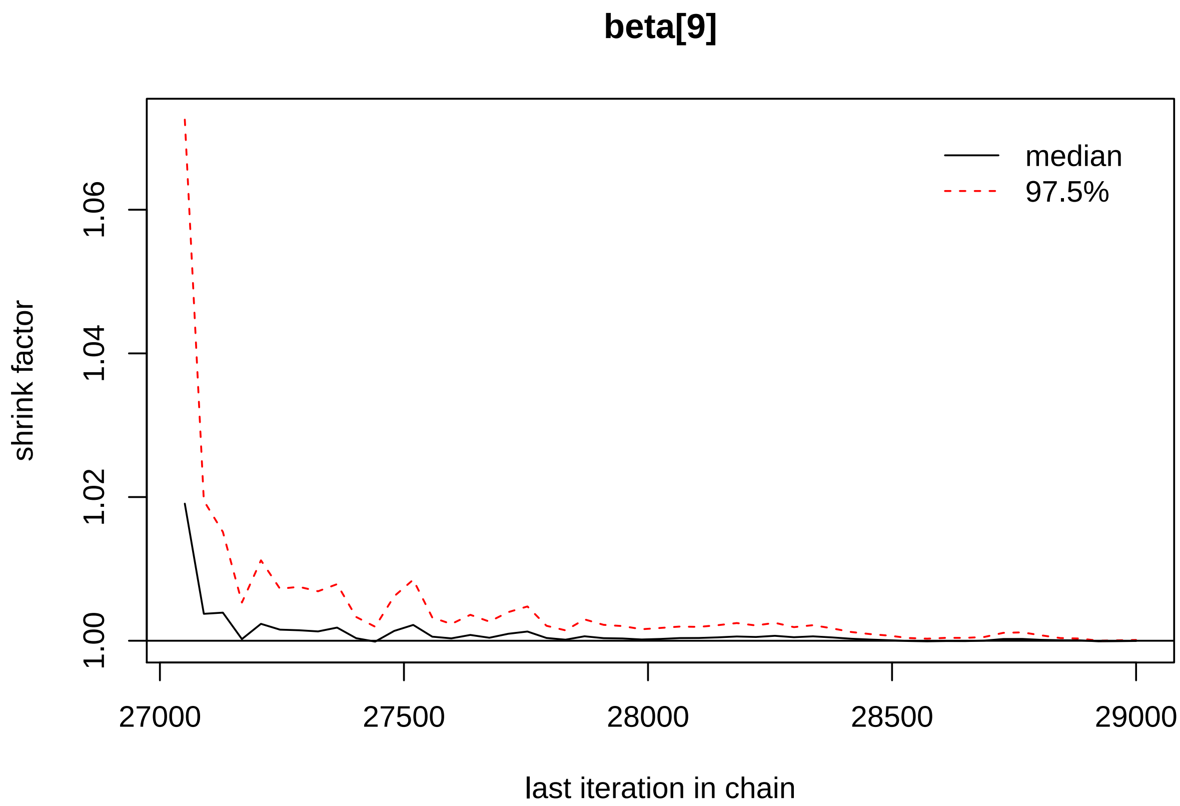

beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag



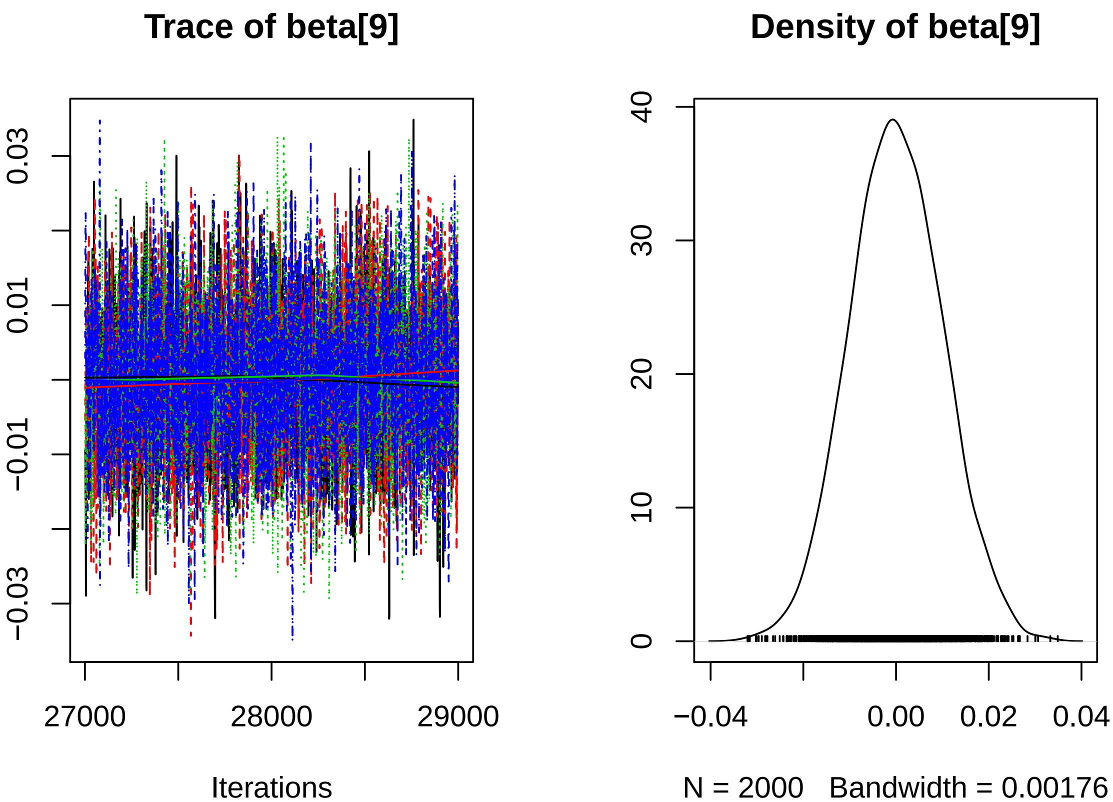
beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

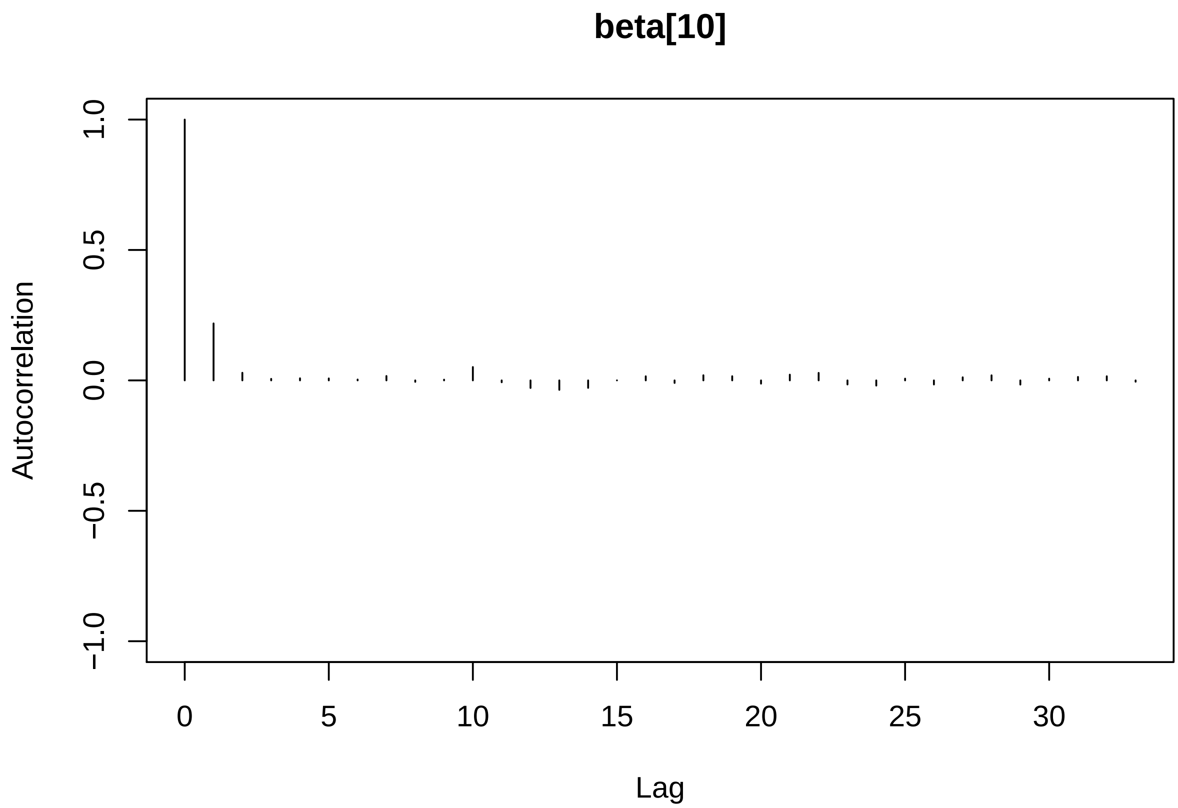

beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[9]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

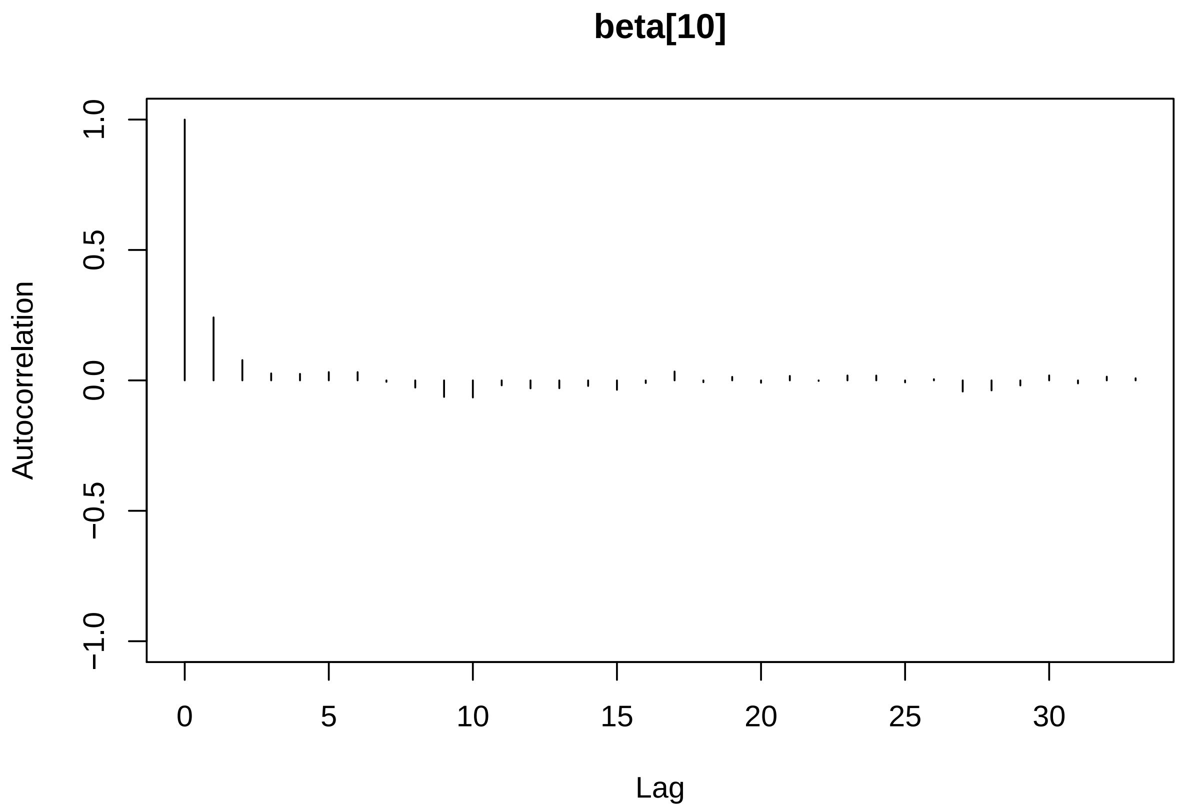


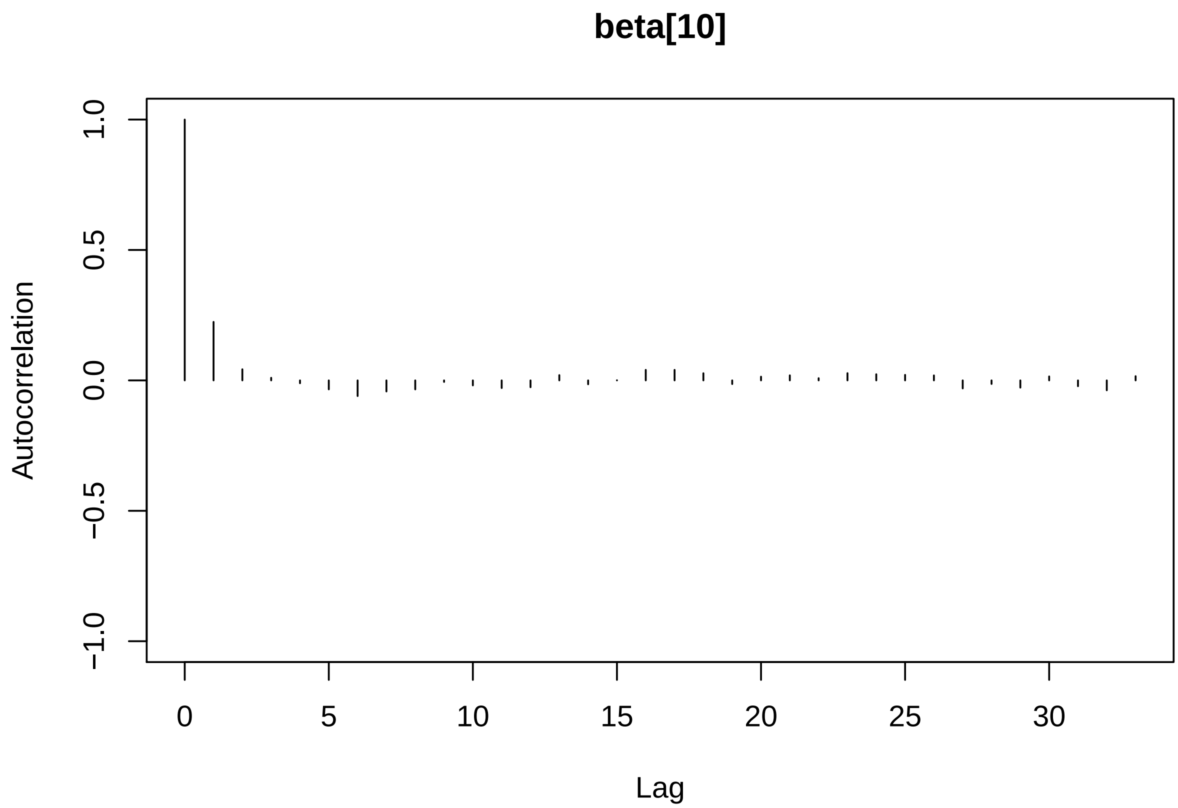
beta[9]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
27000
27500
28000
28500
29000
last iteration in chain

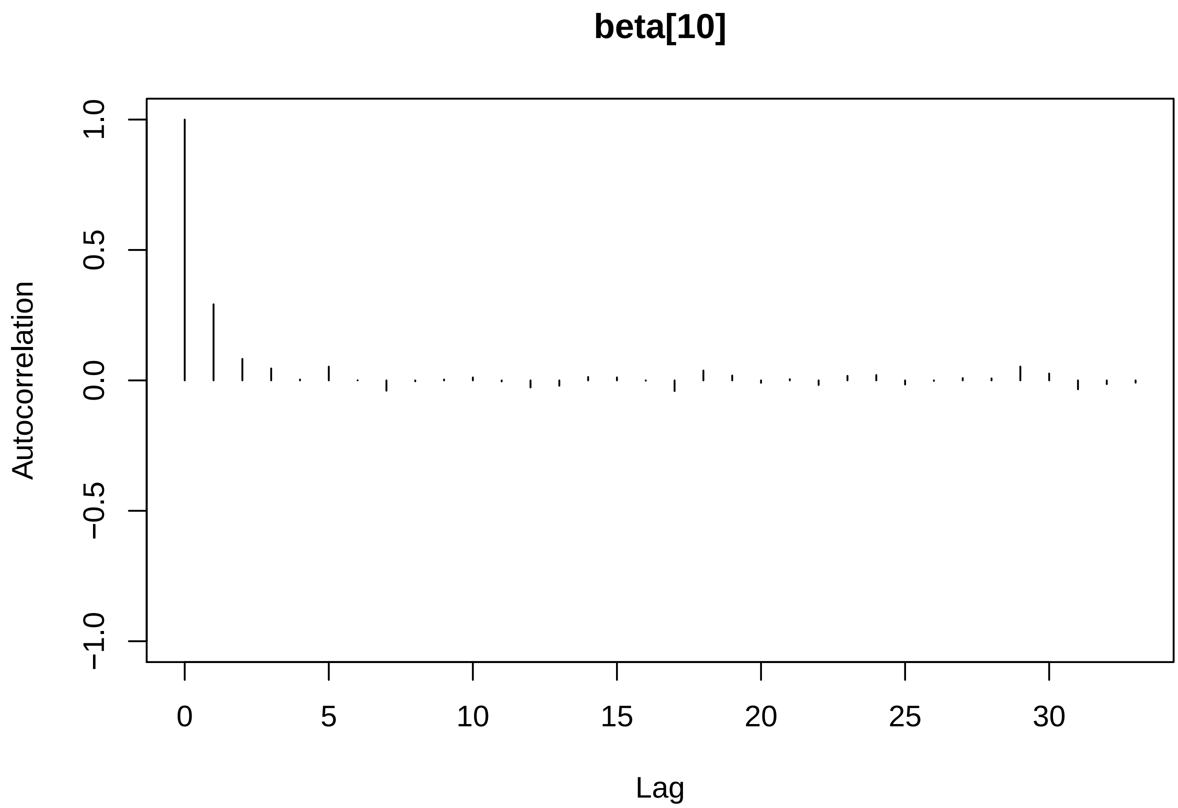

Trace of beta[9]
0.03
0.01
−0.01
−0.03
27000
28000
29000
Iterations
Density of beta[9]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000   Bandwidth = 0.00176

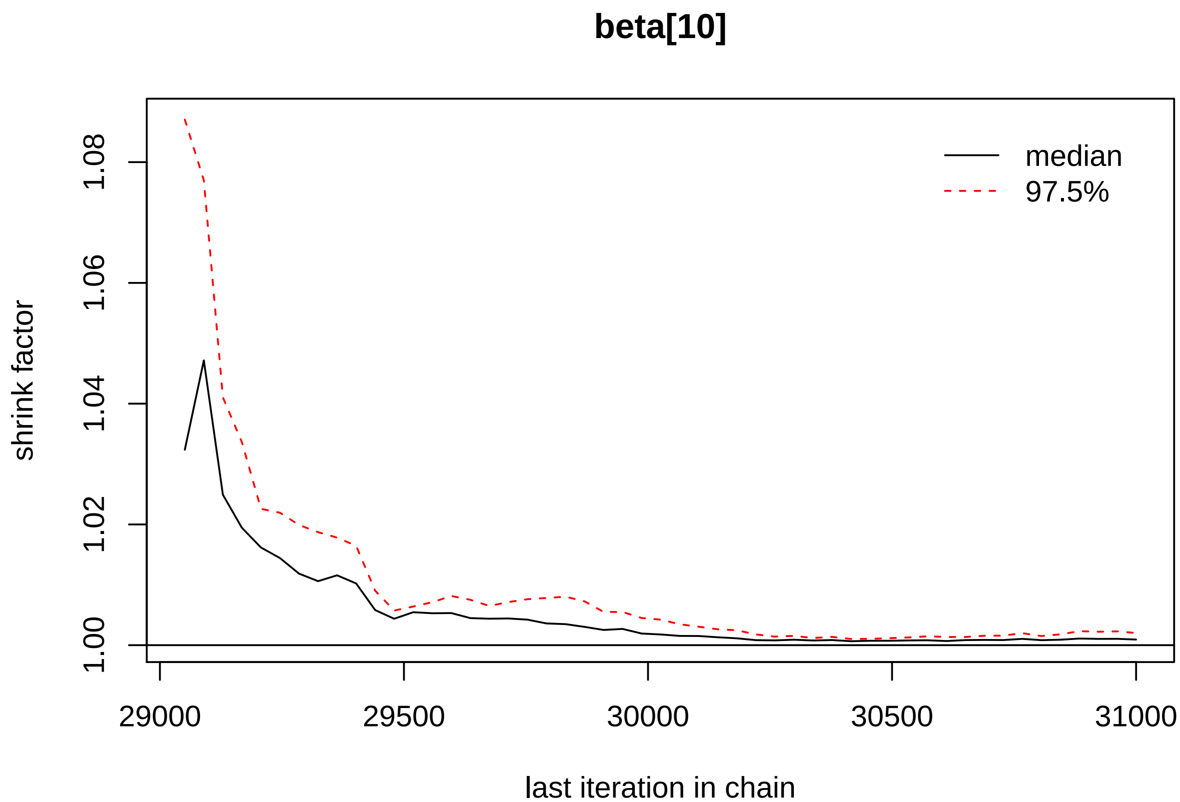


beta[10]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

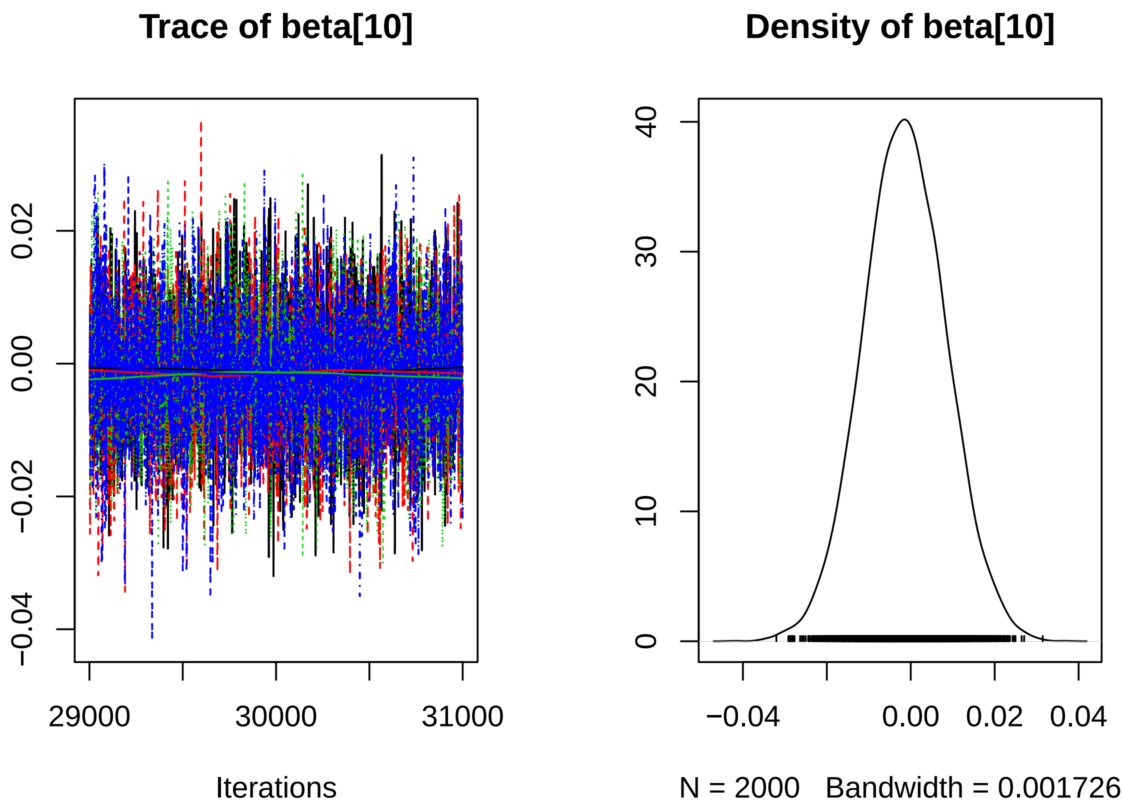

beta[10]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

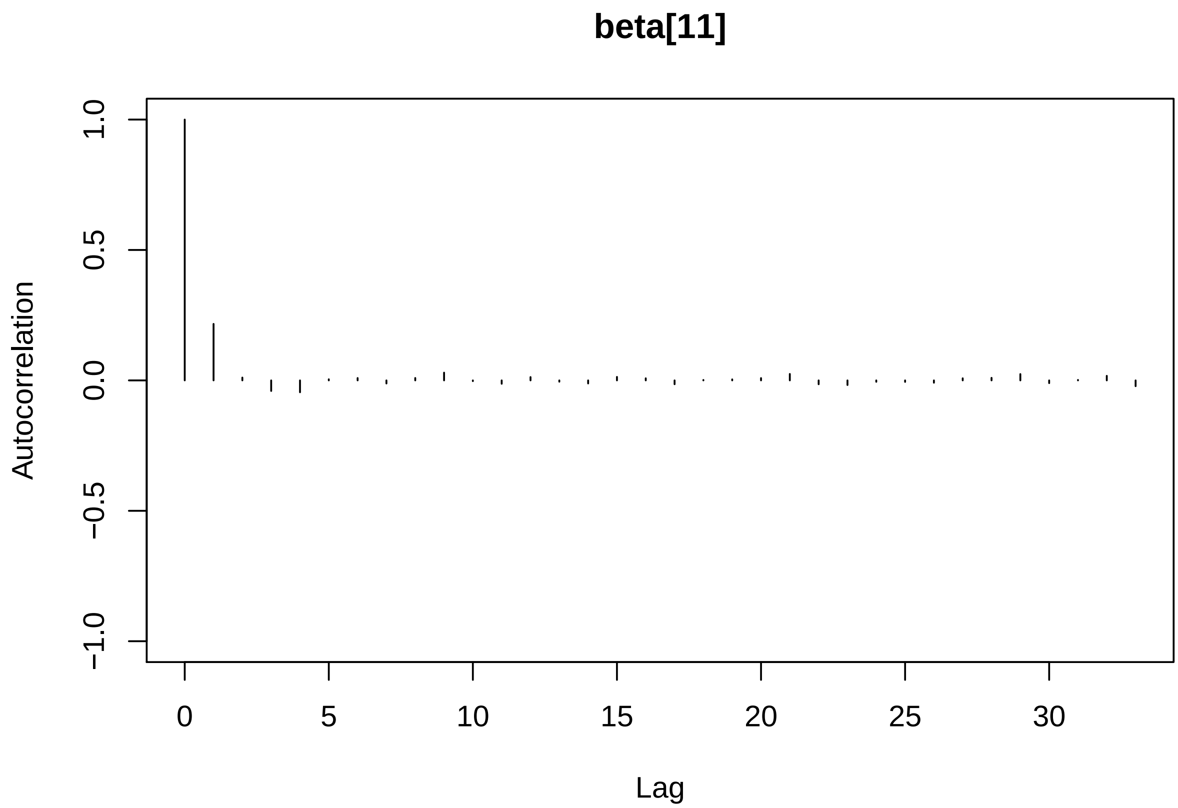

beta[10]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

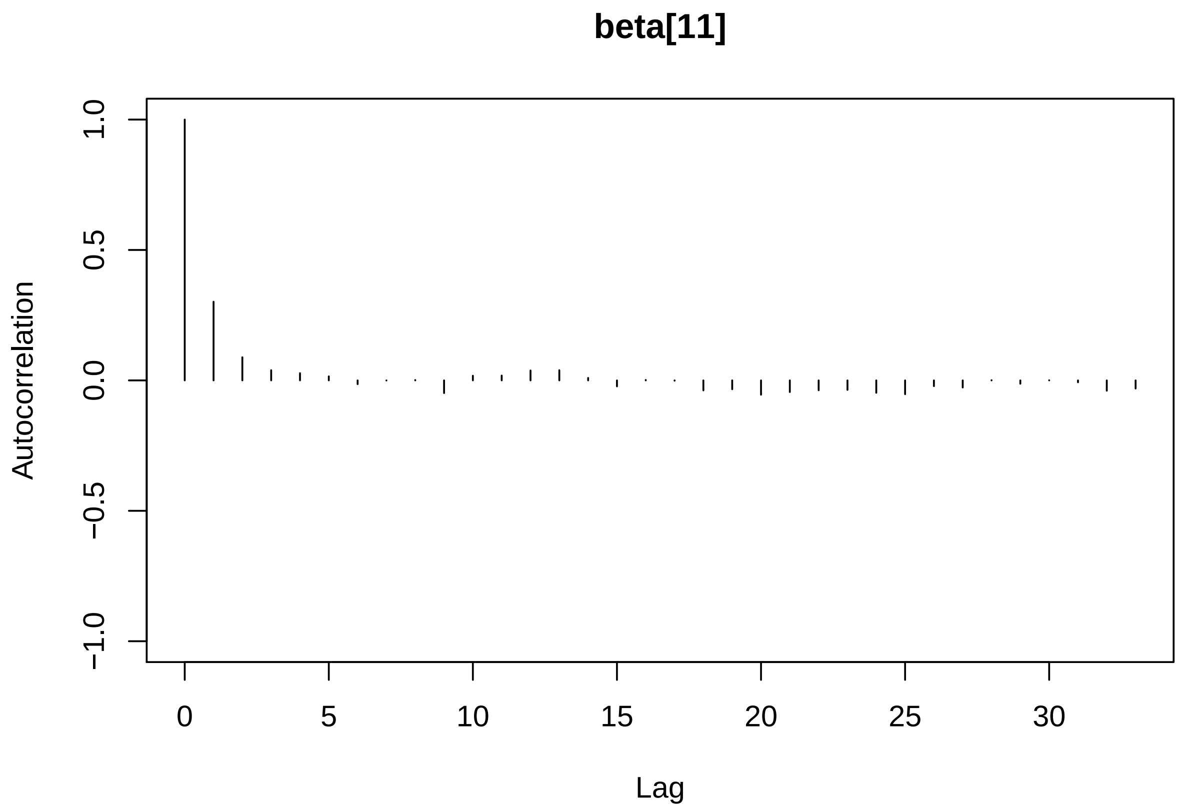

beta[10]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

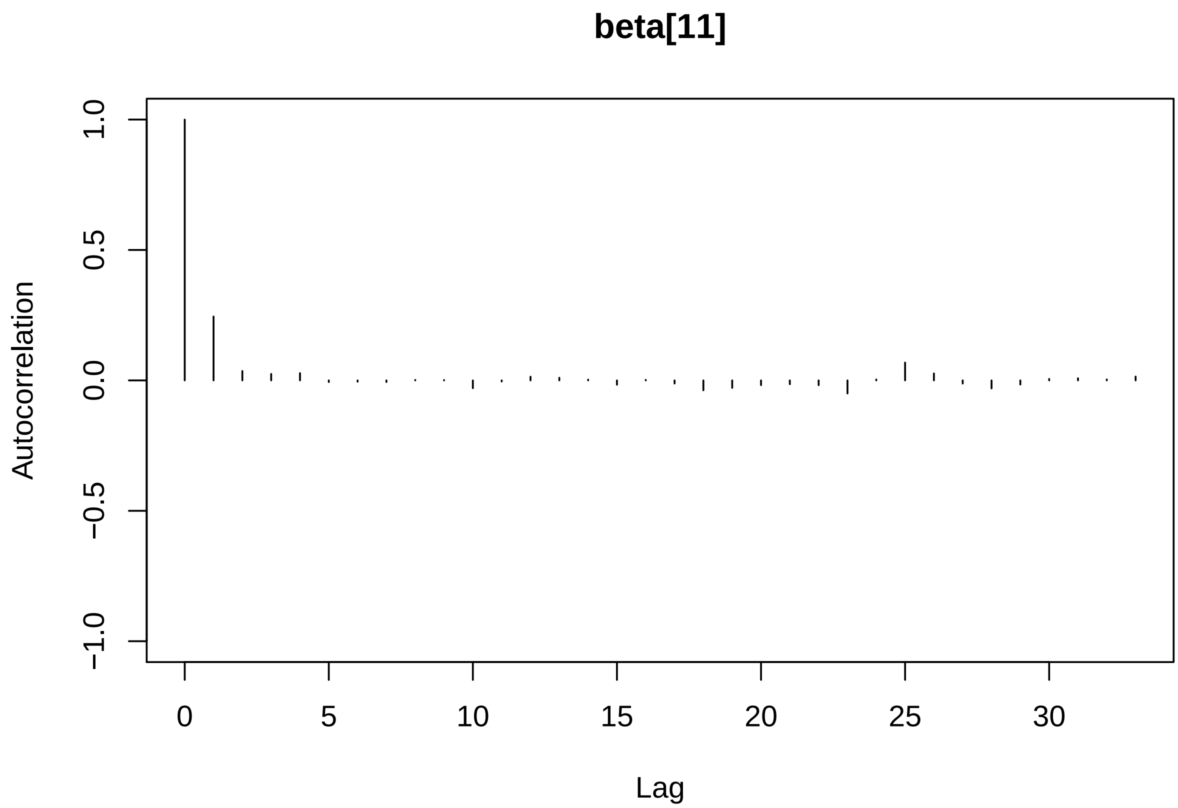


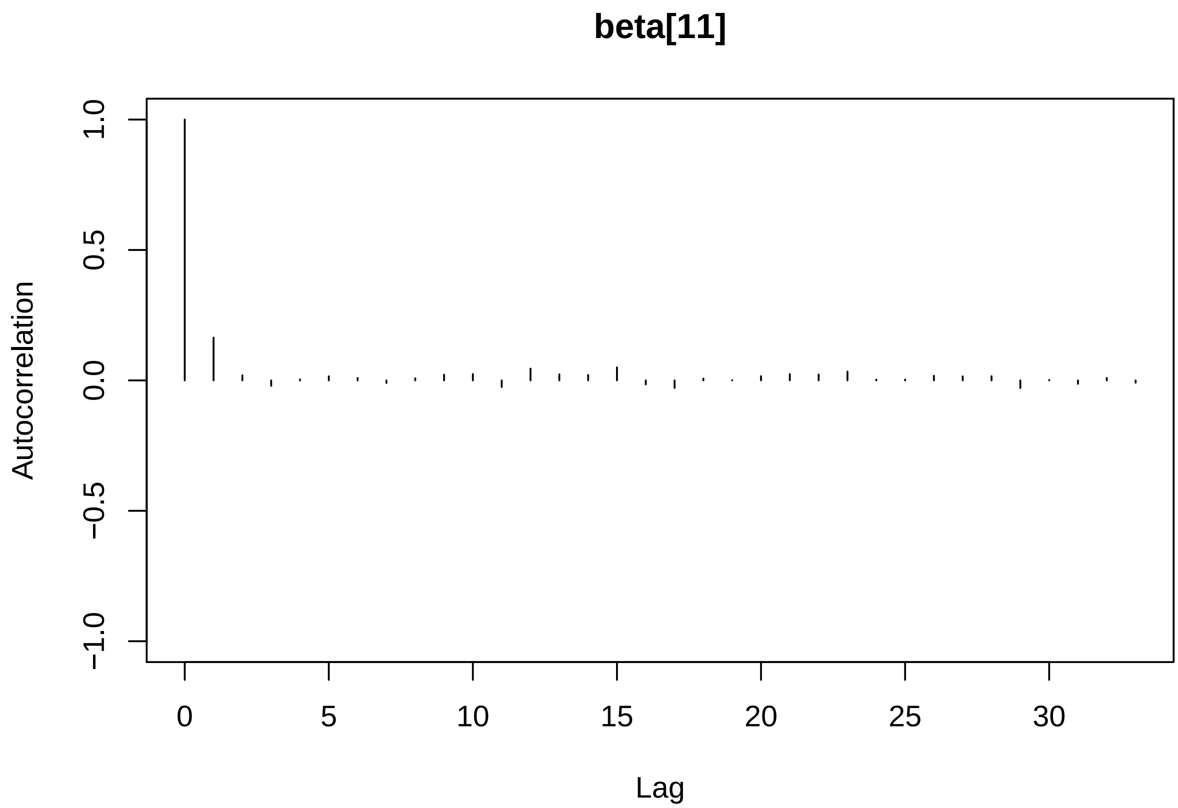
beta[10]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
29000
29500
30000
30500
31000
last iteration in chain

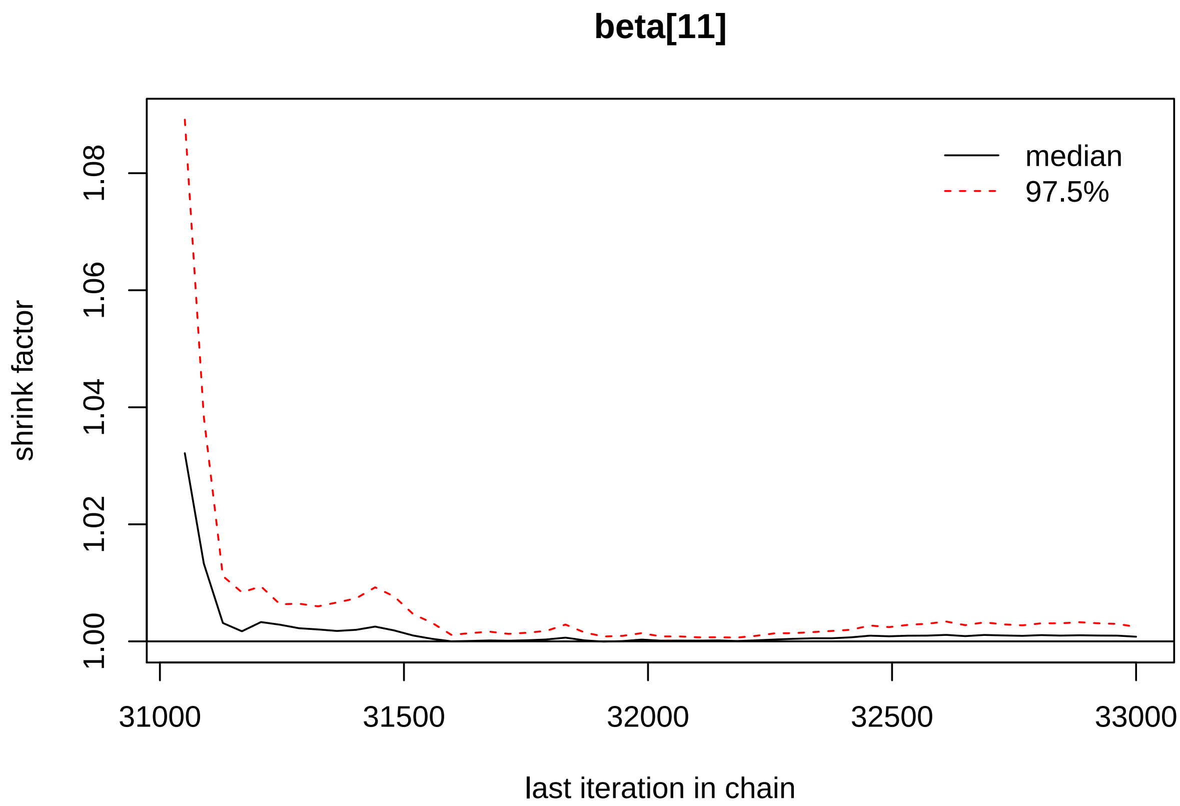

Trace of beta[10]
Density of beta[10]
0.02
0.00
−0.02
−0.04
29000
30000
31000
Iterations
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000   Bandwidth = 0.001726

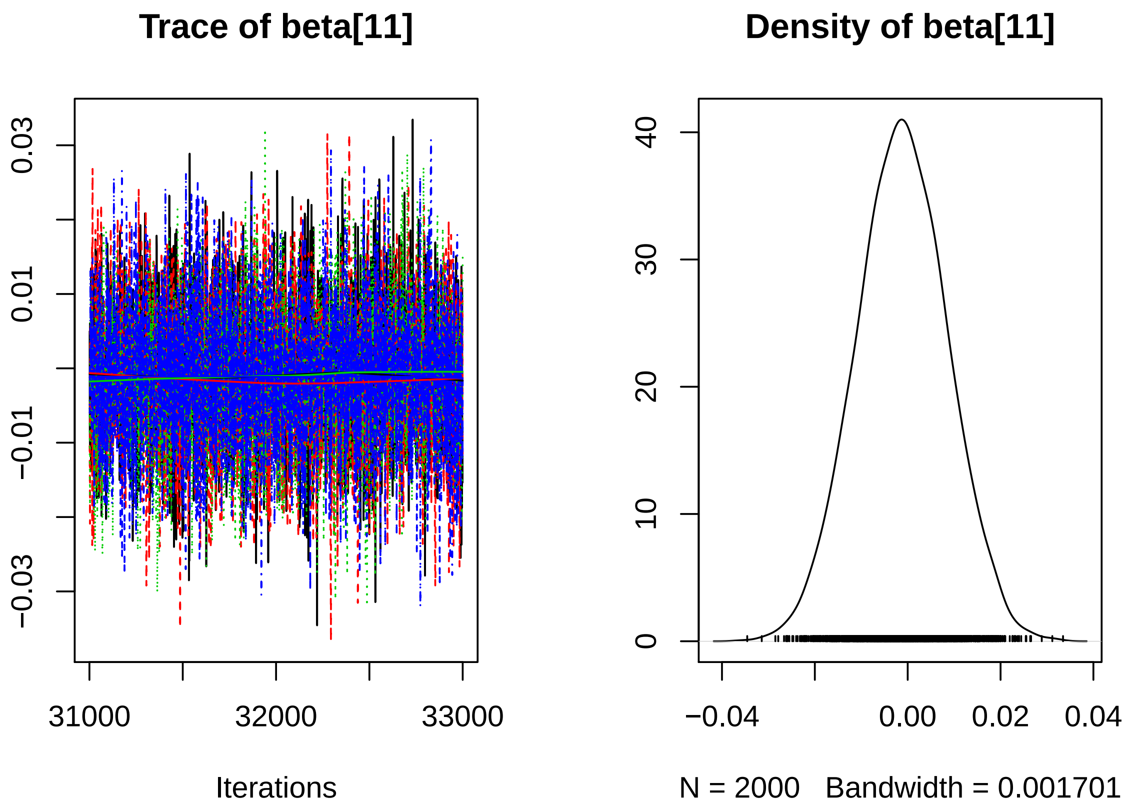

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

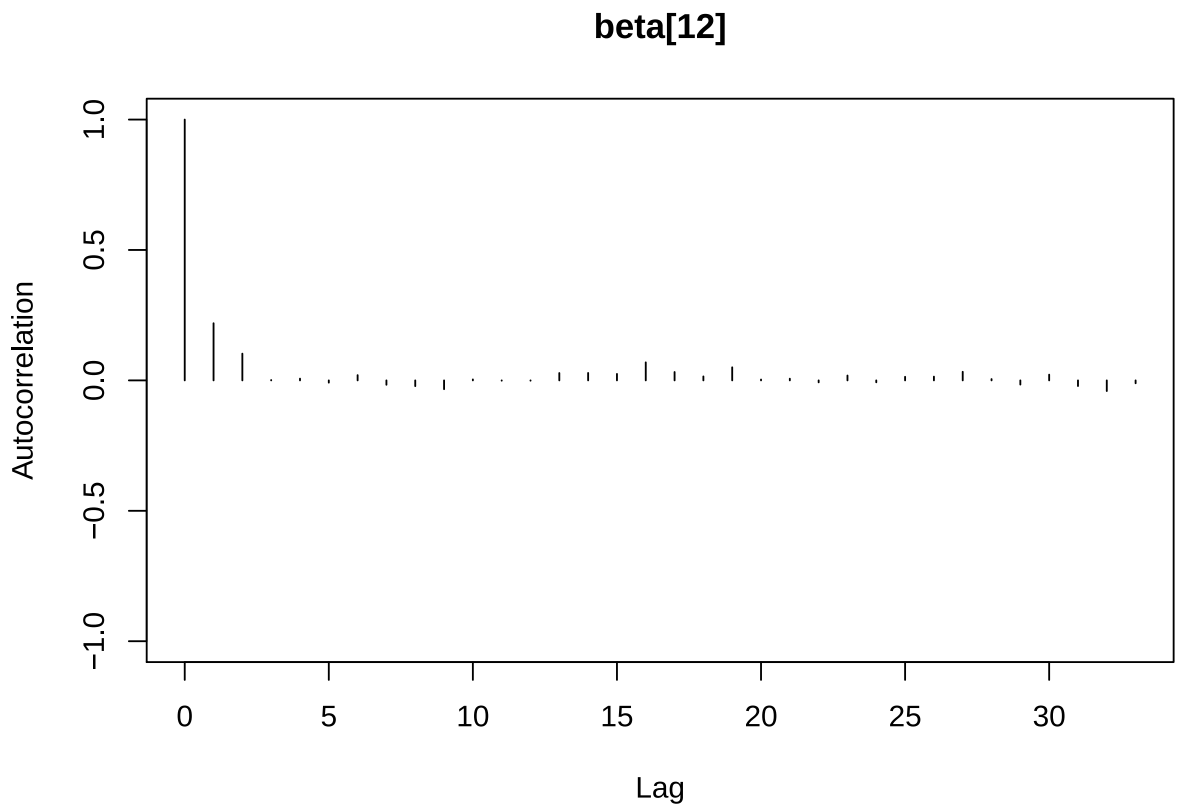

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

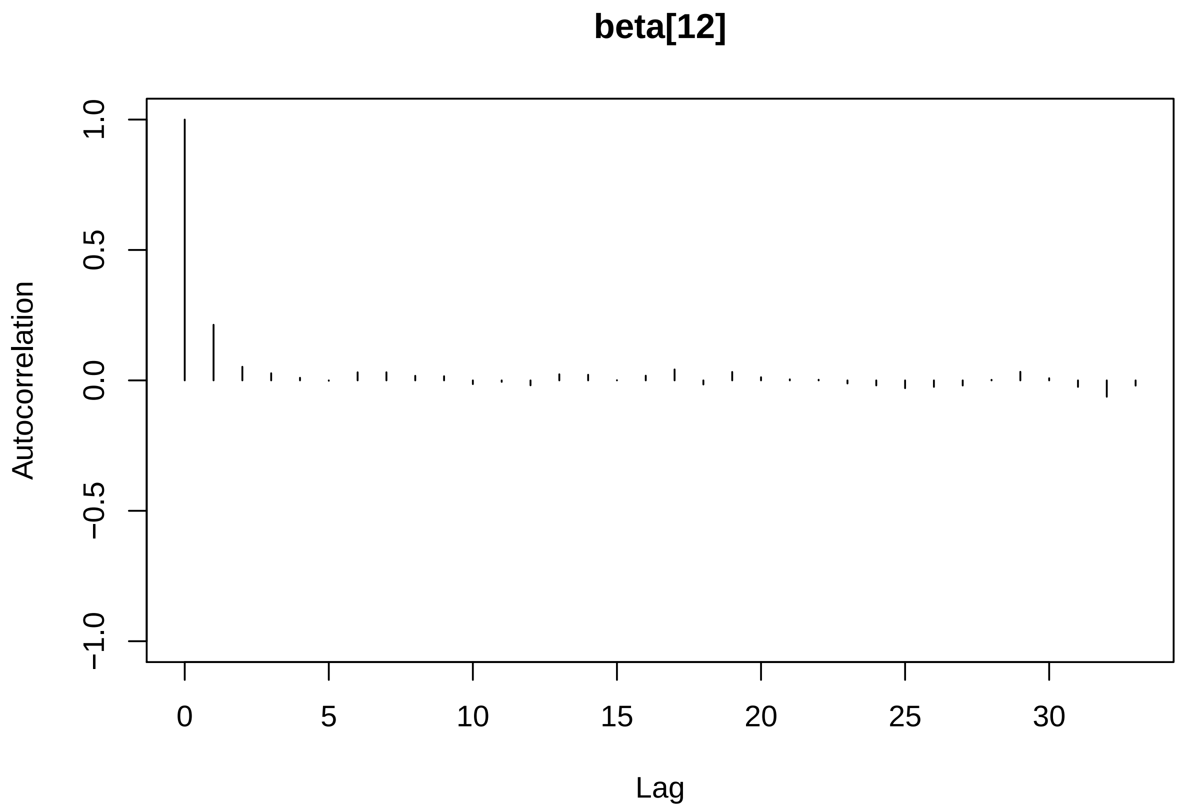

beta[11]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

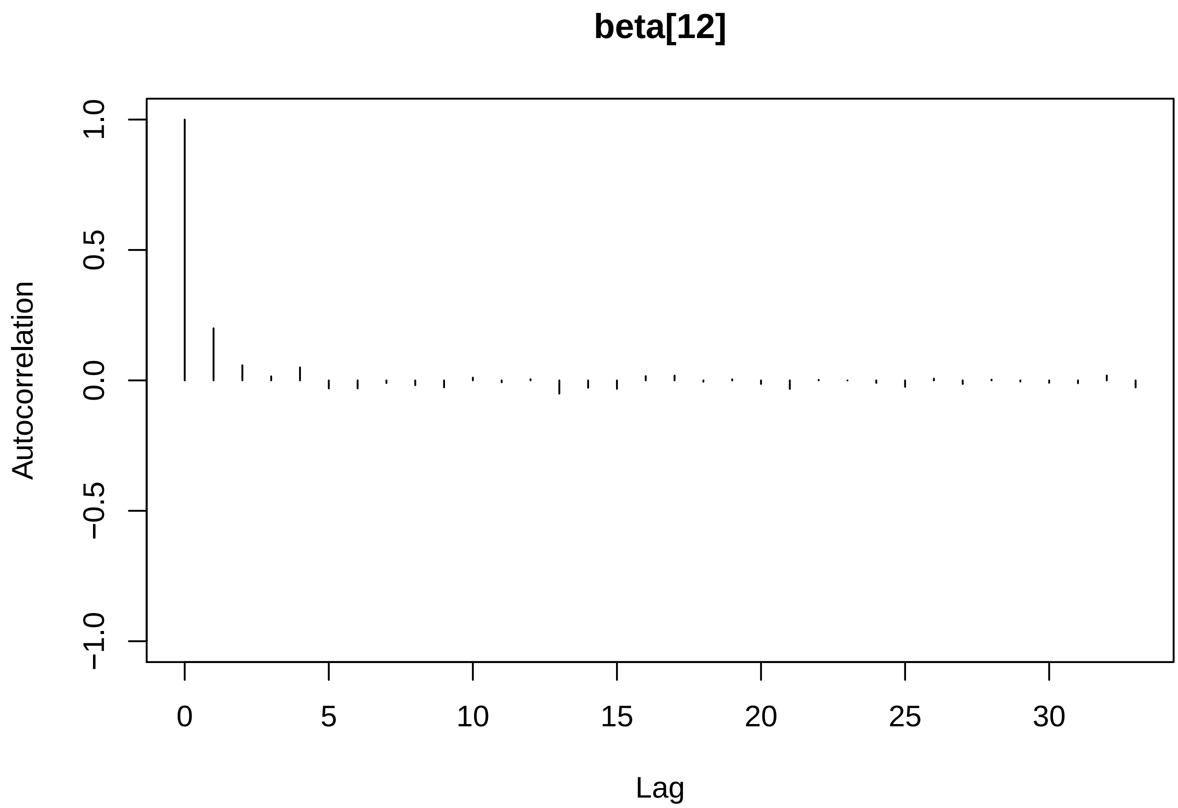

beta[11]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

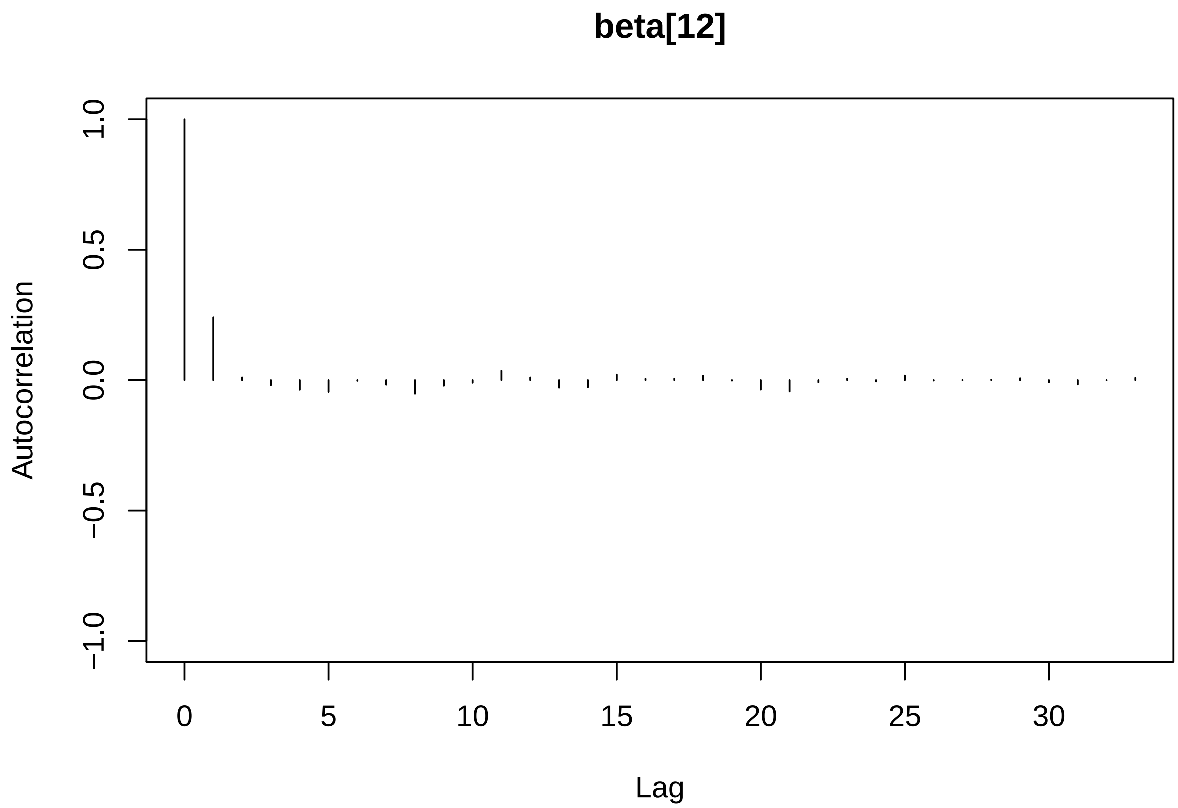

beta[11]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
31000
31500
32000
32500
33000
last iteration in chain

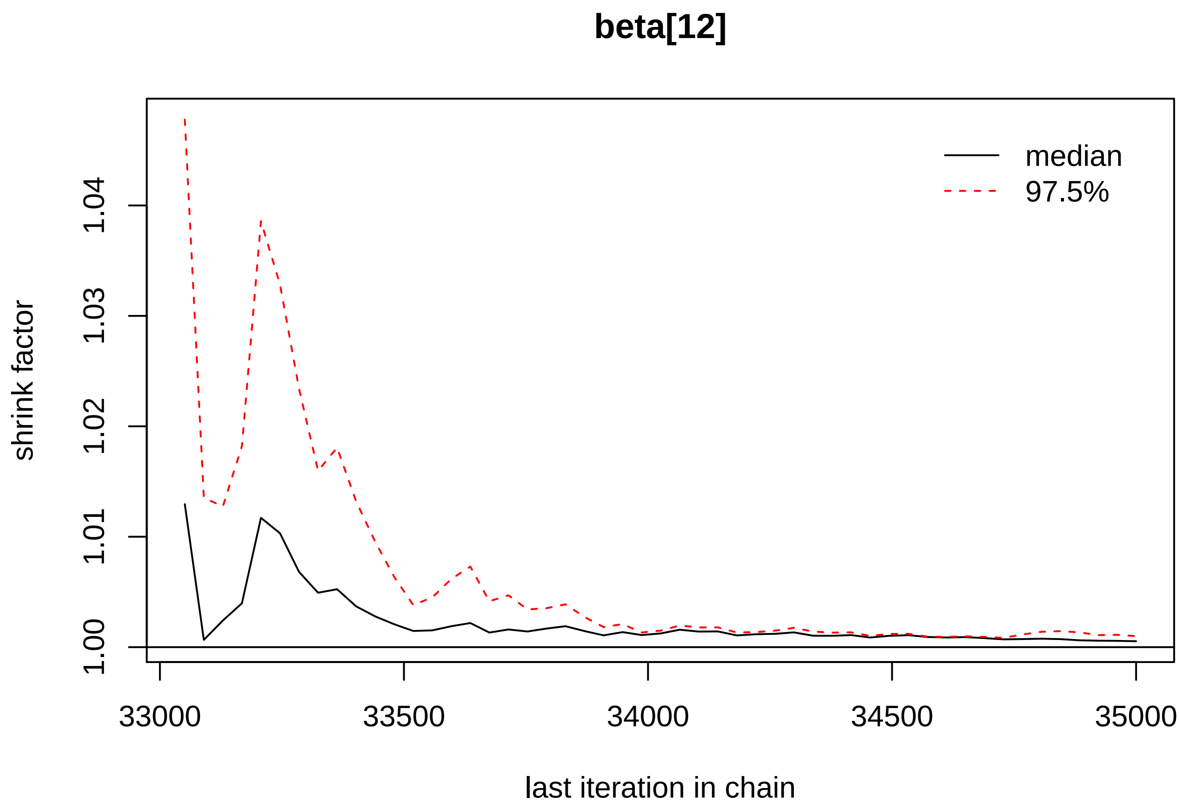

Trace of beta[11]
0.03
0.01
−0.01
−0.03
31000
32000
33000
Iterations
Density of beta[11]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001701

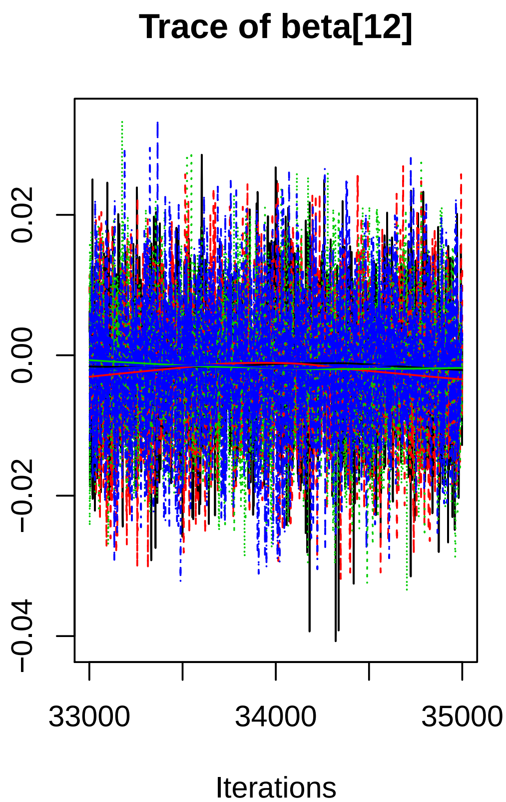

beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

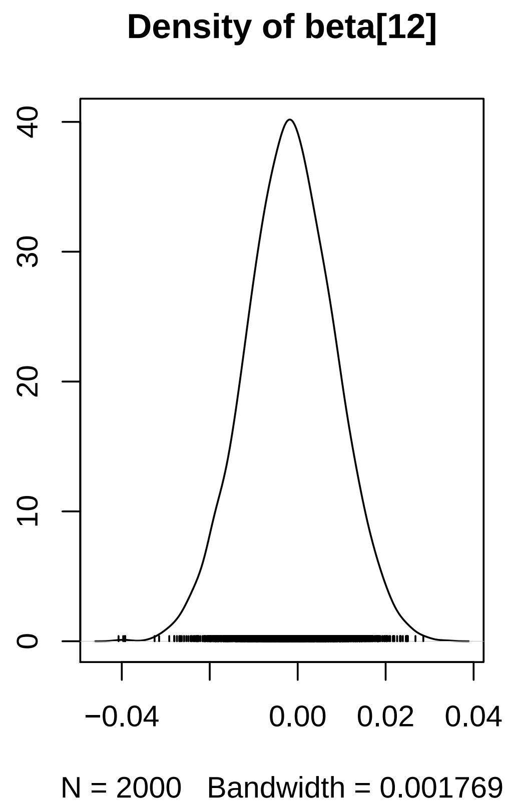

beta[12]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[12]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

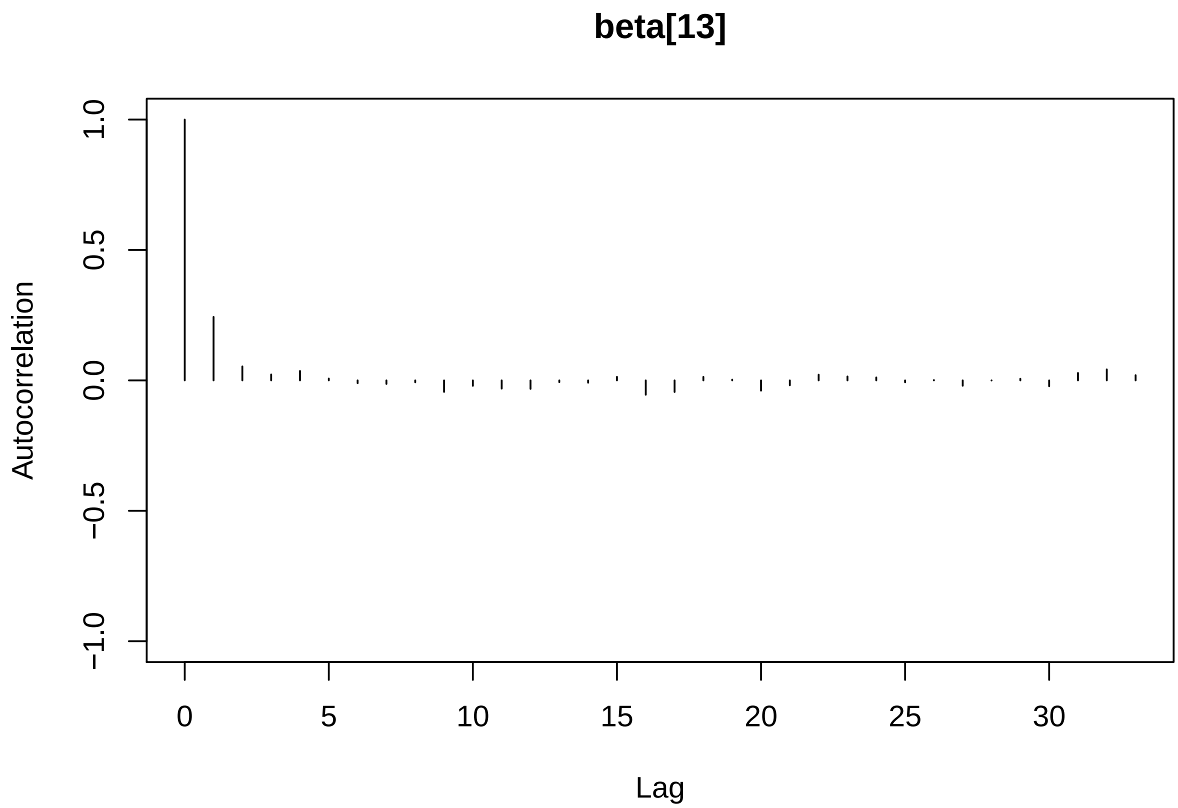

beta[12]
median
97.5%
shrink factor
1.00
1.01
1.02
1.03
1.04
33000
33500
34000
34500
35000
last iteration in chain

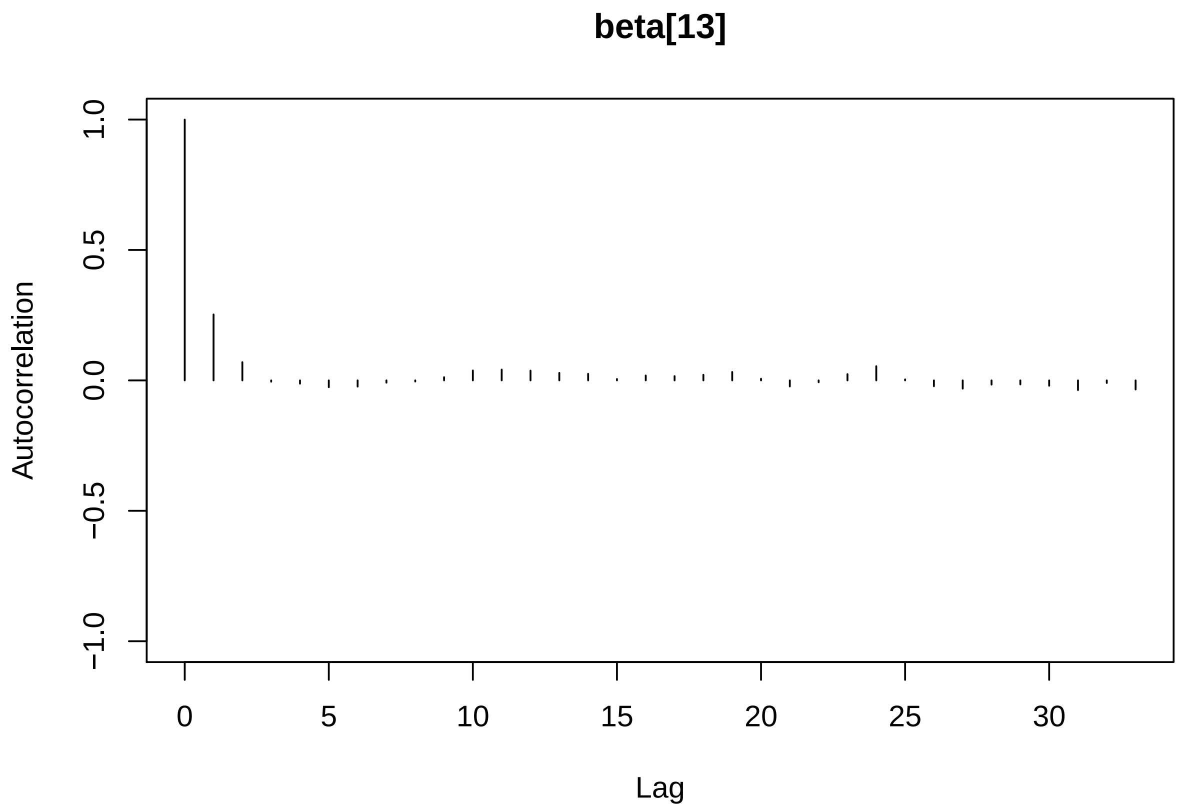


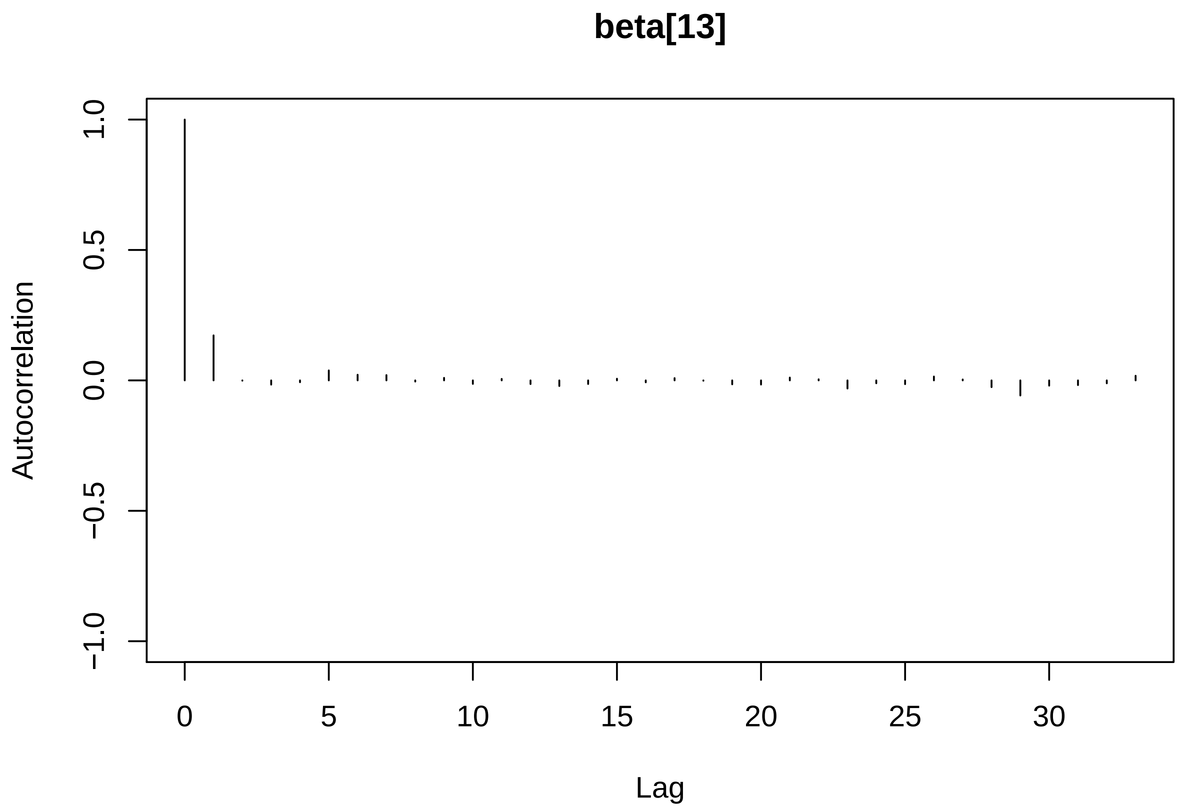
Trace of beta[12]
0.02
0.00
−0.02
−0.04
33000
34000
35000
Iterations

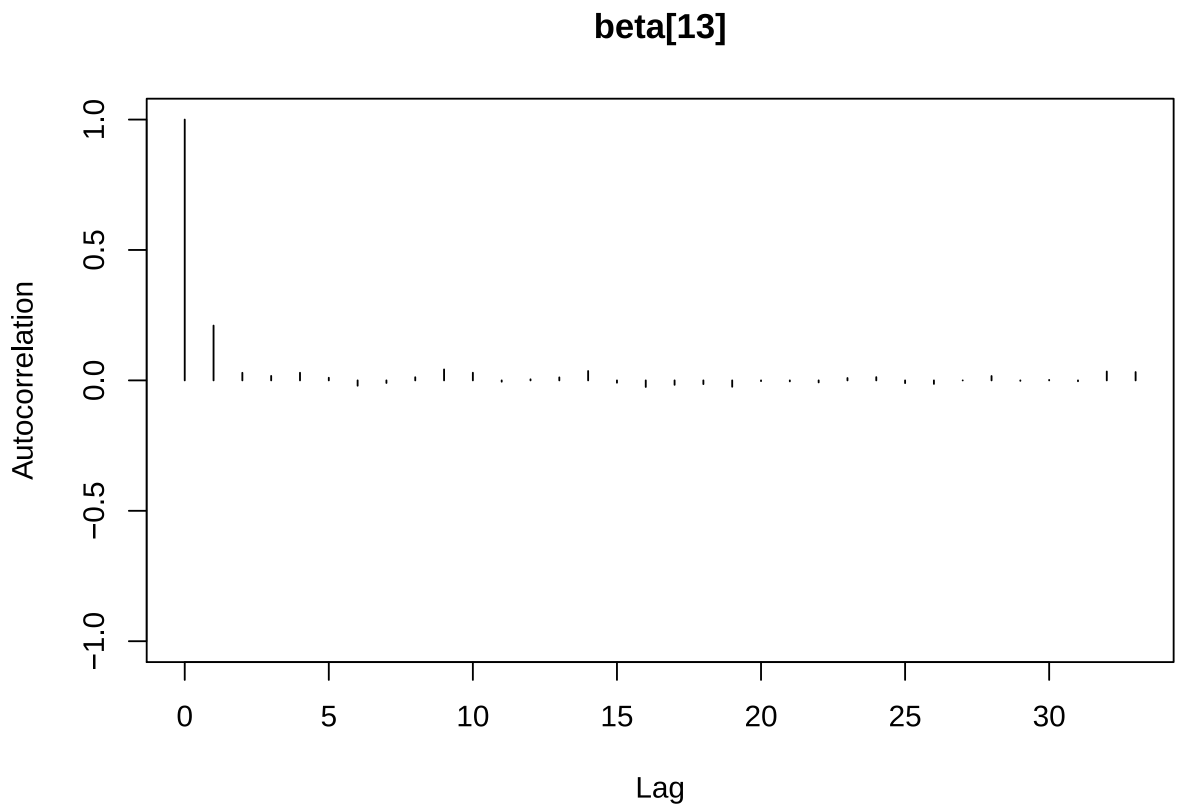


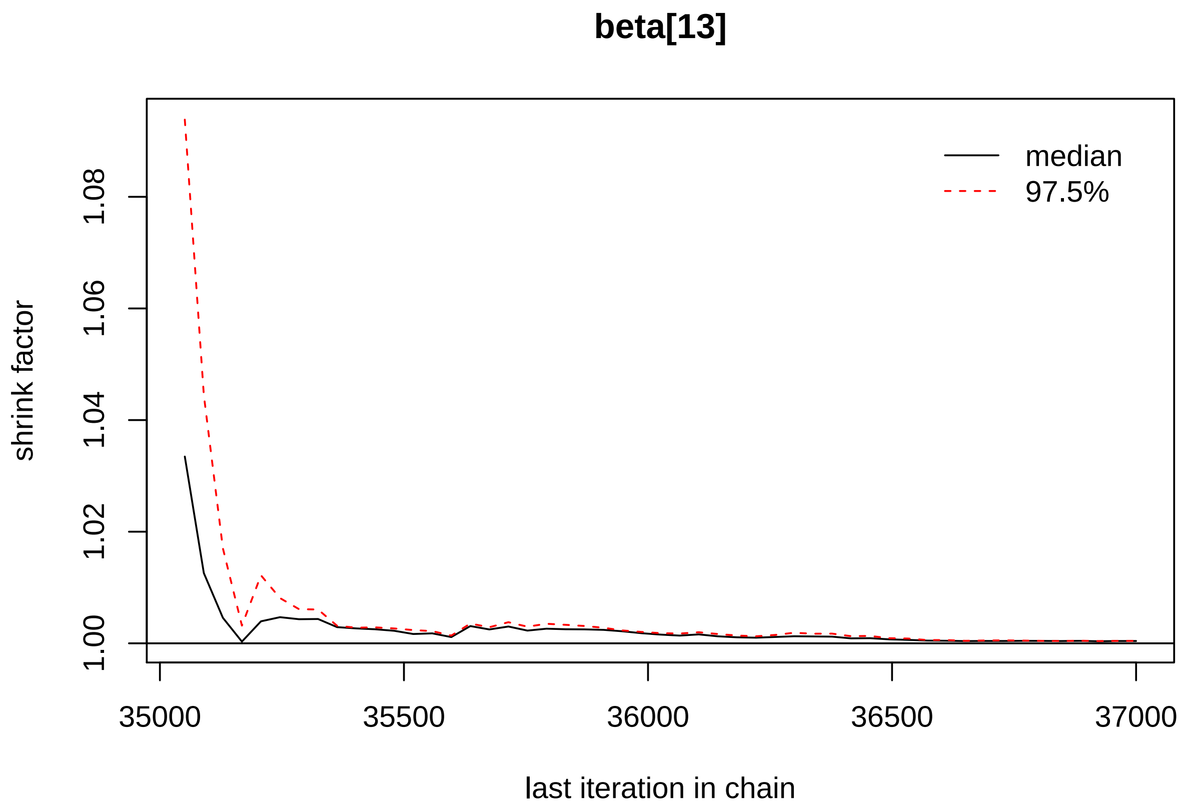
Density of beta[12]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001769

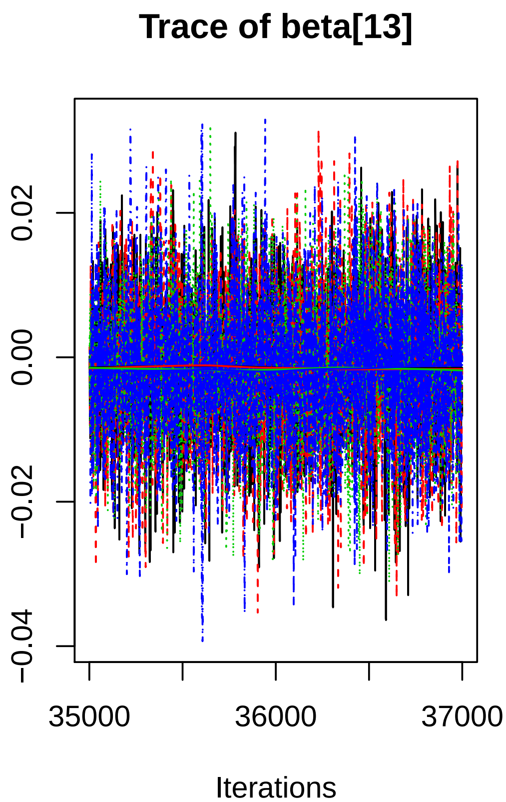

beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

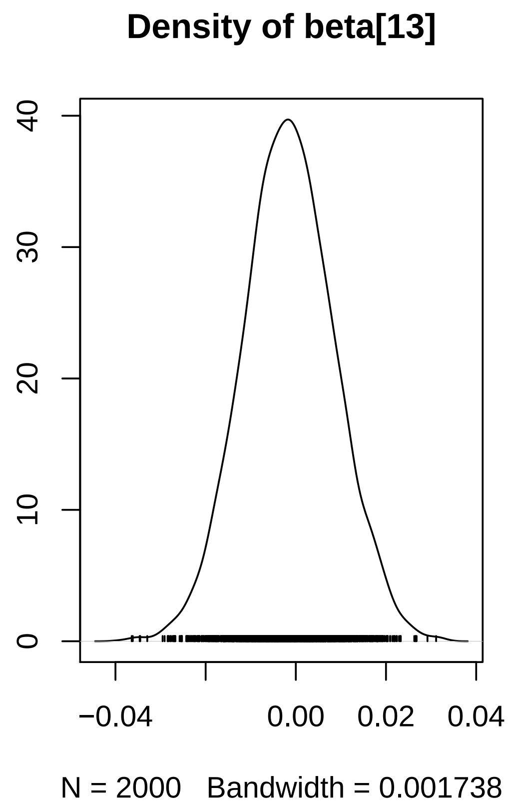

beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

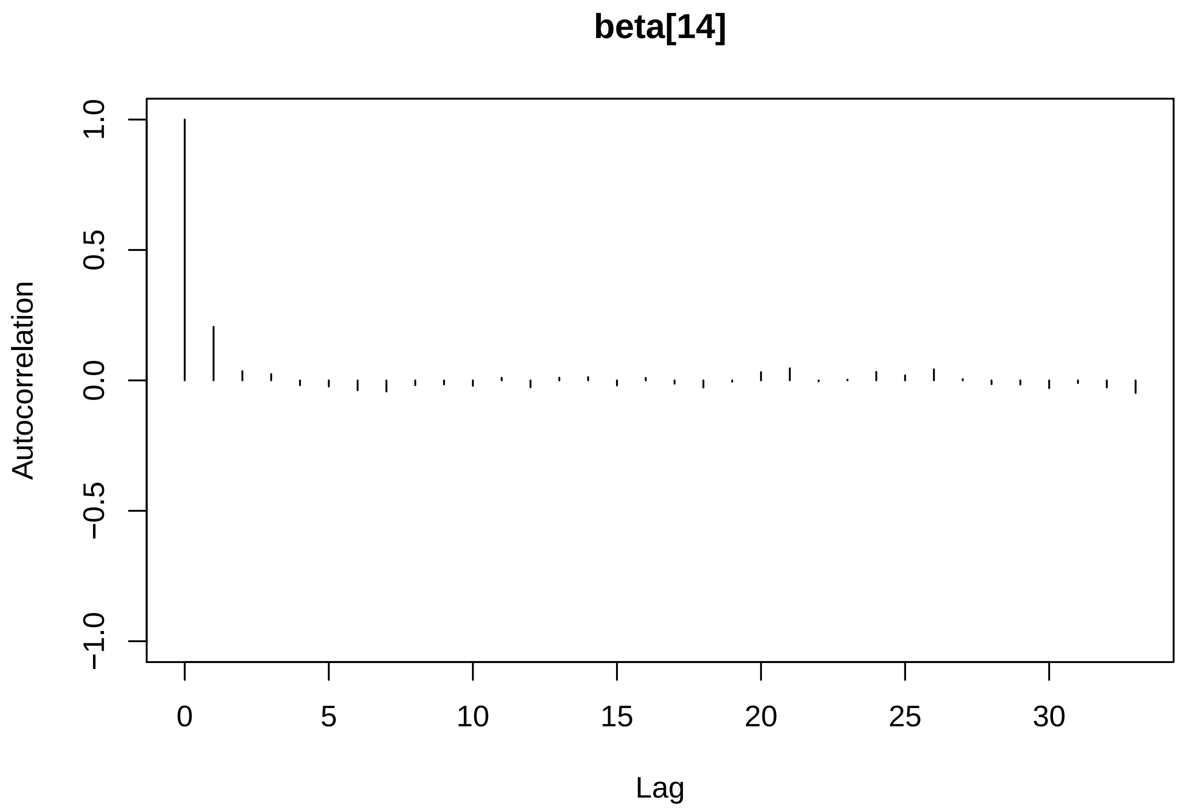

beta[13]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

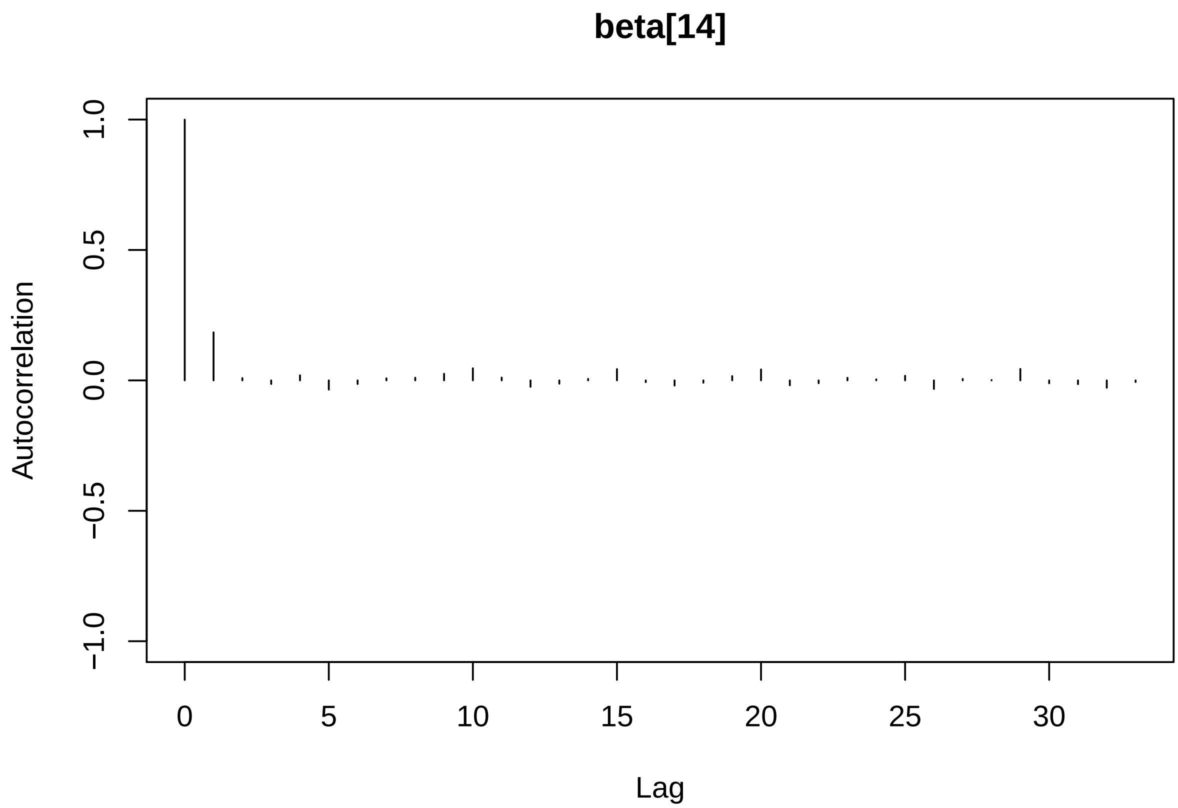

beta[13]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

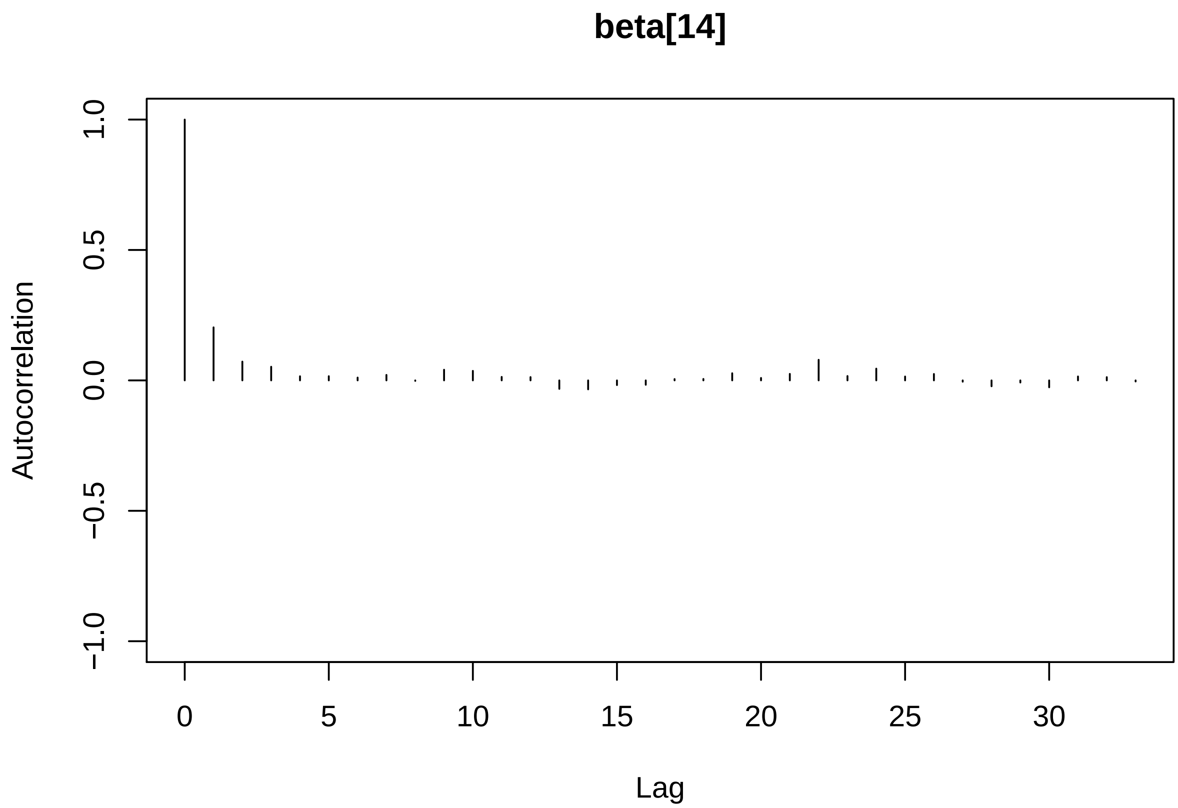

beta[13]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
35000
35500
36000
36500
37000
last iteration in chain

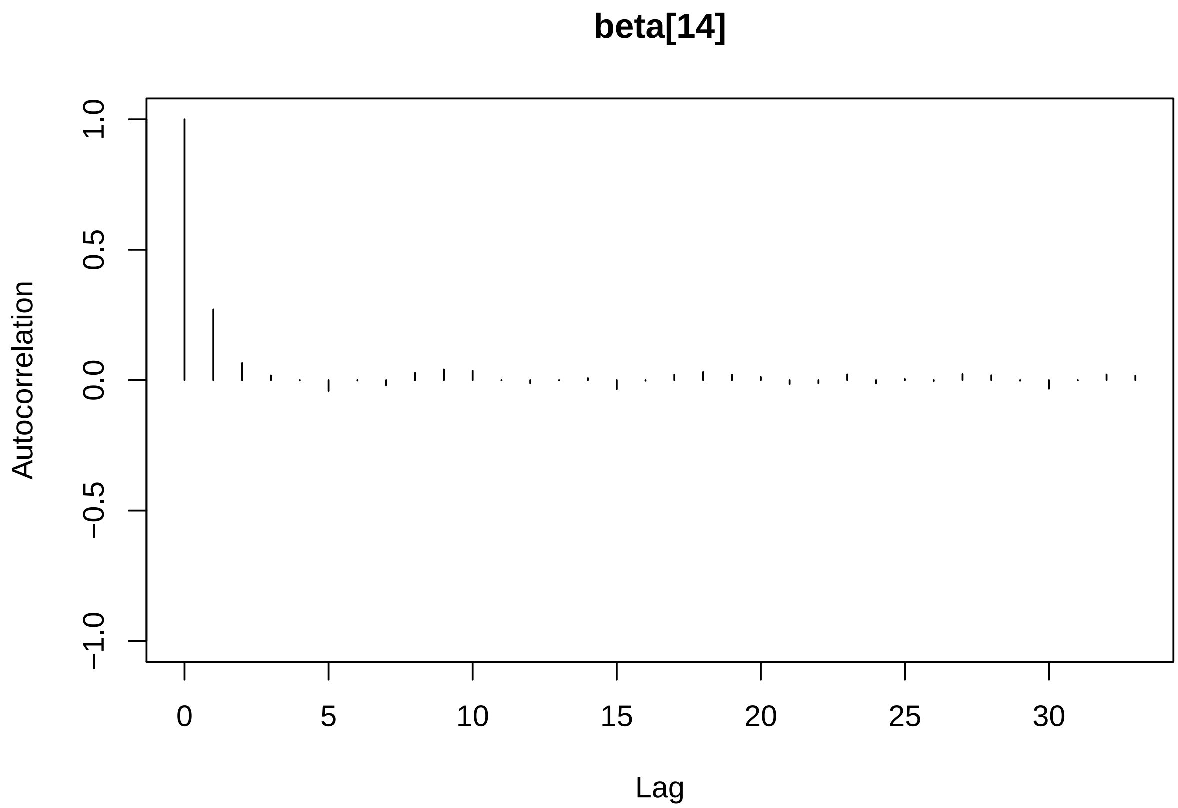

Trace of beta[13]
0.02
0.00
−0.02
−0.04
35000
36000
37000
Iterations

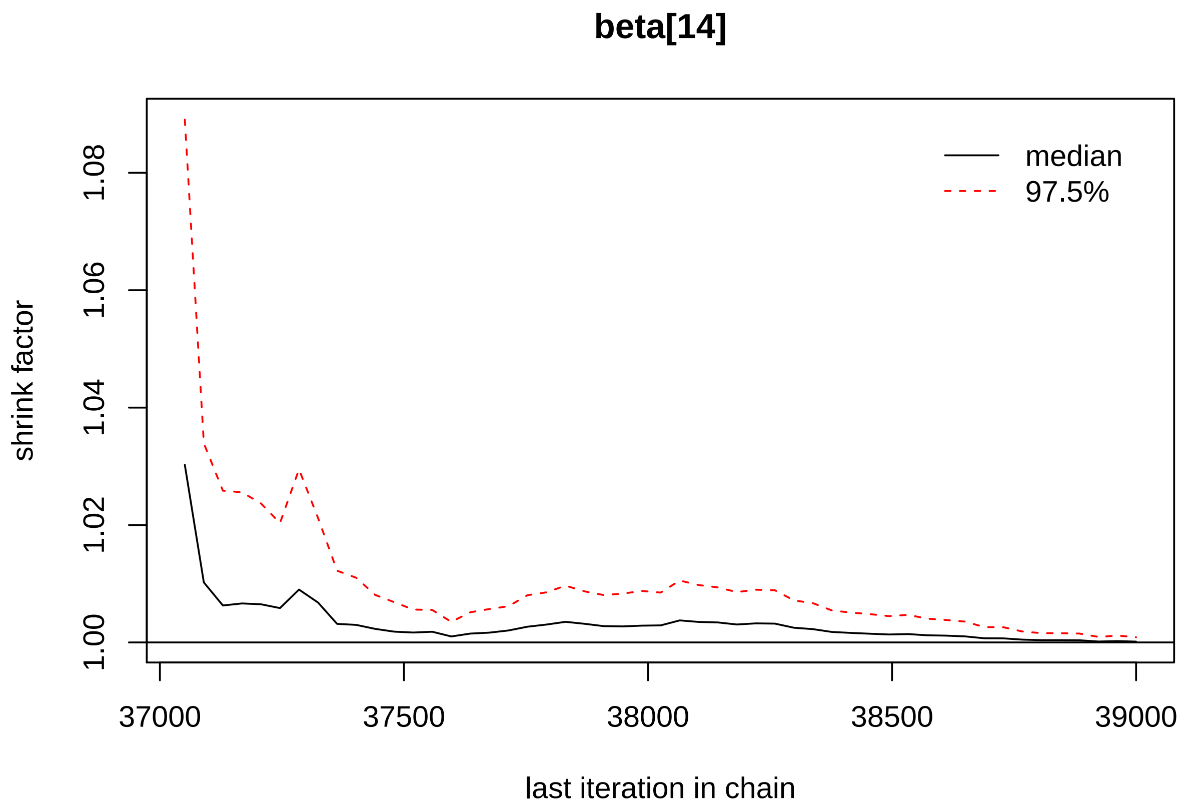

Density of beta[13]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.001738

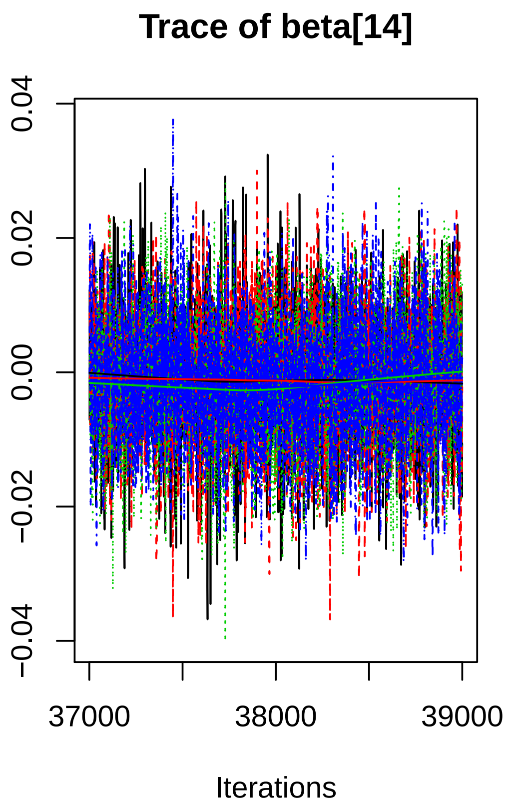

beta[14]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag


beta[14]
1.0
0.5
0.0
−0.5
−1.0
Autocorrelation
0
5
10
15
20
25
30
Lag

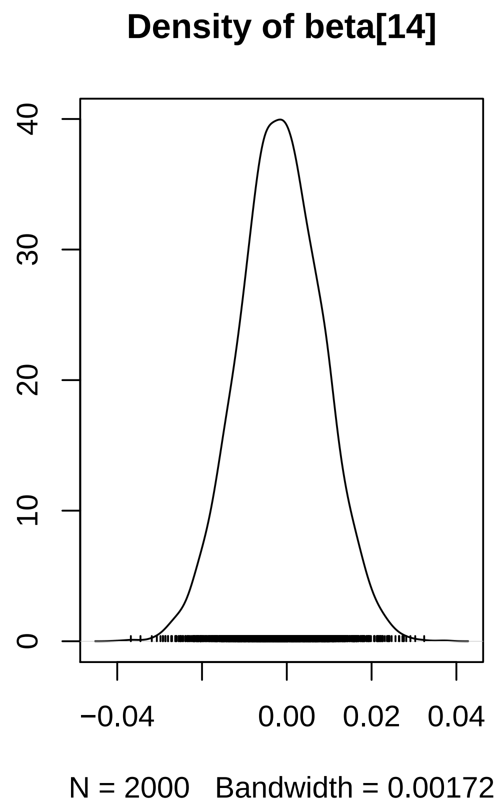

beta[14]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


beta[14]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

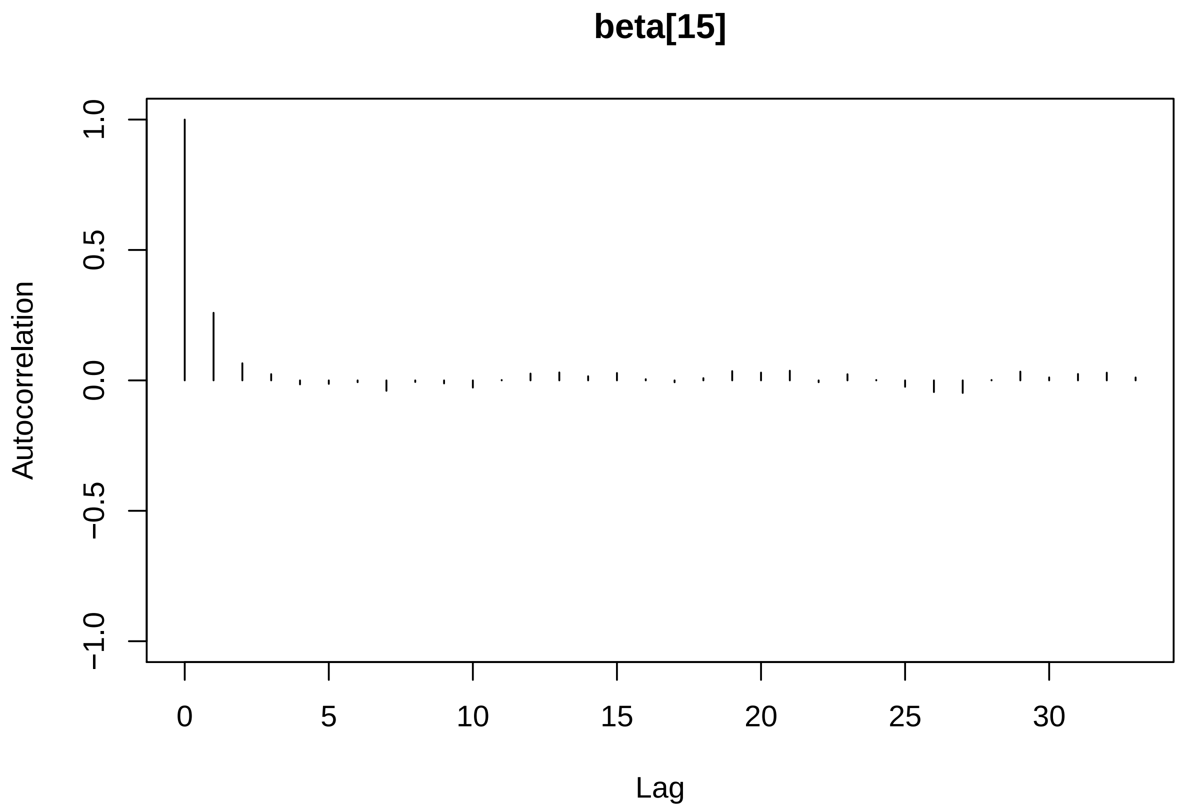

beta[14]
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
37000
37500
38000
38500
39000
last iteration in chain

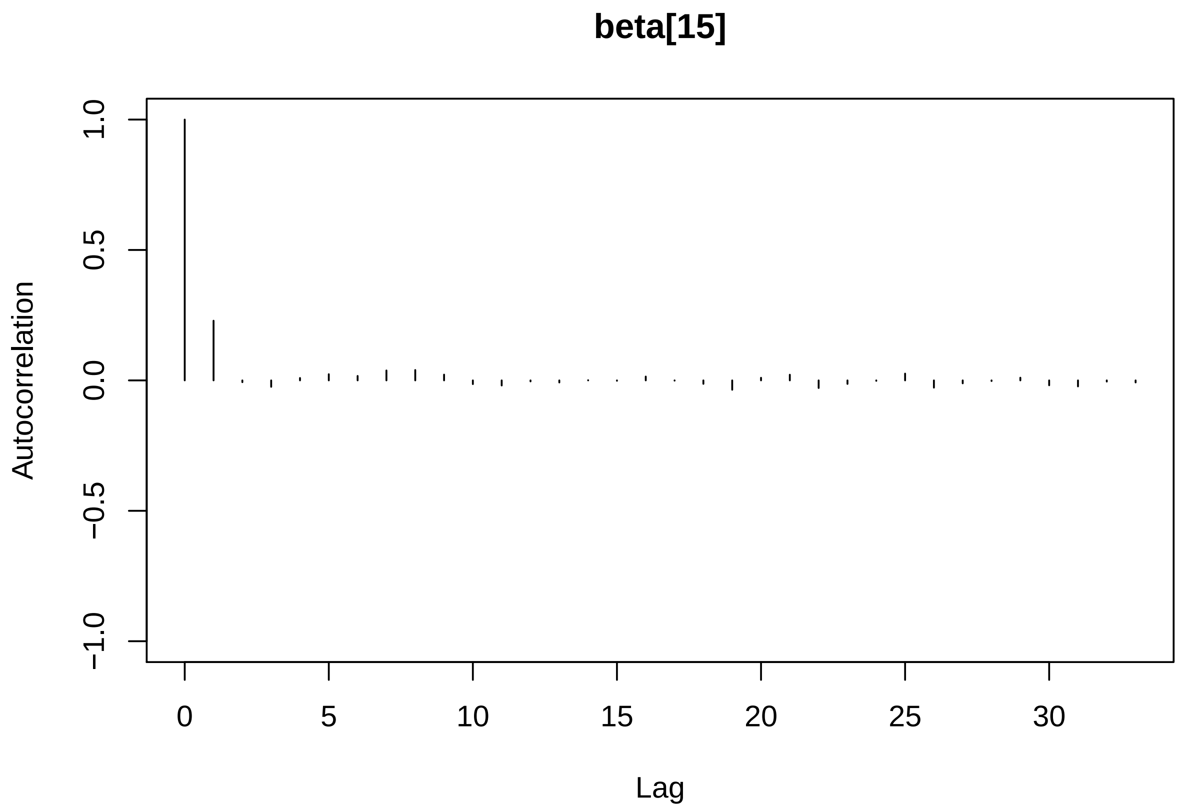


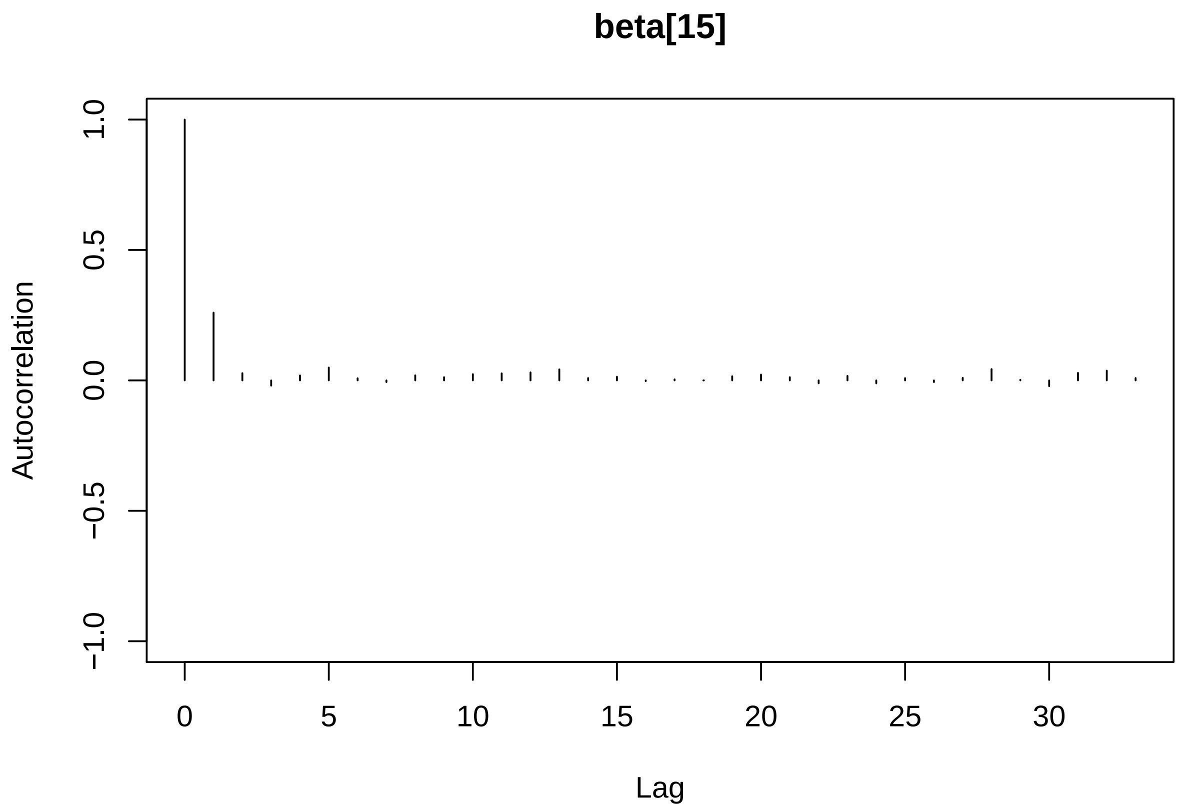
Trace of beta[14]
0.04
0.02
0.00
−0.02
−0.04
37000
38000
39000
Iterations

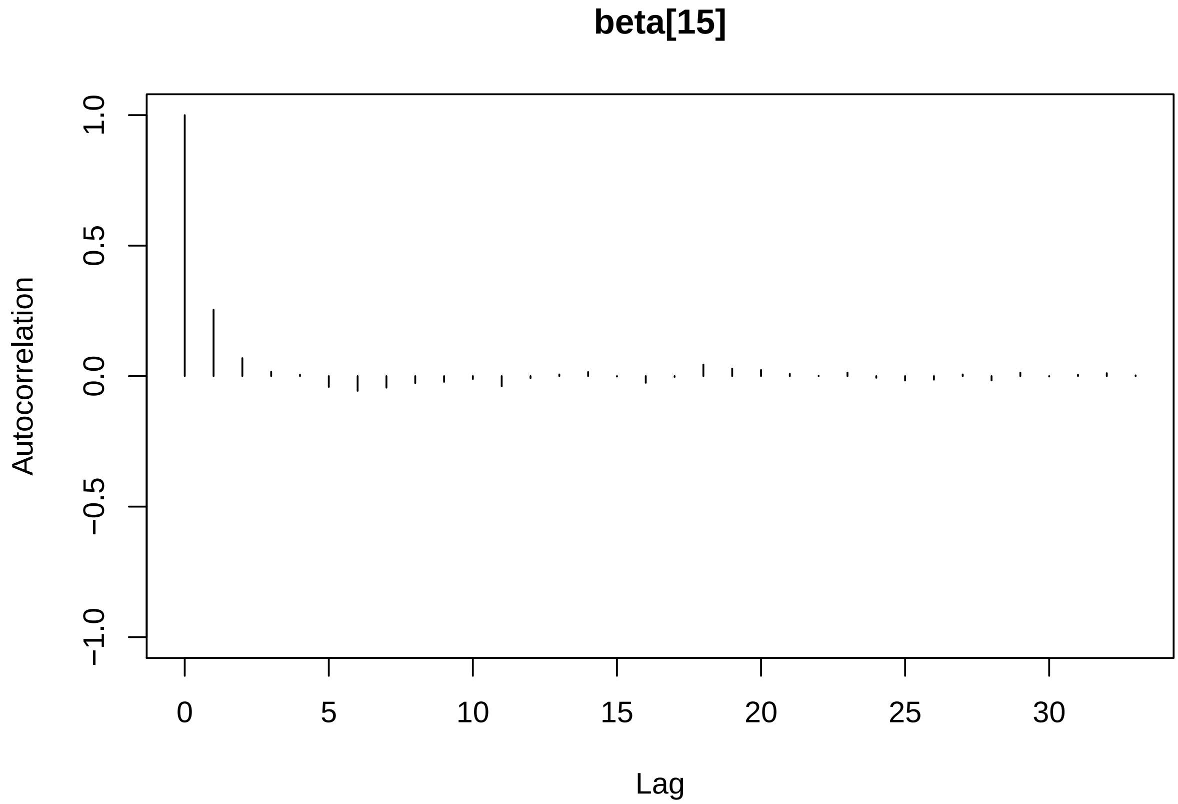


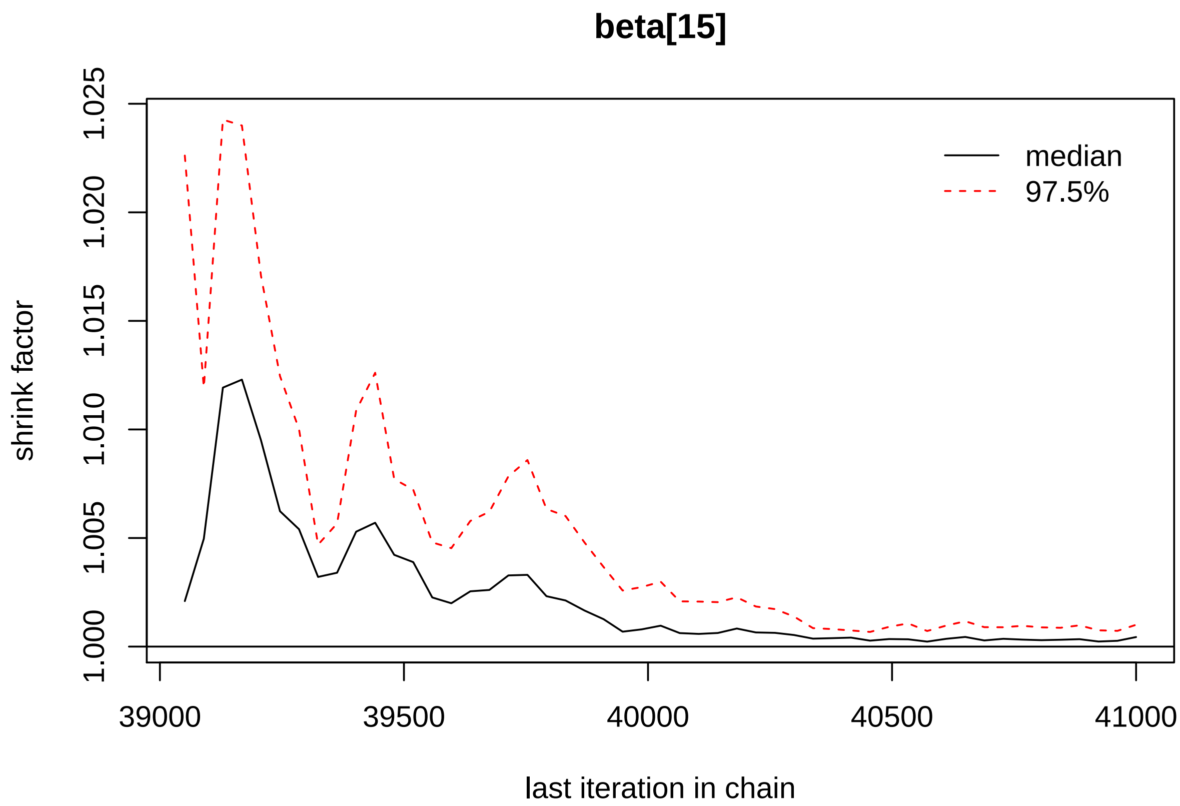
Density of beta[14]
40
30
20
10
0
−0.04
0.00
0.02
0.04
N = 2000 Bandwidth = 0.00172

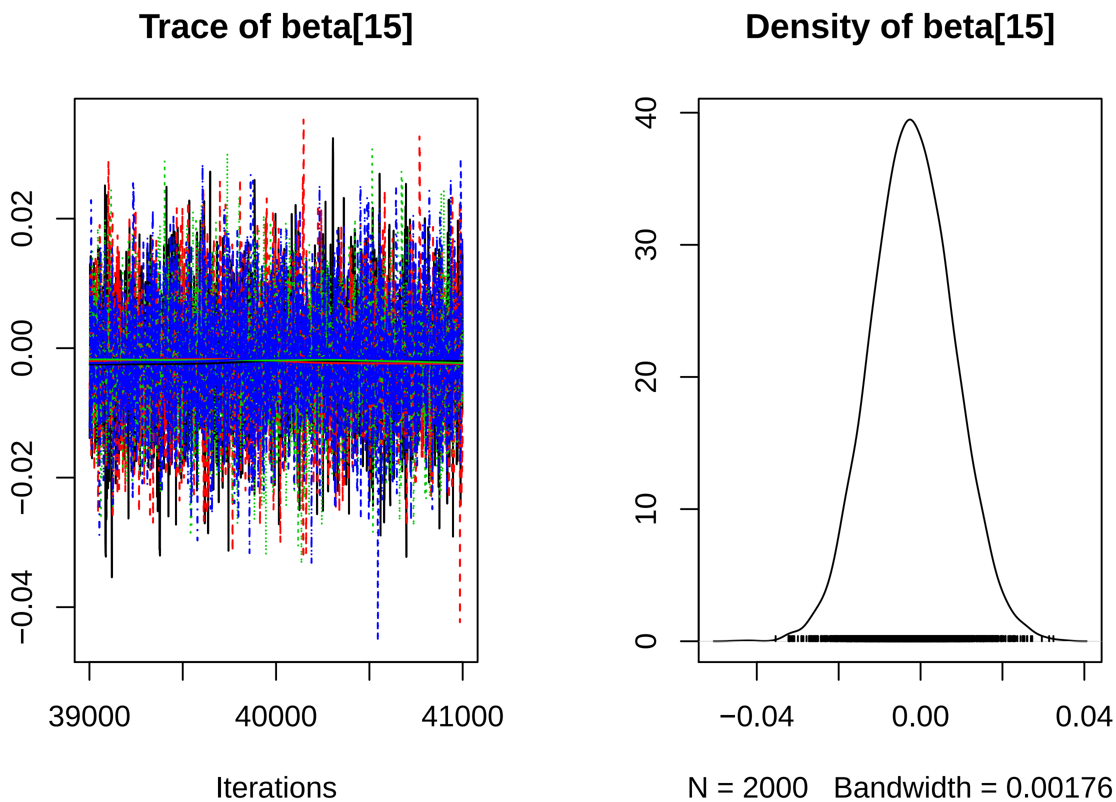

beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

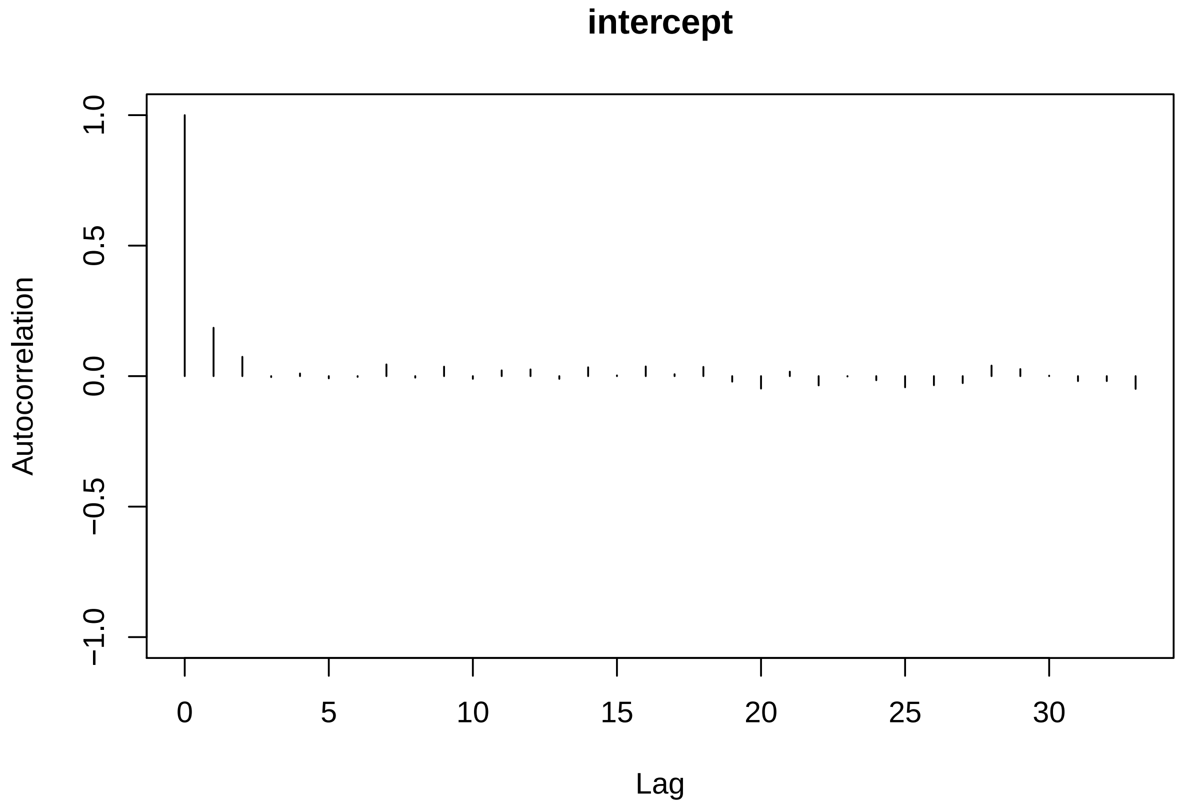

beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

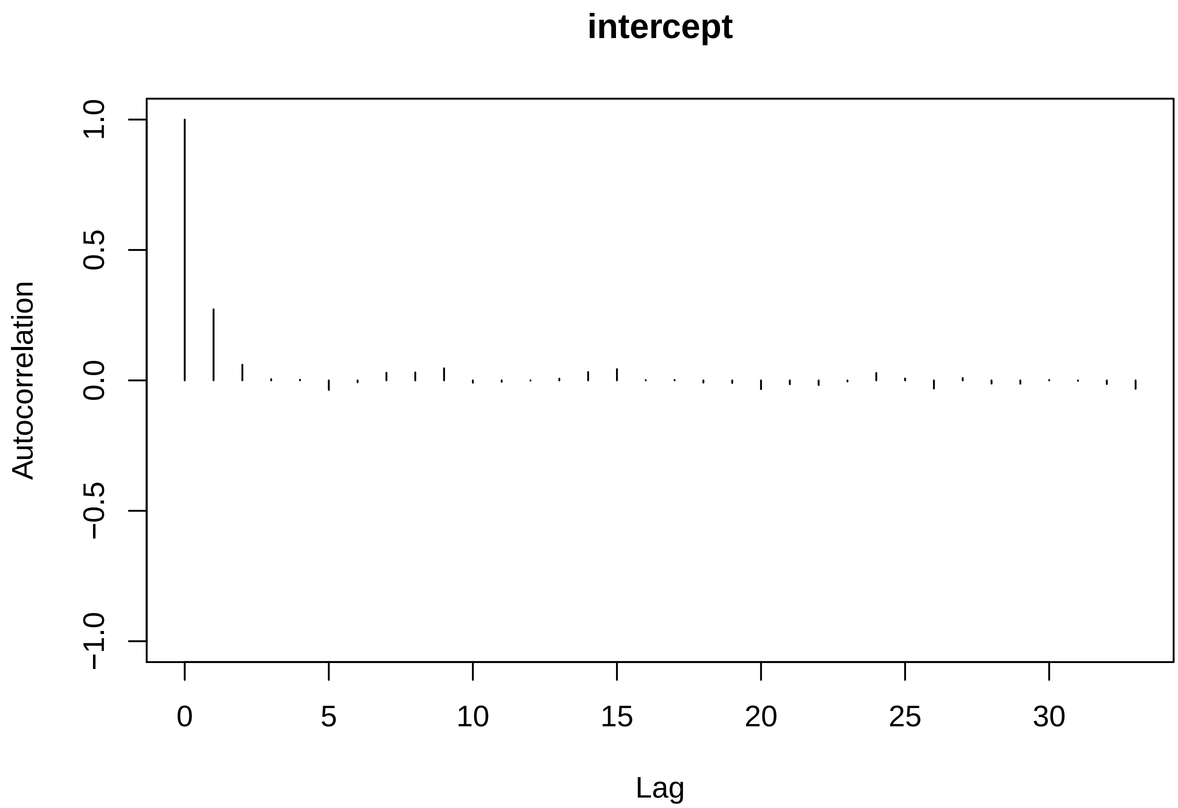

beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

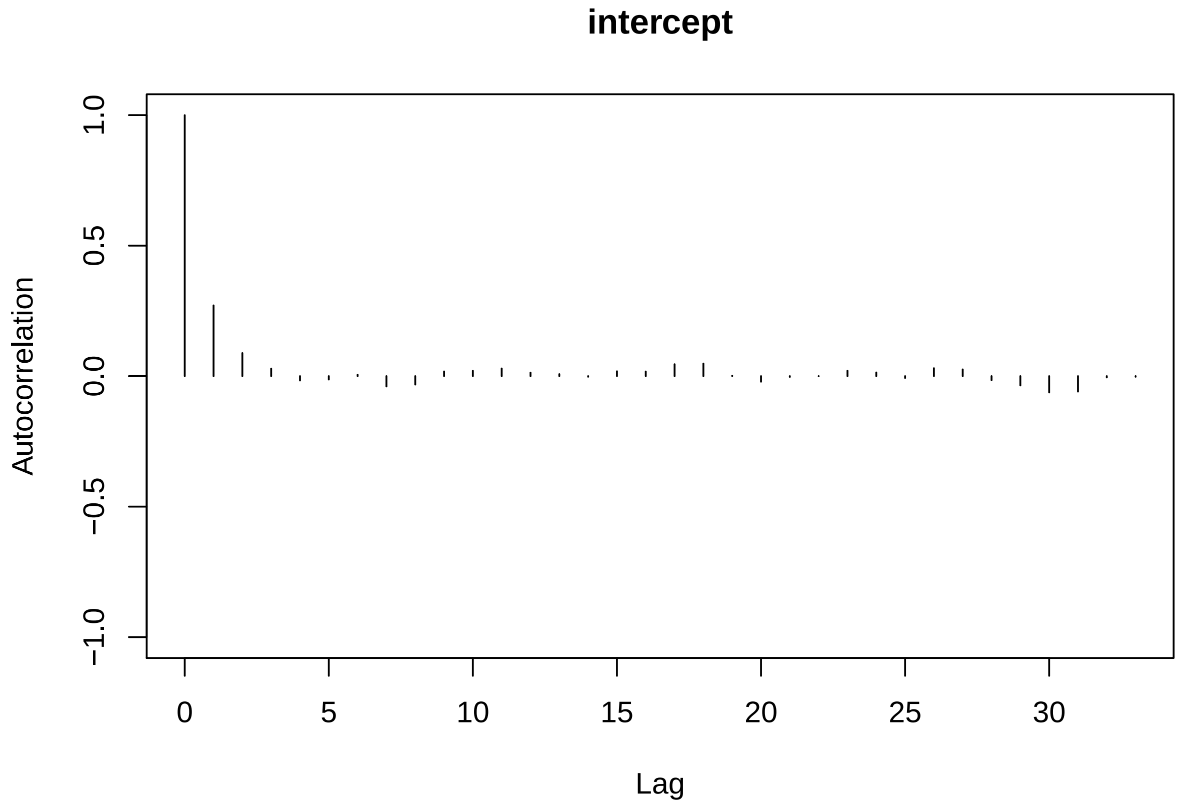

beta[15]
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag

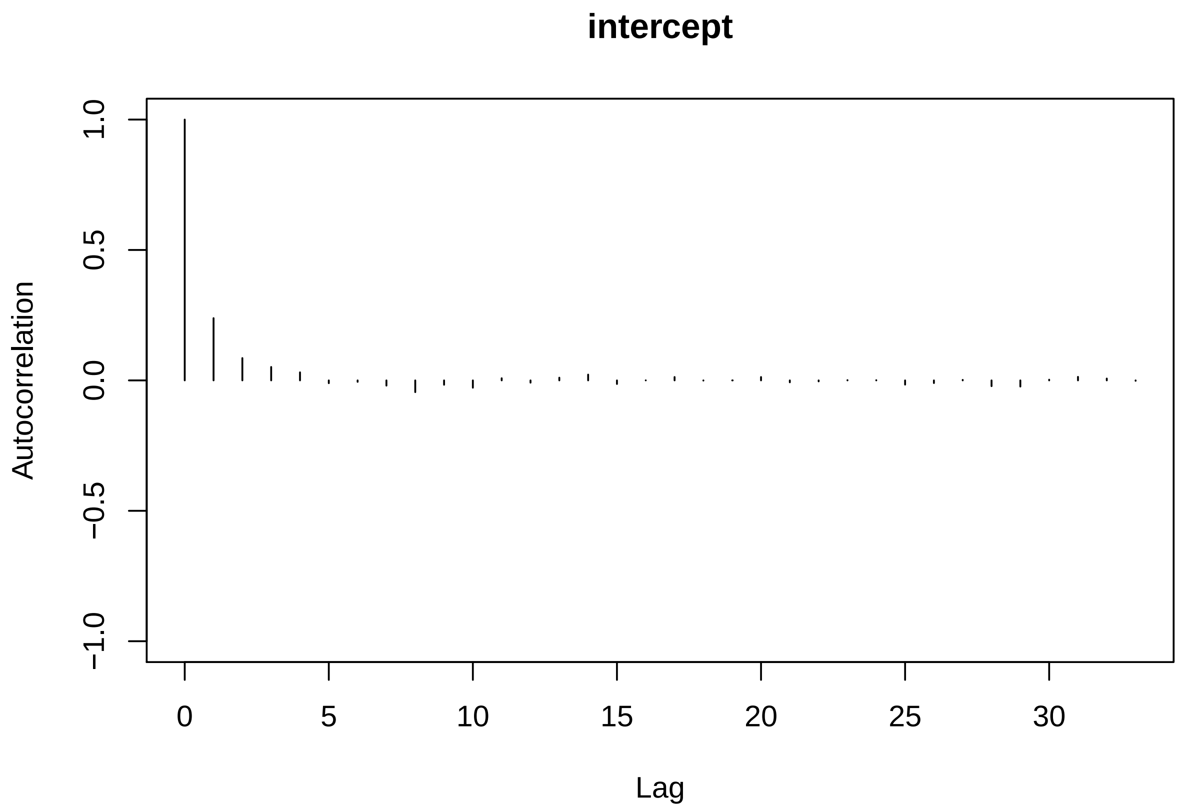

beta[15]
shrink factor
1.000
1.005
1.010
1.015
1.020
1.025
39000
39500
40000
40500
41000
last iteration in chain
median
97.5%

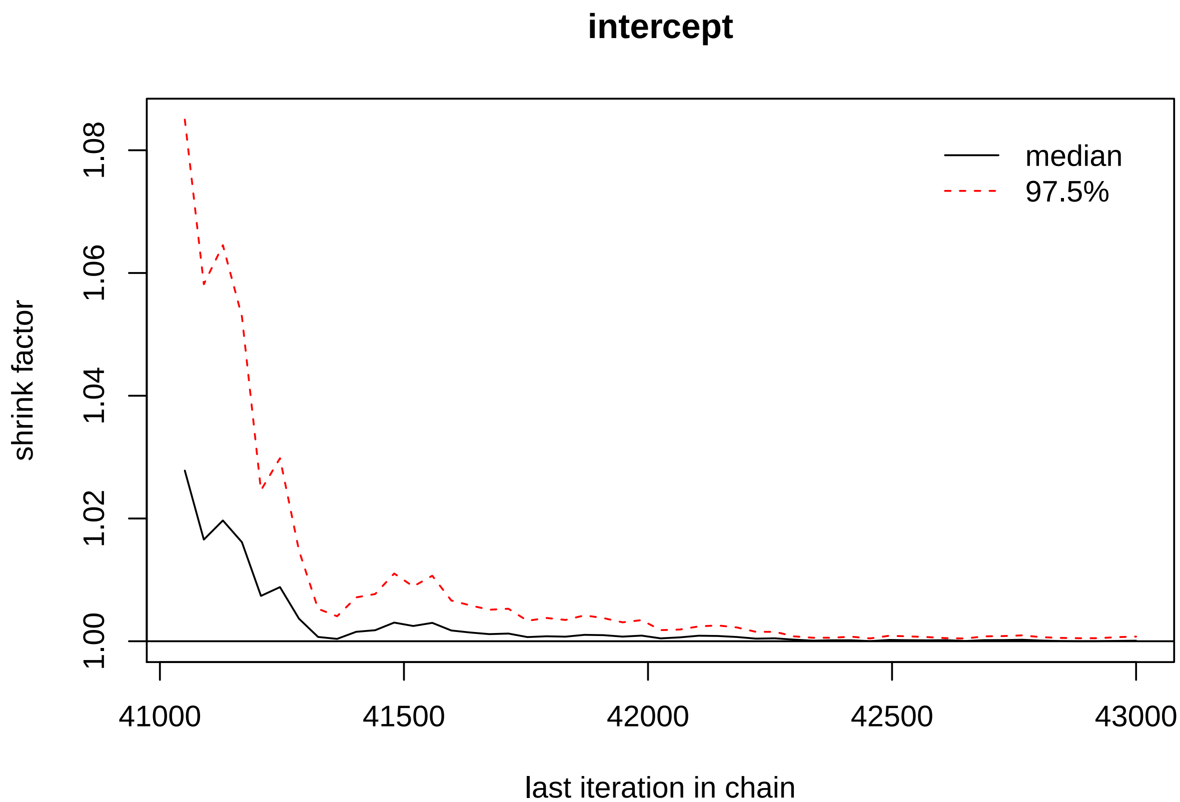

Trace of beta[15]
0.02
0.00
−0.02
−0.04
39000
40000
41000
Iterations
Density of beta[15]
40
30
20
10
0
−0.04
0.00
0.04
N = 2000 Bandwidth = 0.00176

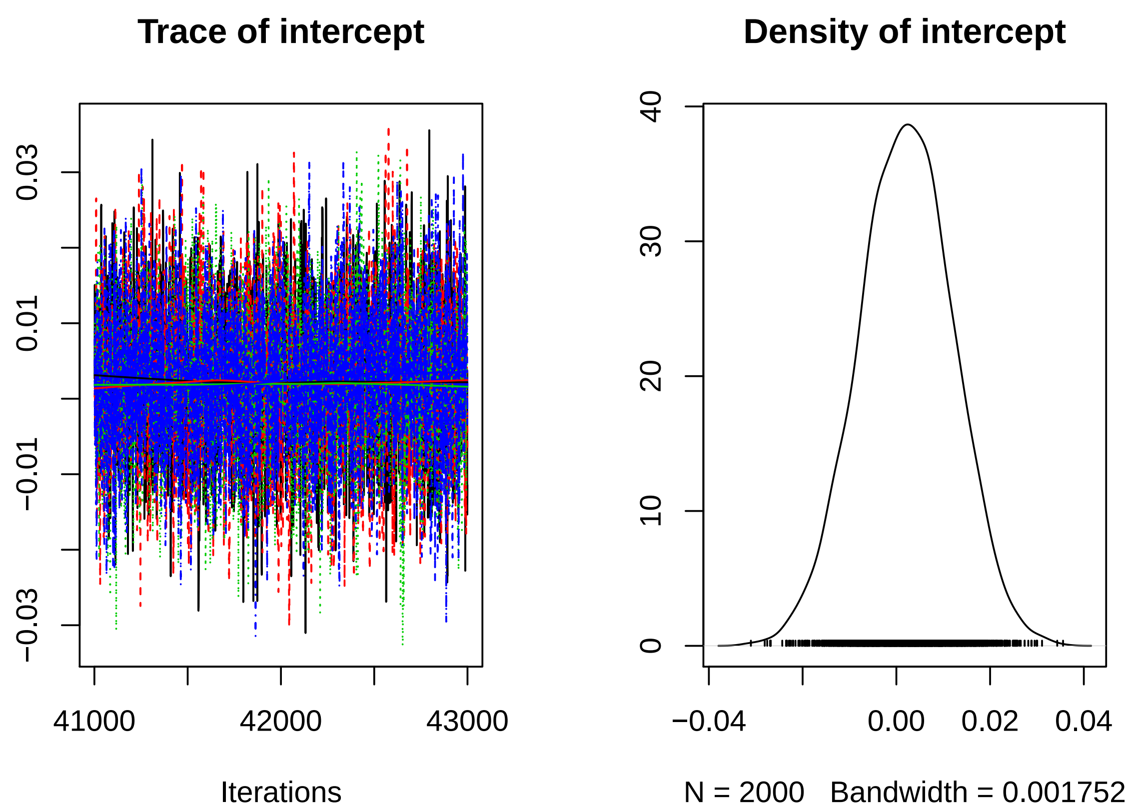

intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
Lag


intercept
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
41000
41500
42000
42500
43000
last iteration in chain



Fit frequentist logistic model for reference.

```
df <- data.frame(cbind(Z, X))
lm.logistic <- glm(Z ~ ., family = binomial,
    df)
summary(lm.logistic)
```

```
##
## Call:
## glm(formula = Z ~ ., family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1986  -0.4521   0.1467   0.4048   3.0622
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.22344    0.33351   3.668 0.000244 ***
## PST120214    1.44242    0.68408   2.109 0.034984 *
## AGE775214   -0.81589    0.46258  -1.764 0.077771 .
## RHI225214   -2.05358    0.48616  -4.224  2.4e-05 ***
```

```
## RHI725214   -1.00849    0.37201  -2.711 0.006709 **
## EDU635213   -0.55048    0.64123  -0.858 0.390634
## EDU685213   -1.99593    0.88355  -2.259 0.023883 *
## HSG445213    0.96082    0.50725   1.894 0.058199 .
## HSG495213    0.87243    0.81258   1.074 0.282975
## INC110213   -0.97476    0.91641  -1.064 0.287475
## PVY020213   -0.36784    0.64185  -0.573 0.566582
## RTN131207   -0.44836    0.41945  -1.069 0.285097
## POP060210   -0.03261    0.76264  -0.043 0.965888
## VET605213    0.46278    0.81324   0.569 0.569321
## MAN450207    0.46521    0.69222   0.672 0.501550
## WTN220207   -1.01084    0.93321  -1.083 0.278726
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.144  on 145  degrees of freedom
## Residual deviance:  93.976  on 130  degrees of freedom
## AIC: 125.98
##
## Number of Fisher Scoring iterations: 6
```