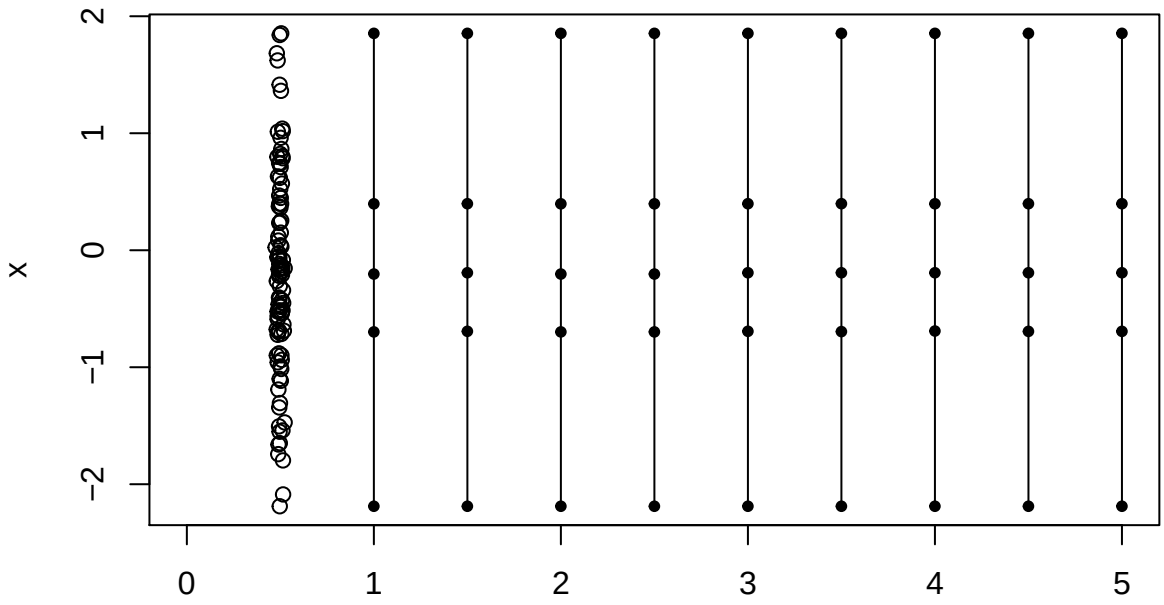
185 HW2

Yuetong Lyu

4/20/2019

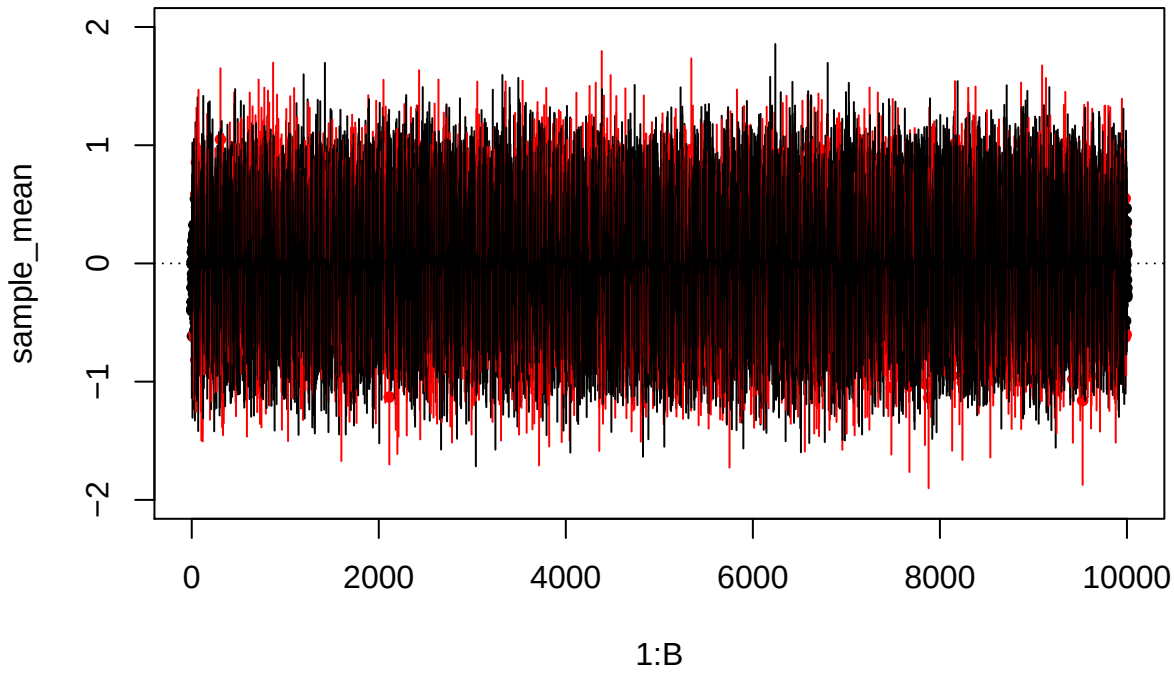
```
#Problem 1
x <- rnorm(100)
qtl1 <- quantile(x,type = 1)
plot(rep(0.5,100) + rnorm(100,sd = 0.01), x, xlim = c(0,5), ylim = c(qtl1[1],qtl1[5]))

segments(1,qtl1[1],1,qtl1[5])
points(rep(1,5),qtl1,pch = 20)

qtl2 <- quantile(x,type = 2)
segments(1.5,qtl2[1],1.5,qtl2[5])
points(rep(1.5,5),qtl2,pch = 20)

qtl3 <- quantile(x,type = 3)
segments(2,qtl3[1],2,qtl3[5])
points(rep(2,5),qtl3,pch = 20)

qtl4 <- quantile(x,type = 4)
segments(2.5,qtl4[1],2.5,qtl4[5])
points(rep(2.5,5),qtl4,pch = 20)

qtl5 <- quantile(x,type = 5)
segments(3,qtl5[1],3,qtl5[5])
points(rep(3,5),qtl5,pch = 20)

qtl6 <- quantile(x,type = 6)
segments(3.5,qtl6[1],3.5,qtl6[5])
points(rep(3.5,5),qtl6,pch = 20)

qtl7 <- quantile(x,type = 7)
segments(4,qtl7[1],4,qtl7[5])
points(rep(4,5),qtl7,pch = 20)

qtl8 <- quantile(x,type = 8)
segments(4.5,qtl8[1],4.5,qtl8[5])
points(rep(4.5,5),qtl8,pch = 20)

qtl9 <- quantile(x,type = 9)
segments(5,qtl9[1],5,qtl9[5])
points(rep(5,5),qtl9,pch = 20)
```



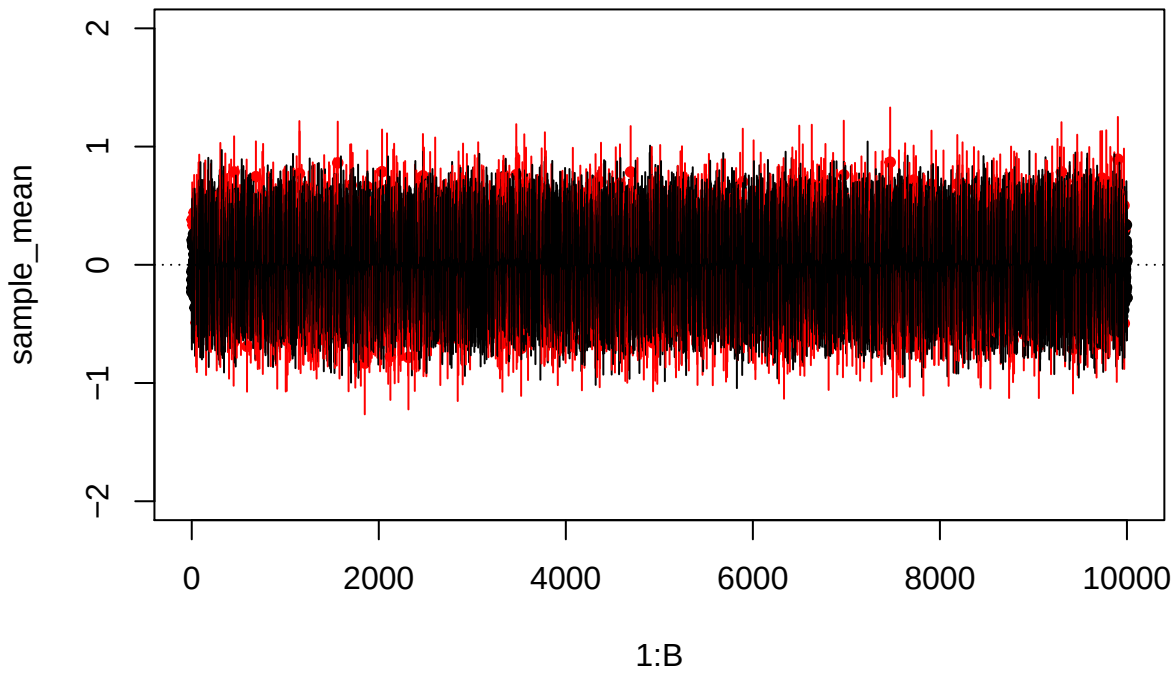

```r
#Problem 2
n = 10
for(i in 1:10){
  alpha = 0.10
  B     = 10000
  sample_mean   = numeric(B)
  conf_upper = numeric(B)
  conf_lower = numeric(B)
  num_success <- 0

  for (j in 1:B) {
    x <- rnorm(n)
    out = t.test(x, conf=1-alpha)
    conf_lower[j] = out$conf.int[1]
    conf_upper[j] = out$conf.int[2]
    sample_mean[j]   = out$estimate
    if((conf_lower[j] < 0) & (0 < conf_upper[j])){
    num_success <- num_success + 1
   }
  }
  frac_contain_true_mean <- num_success/B
  print(paste("fraction of true mean =",frac_contain_true_mean))
  contains_mean = (conf_lower < 0) & (0 < conf_upper) # T/F
  color_vec     = ifelse(contains_mean == TRUE, 1, 2) #if true,1;if false,2
  plot(1:B, sample_mean, pch=20, ylim = c(-2, 2), col=color_vec)
  segments(1:B, conf_lower, 1:B, conf_upper, col=color_vec)
```

```
  abline(h=0, lty="dotted") #or solid

  n = n + 10
}
```

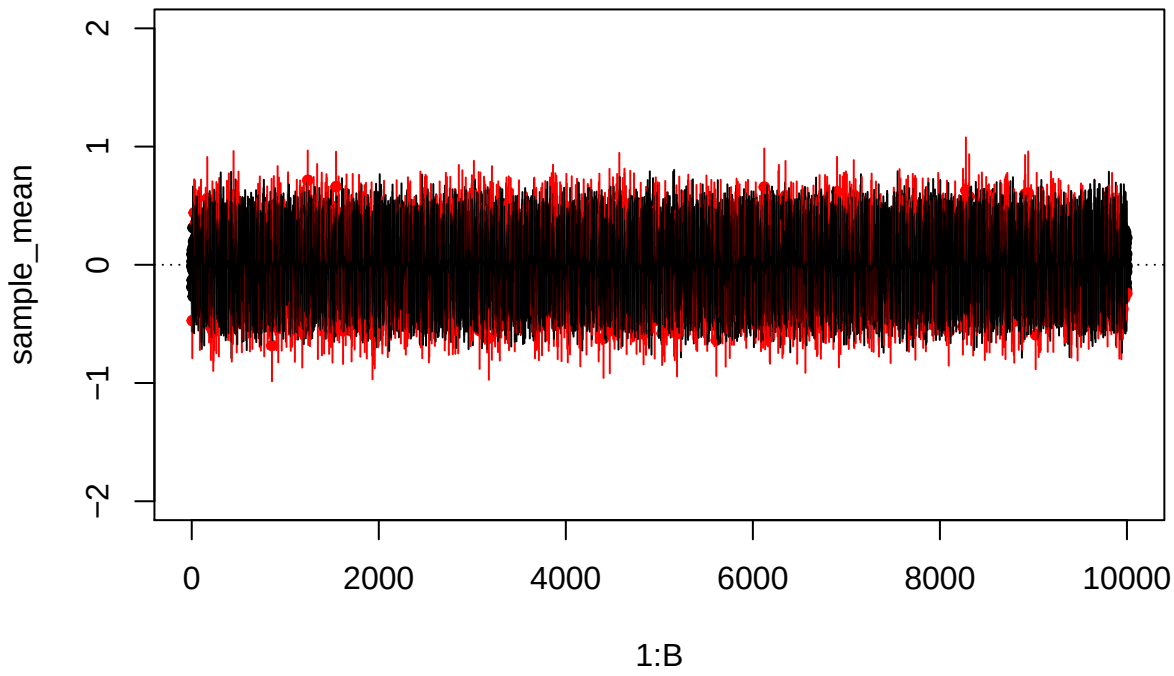
```
## [1] "fraction of true mean = 0.9008"
```



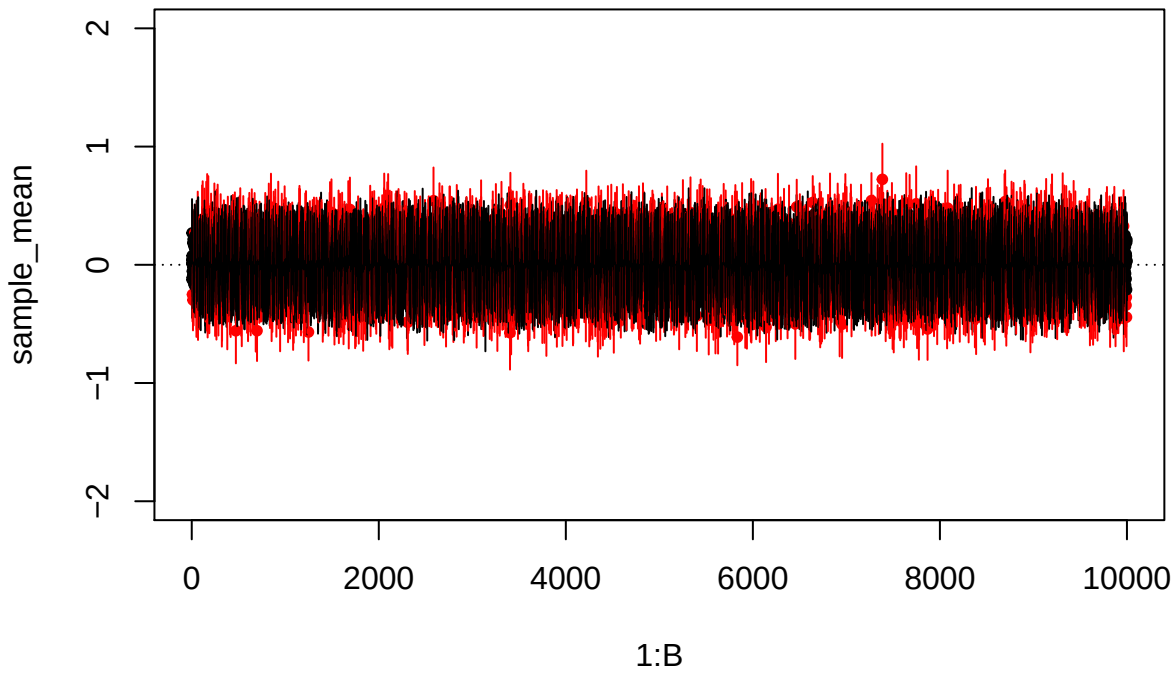

```
## [1] "fraction of true mean = 0.8988"
```



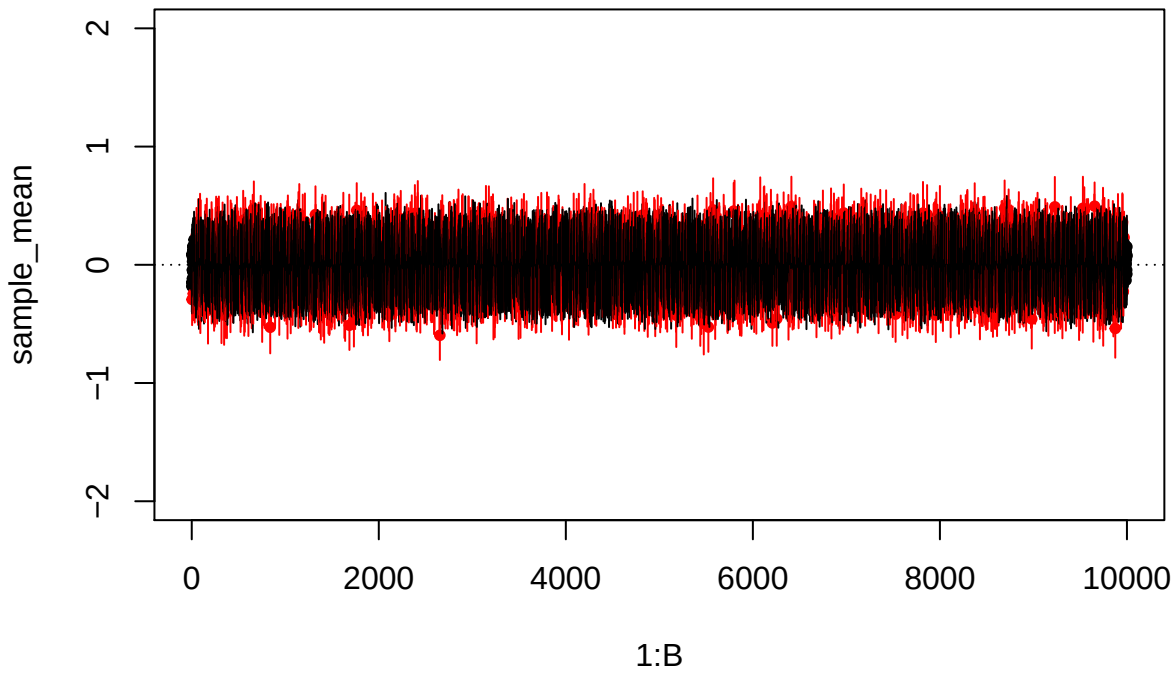

```
## [1] "fraction of true mean = 0.9013"
```



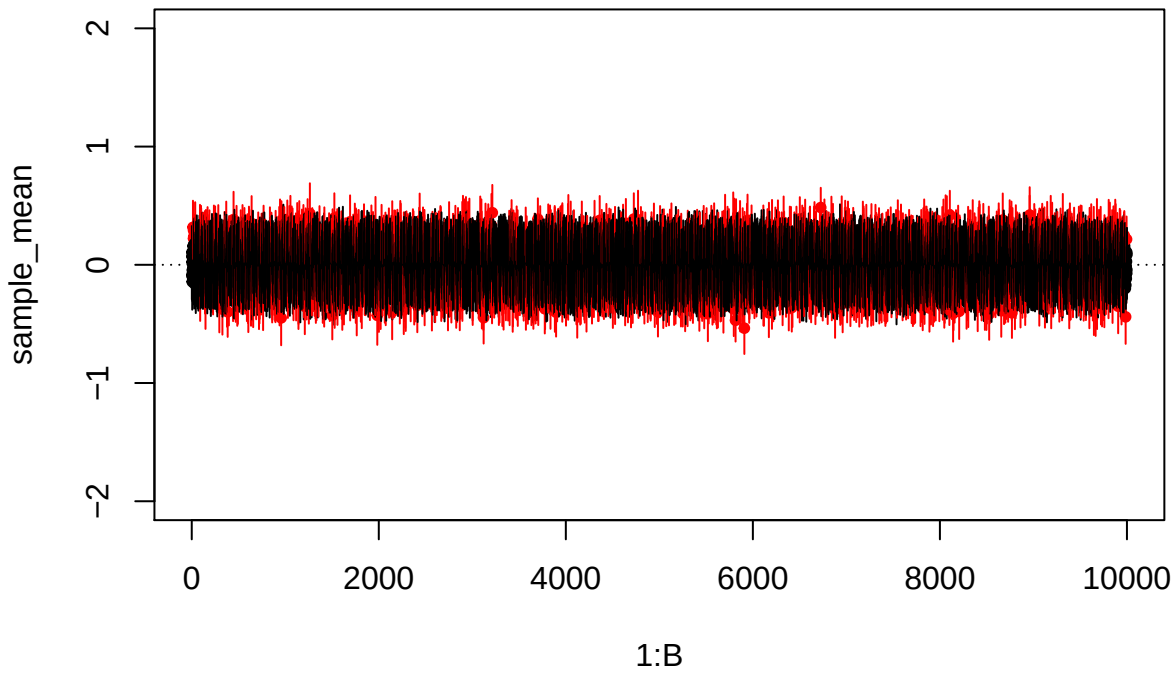

```
## [1] "fraction of true mean = 0.8936"
```



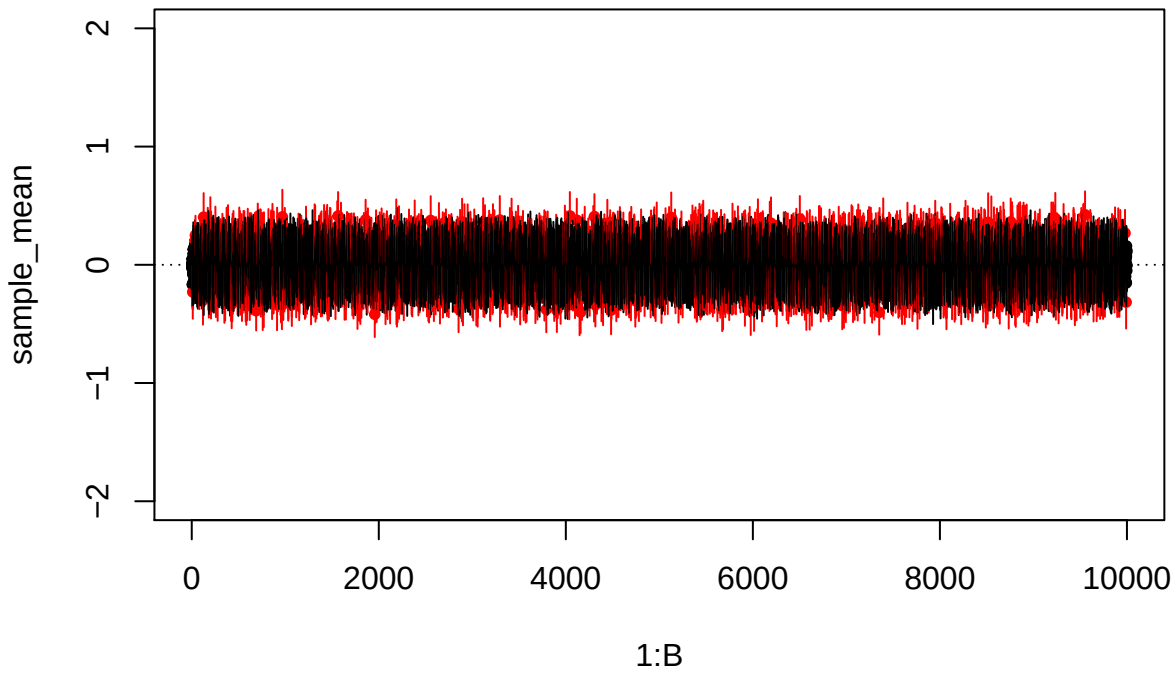

```
## [1] "fraction of true mean = 0.9003"
```



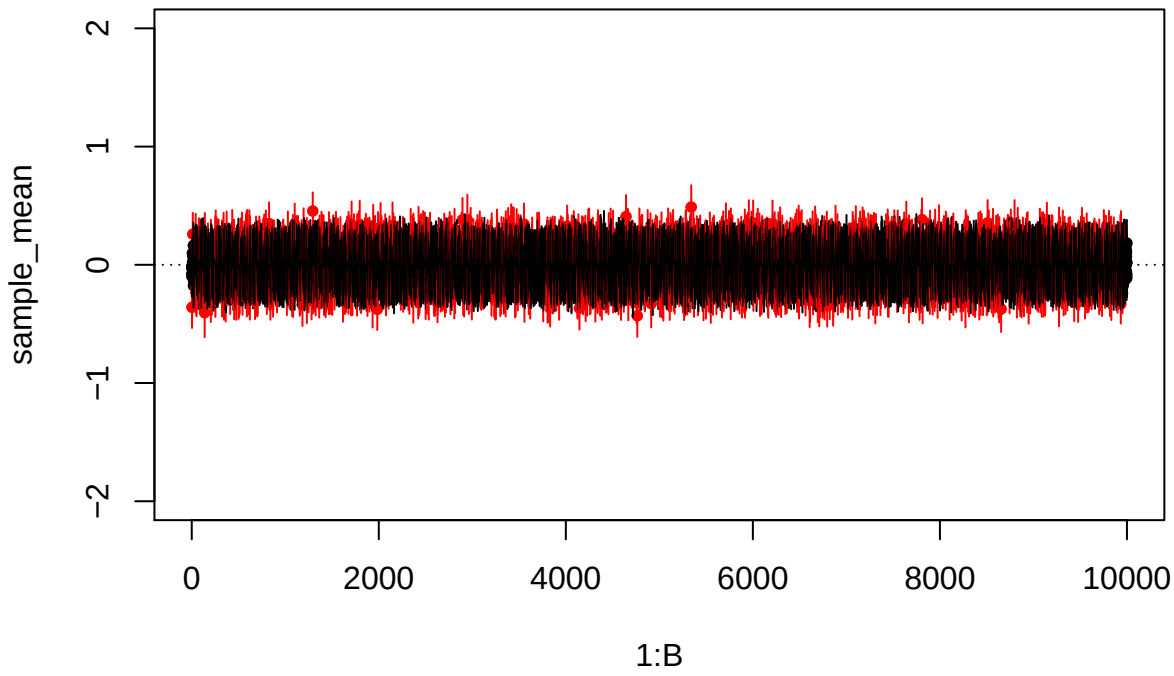

```
## [1] "fraction of true mean = 0.8992"
```



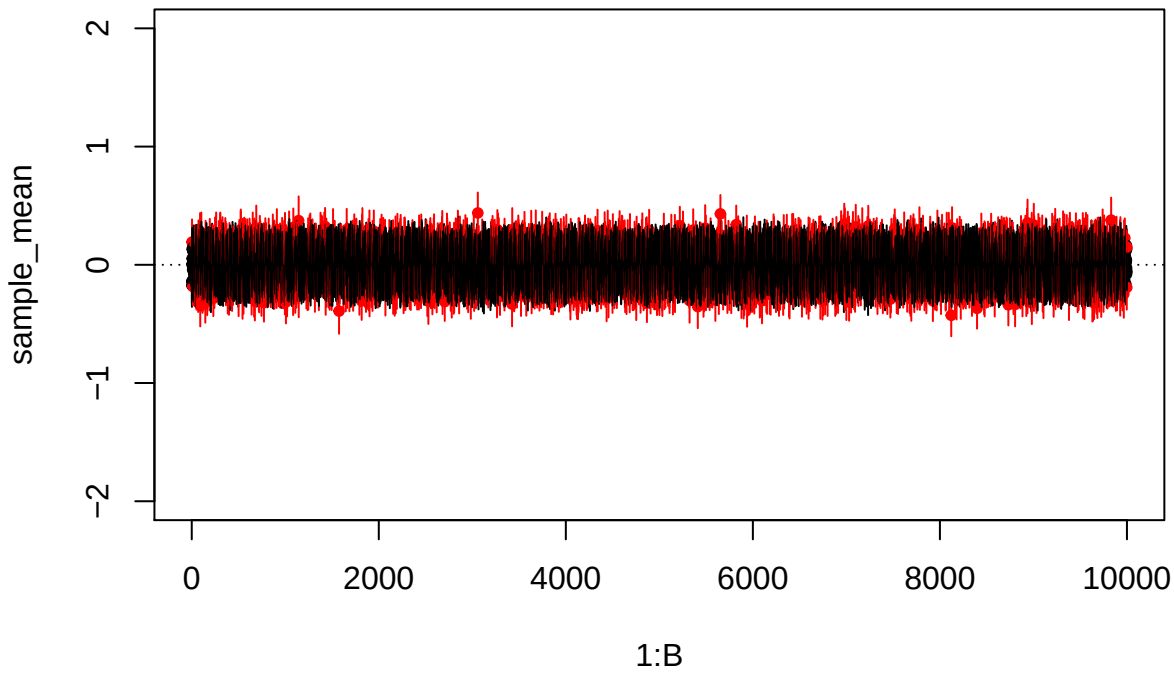

```
## [1] "fraction of true mean = 0.8992"
```



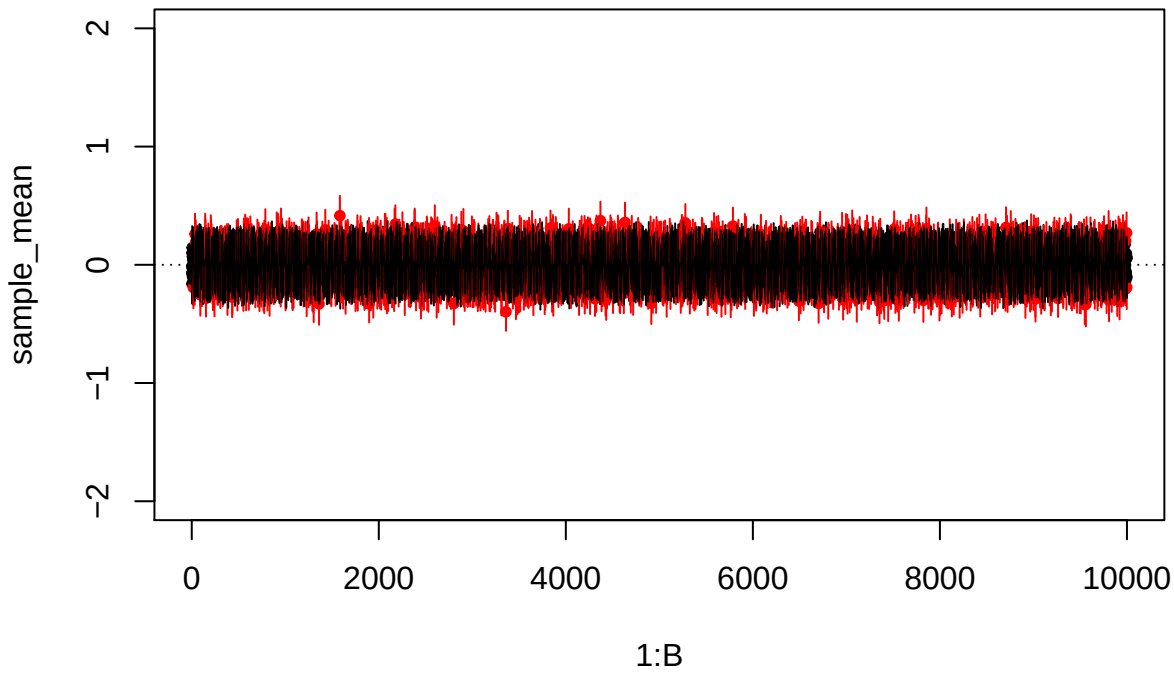

```
## [1] "fraction of true mean = 0.8993"
```




```
## [1] "fraction of true mean = 0.903"
```




```
## [1] "fraction of true mean = 0.9009"
```



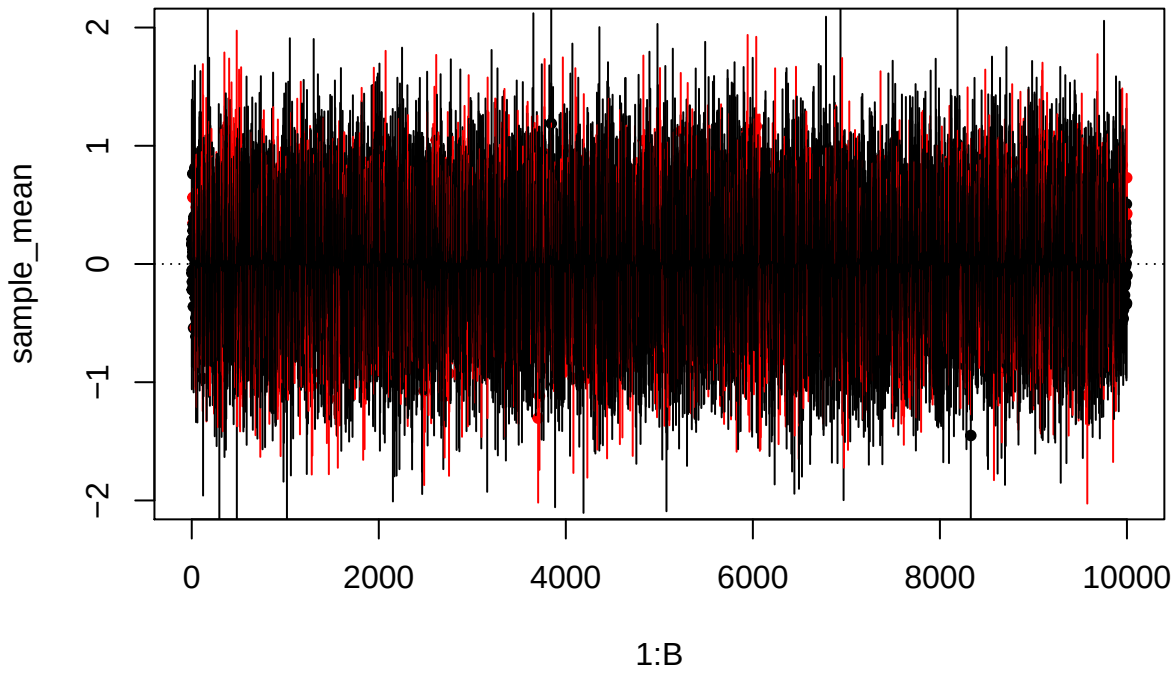

```
library(rmutil)
```

```
##
## Attaching package: 'rmutil'

## The following object is masked from 'package:stats':
##
##     nobs

## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

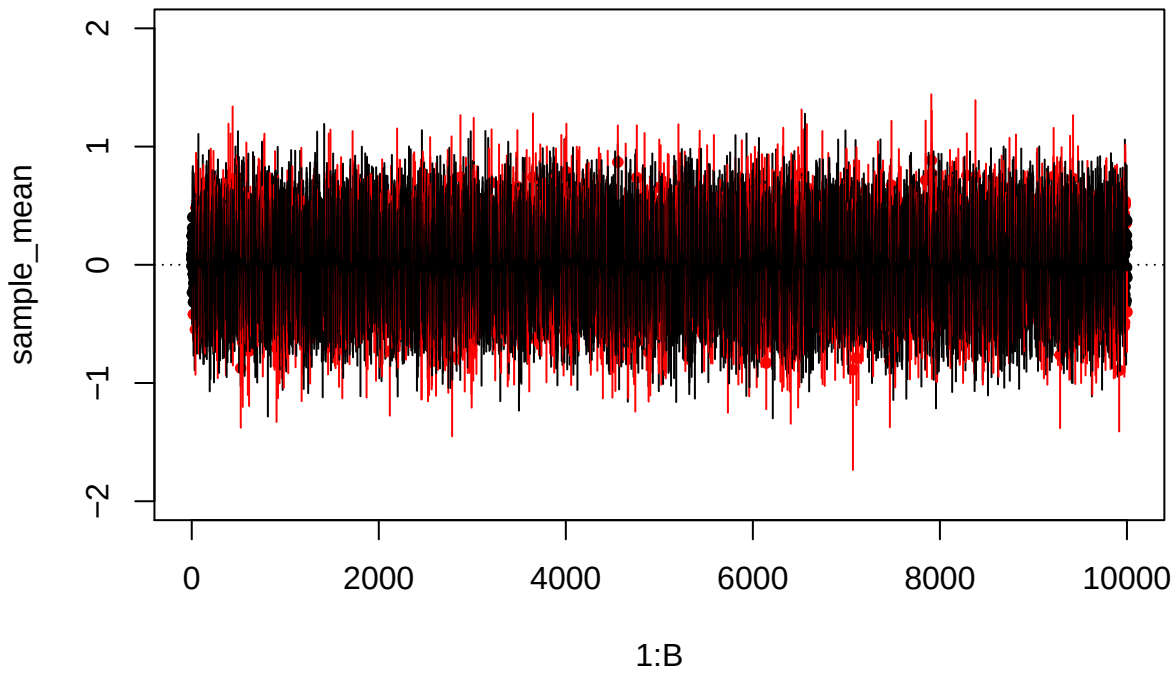
```
n = 10
for(i in 1:10){
  alpha = 0.10
  B     = 10000
  sample_mean   = numeric(B)
  conf_upper = numeric(B)
  conf_lower = numeric(B)
  num_success <- 0
  for (j in 1:B) {
    x <- rlaplace(n, m = 0, s = 1/sqrt(2))
    out = t.test(x, conf=1-alpha)
```

```
    conf_lower[j] = out$conf.int[1]
    conf_upper[j] = out$conf.int[2]
    sample_mean[j]   = out$estimate
     if((conf_lower[j] < 0) & (0 < conf_upper[j])){
    num_success <- num_success + 1
   }
  }
  frac_contain_true_mean <- num_success/B
  print(paste("fraction of true mean =",frac_contain_true_mean))
  contains_mean = (conf_lower < 0) & (0 < conf_upper) # T/F
  color_vec     = ifelse(contains_mean == TRUE, 1, 2) #if true,1;if false,2
  plot(1:B, sample_mean, pch=20, ylim = c(-2, 2), col=color_vec)
  segments(1:B, conf_lower, 1:B, conf_upper, col=color_vec)
  abline(h=0, lty="dotted") #or solid
  n = n + 10
}
```

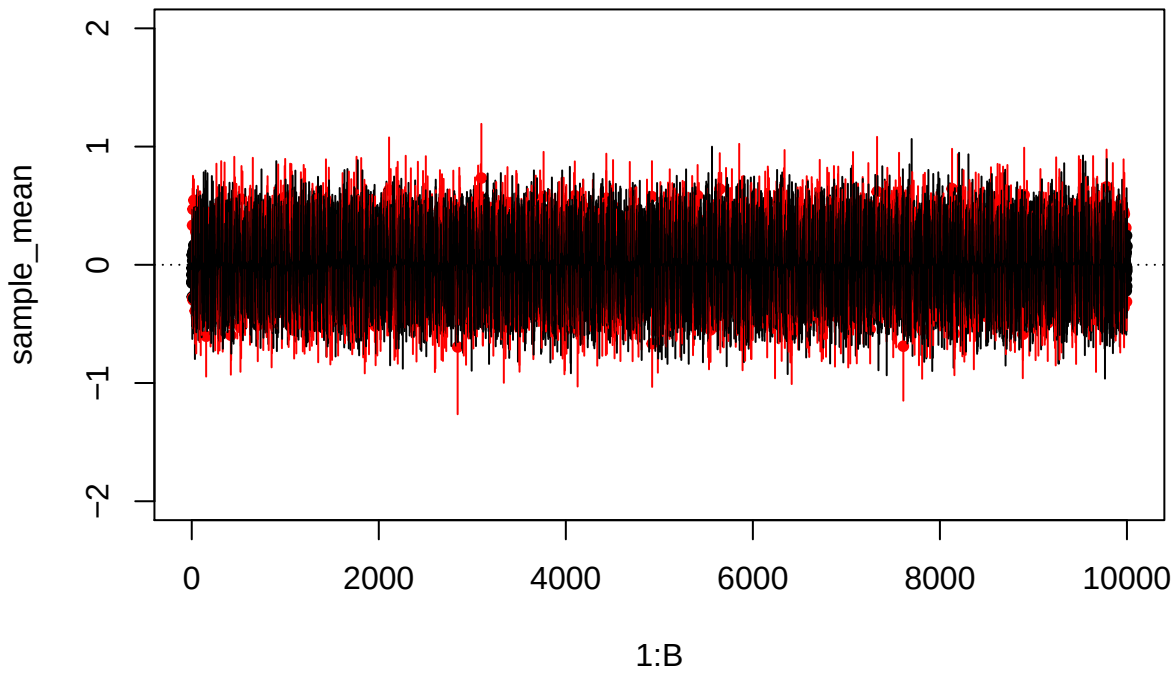
```
## [1] "fraction of true mean = 0.9093"
```



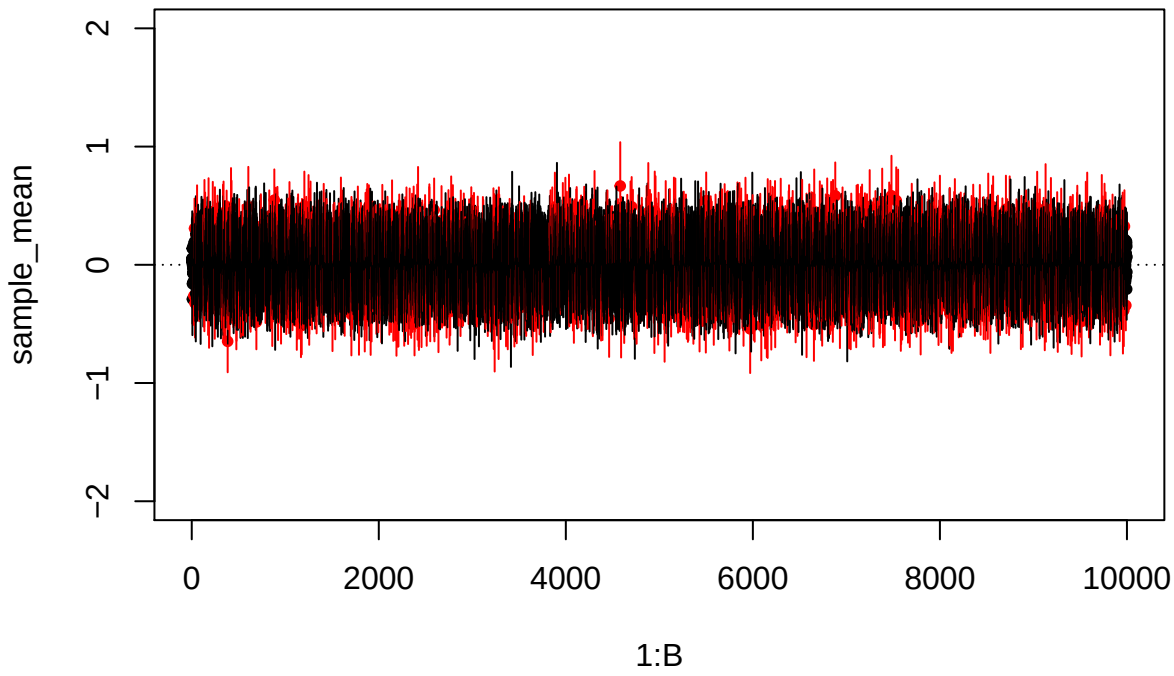

```
## [1] "fraction of true mean = 0.8984"
```



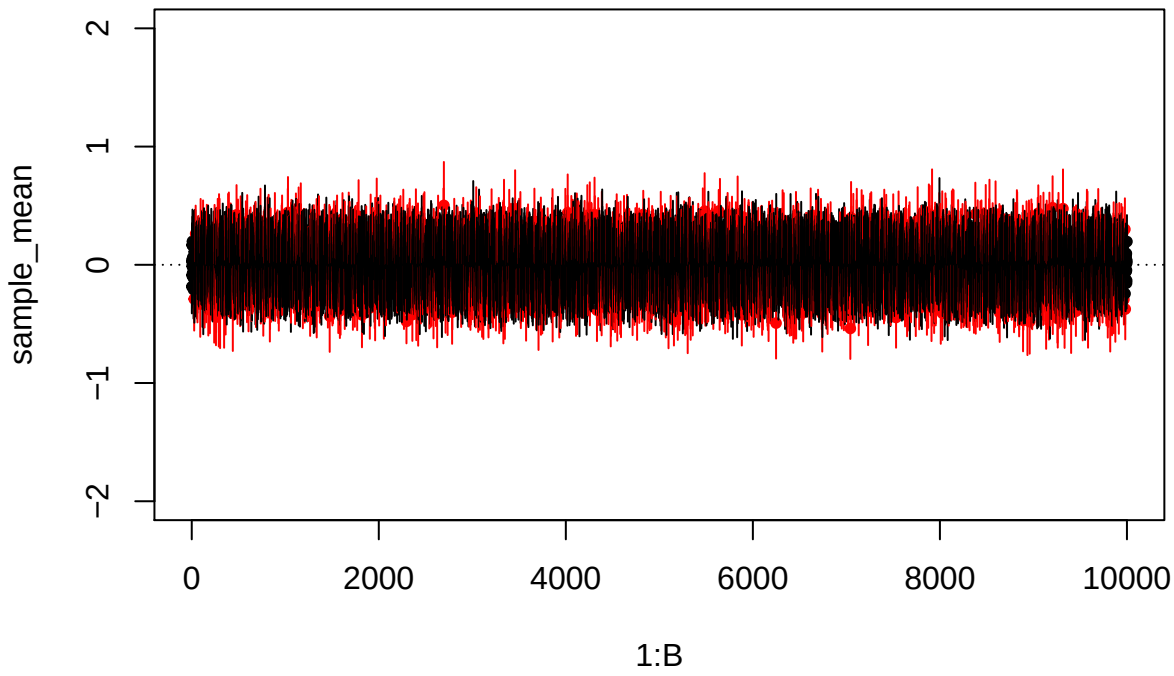

```
## [1] "fraction of true mean = 0.8997"
```



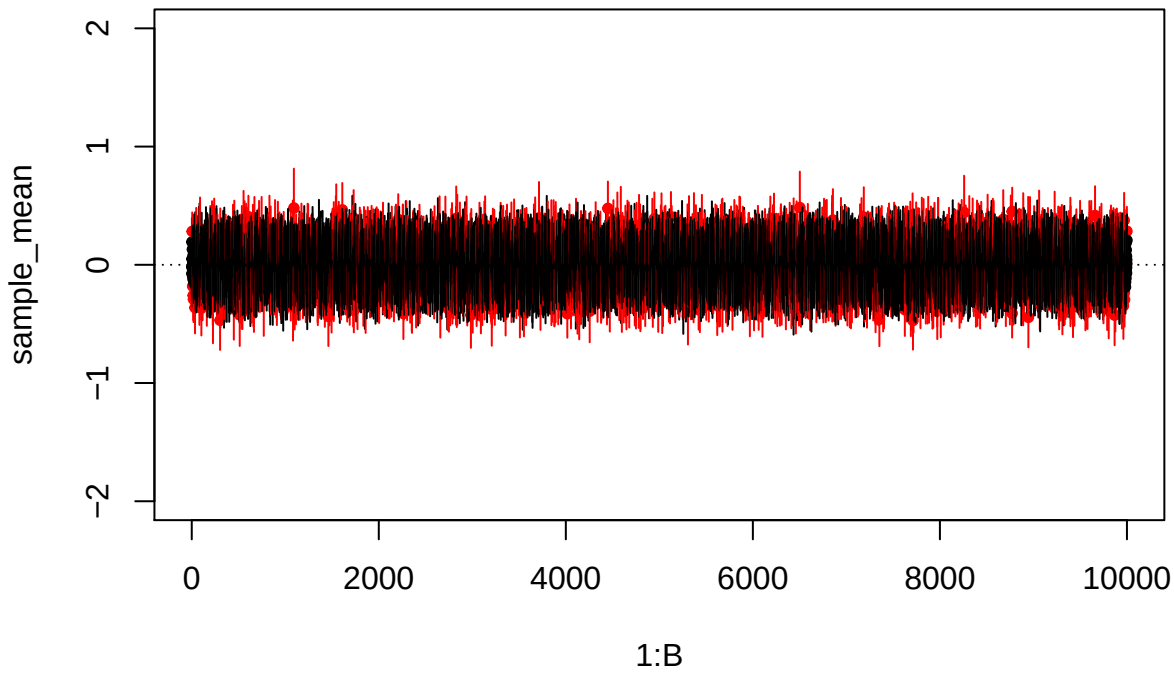

```
## [1] "fraction of true mean = 0.903"
```



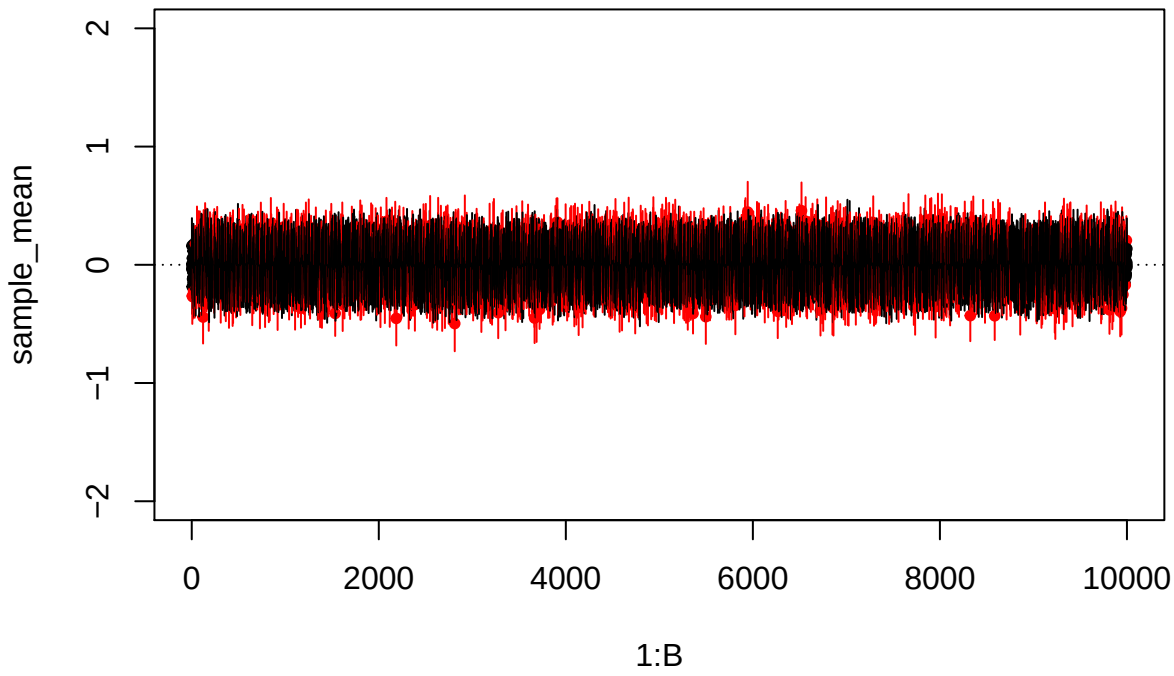

```
## [1] "fraction of true mean = 0.8996"
```



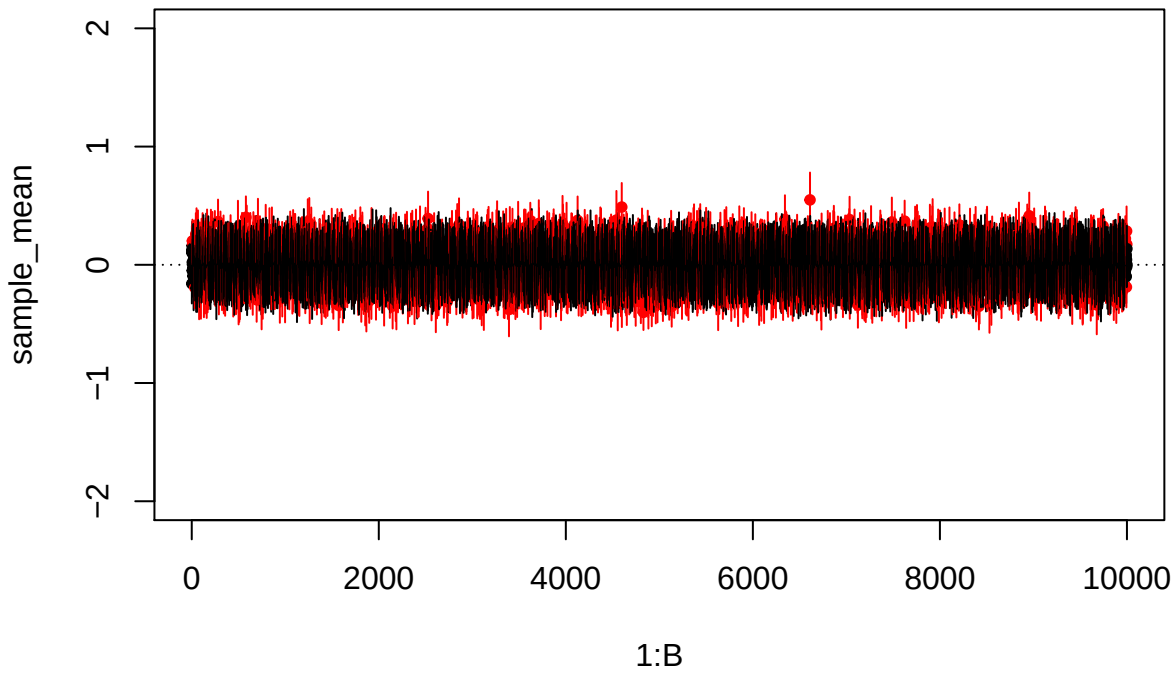

```
## [1] "fraction of true mean = 0.9004"
```



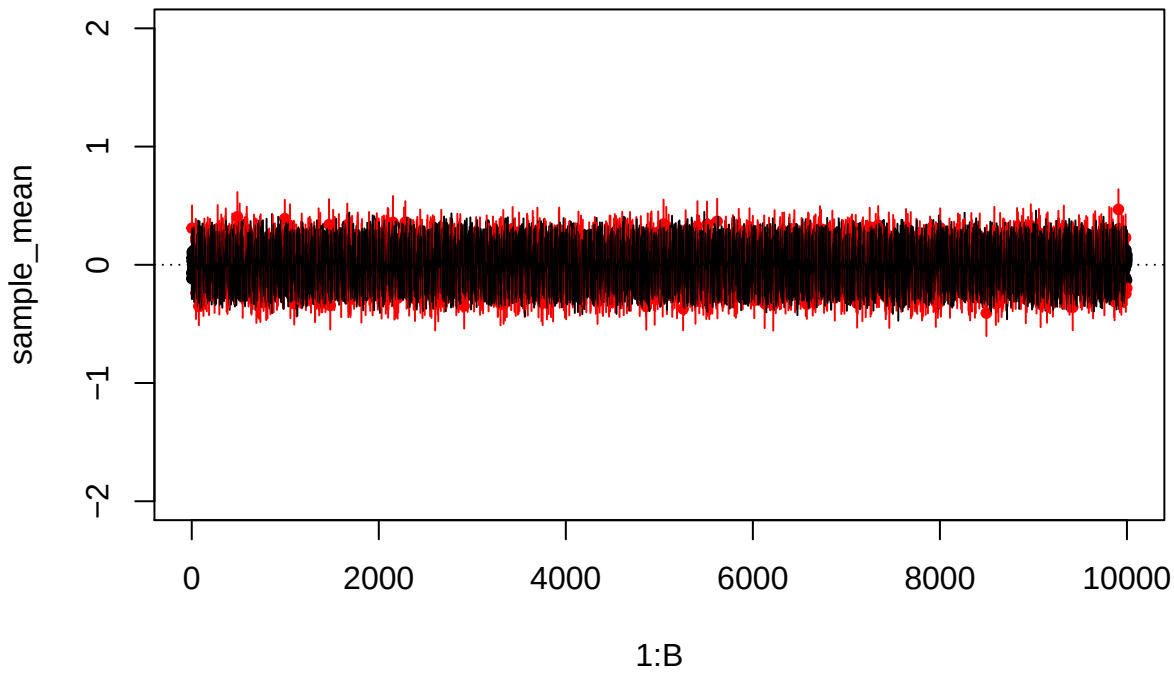

```
## [1] "fraction of true mean = 0.9005"
```



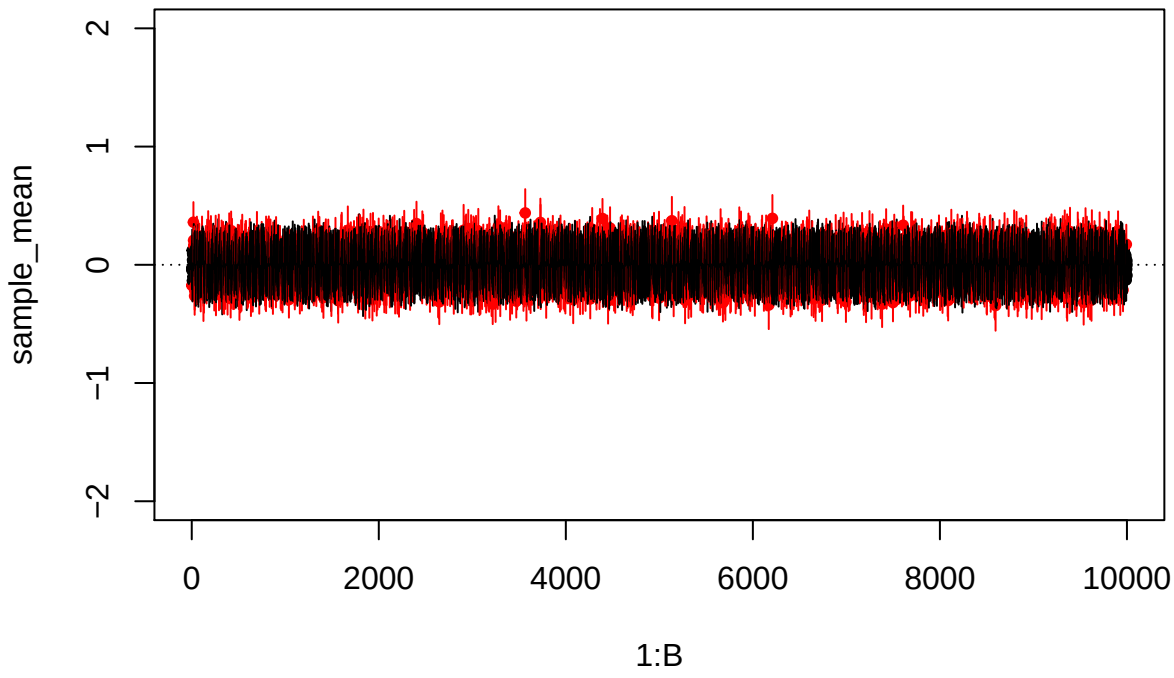

```
## [1] "fraction of true mean = 0.8995"
```




```
## [1] "fraction of true mean = 0.9015"
```



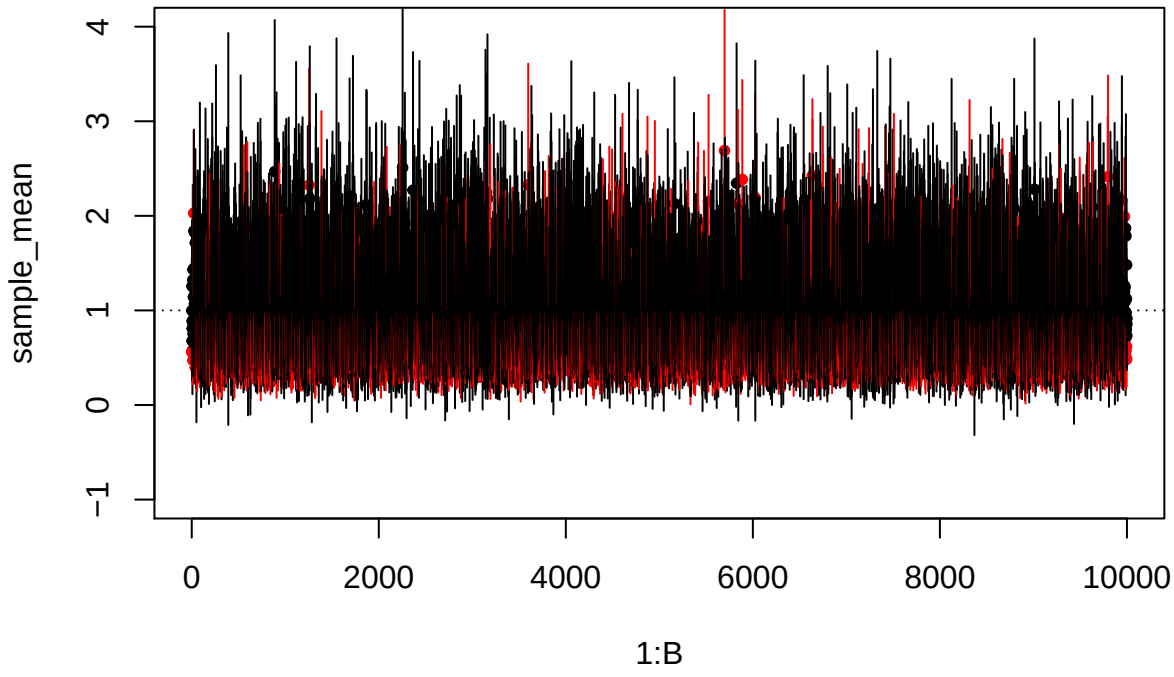

```
## [1] "fraction of true mean = 0.9"
```



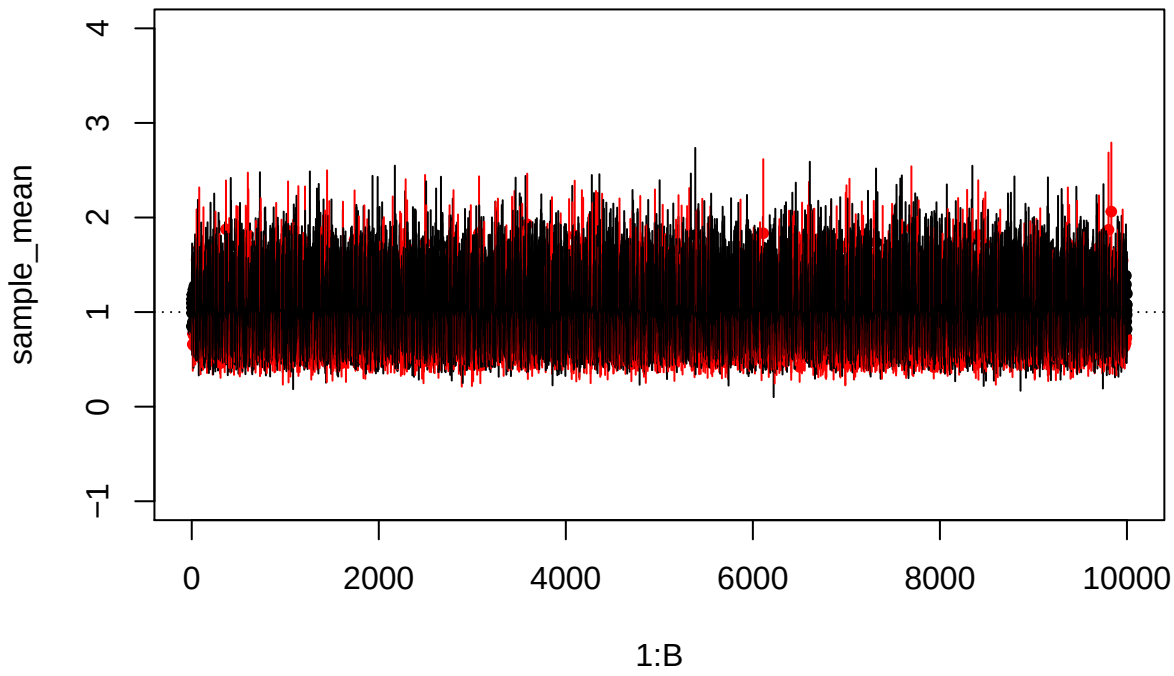

```
n = 10
for(i in 1:10){
  alpha = 0.10
  B     = 10000
  sample_mean   = numeric(B)
  conf_upper = numeric(B)
  conf_lower = numeric(B)
  num_success <- 0
  for (j in 1:B) {
    x <- rexp(n, rate = 1)
    out = t.test(x, conf=1-alpha)
    conf_lower[j] = out$conf.int[1]
    conf_upper[j] = out$conf.int[2]
    sample_mean[j]   = out$estimate
    if((conf_lower[j] < 0) & (0 < conf_upper[j])){
    num_success <- num_success + 1
   }
  }
  frac_contain_true_mean <- num_success/B
  print(paste("fraction of true mean =",frac_contain_true_mean))
  contains_mean = (conf_lower < 1) & (1 < conf_upper) # T/F
  color_vec     = ifelse(contains_mean == TRUE, 1, 2) #if true,1;if false,2

  plot(1:B, sample_mean, pch=20, ylim = c(-1, 4), col=color_vec) #sample mean
  segments(1:B, conf_lower, 1:B, conf_upper, col=color_vec)
  abline(h=1, lty="dotted") #or solid
```

```
  n = n + 10
}
```

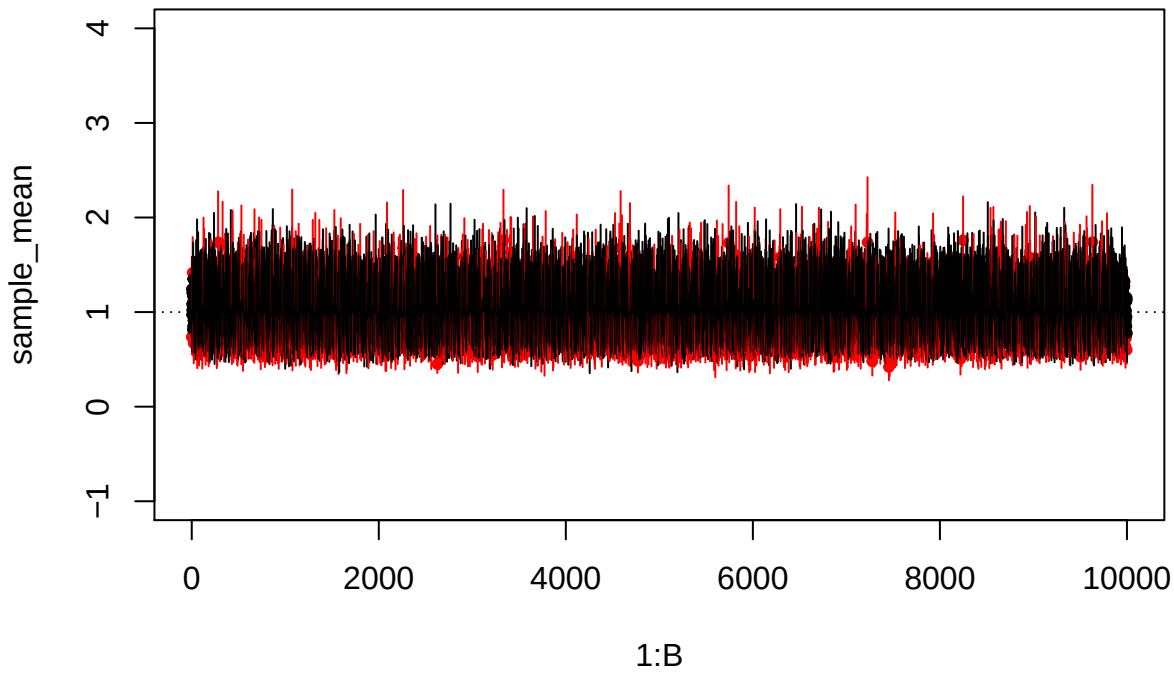
```
## [1] "fraction of true mean = 0.0043"
```



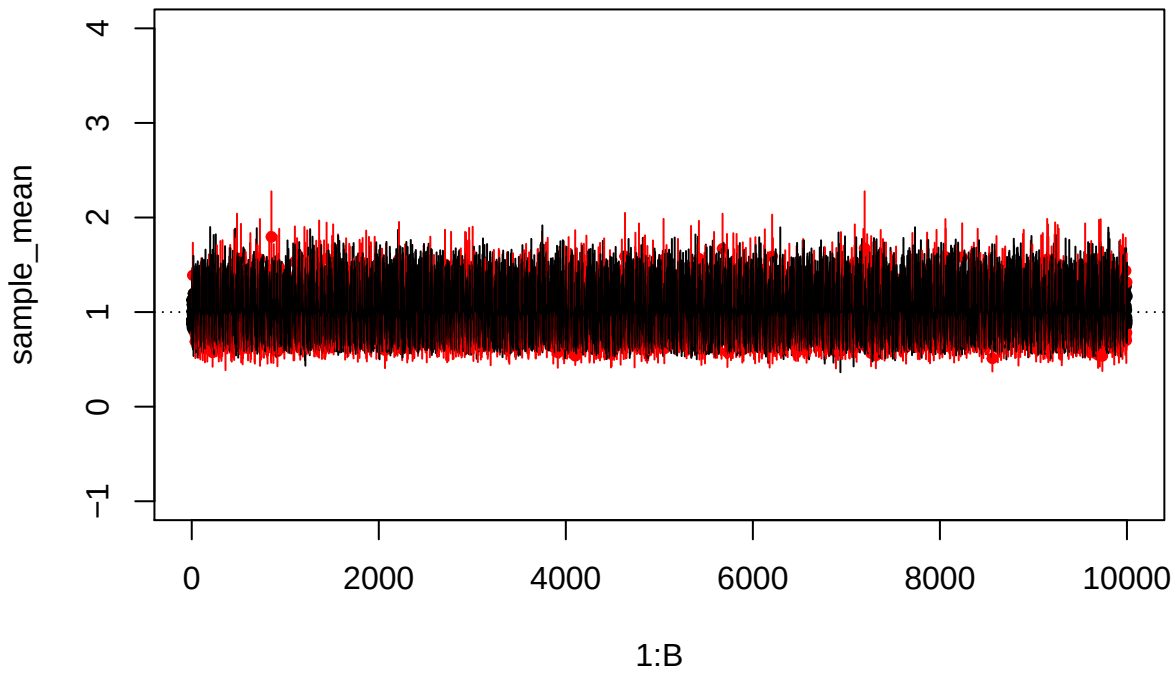

```
## [1] "fraction of true mean = 0"
```



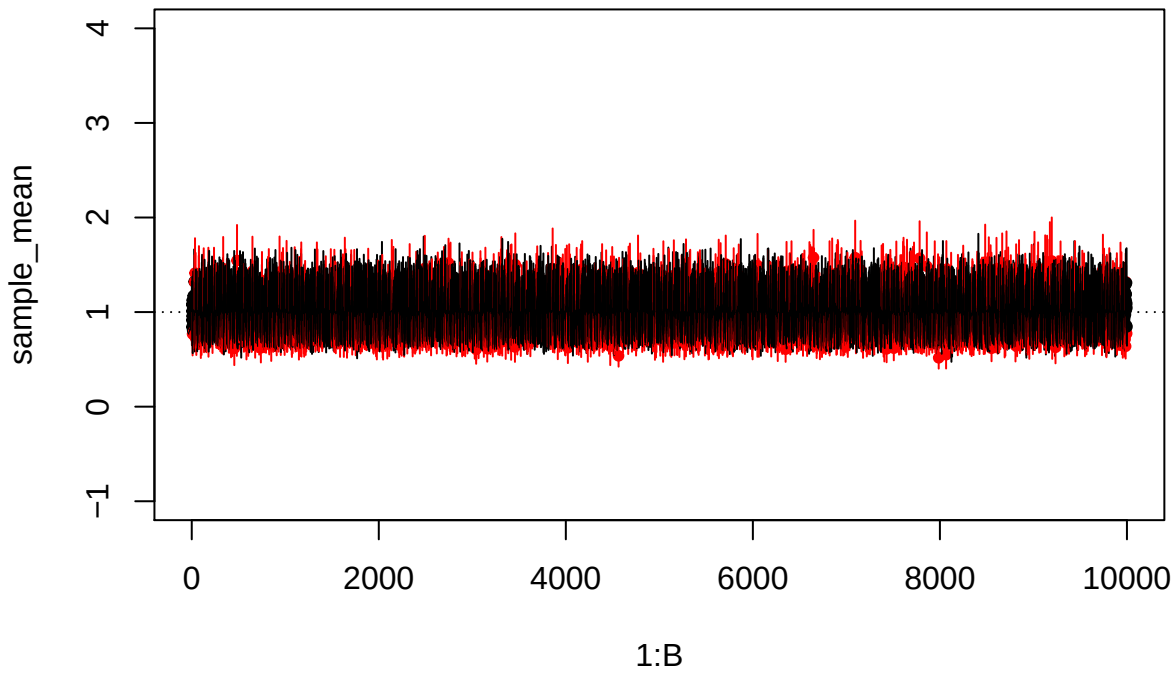

```
## [1] "fraction of true mean = 0"
```



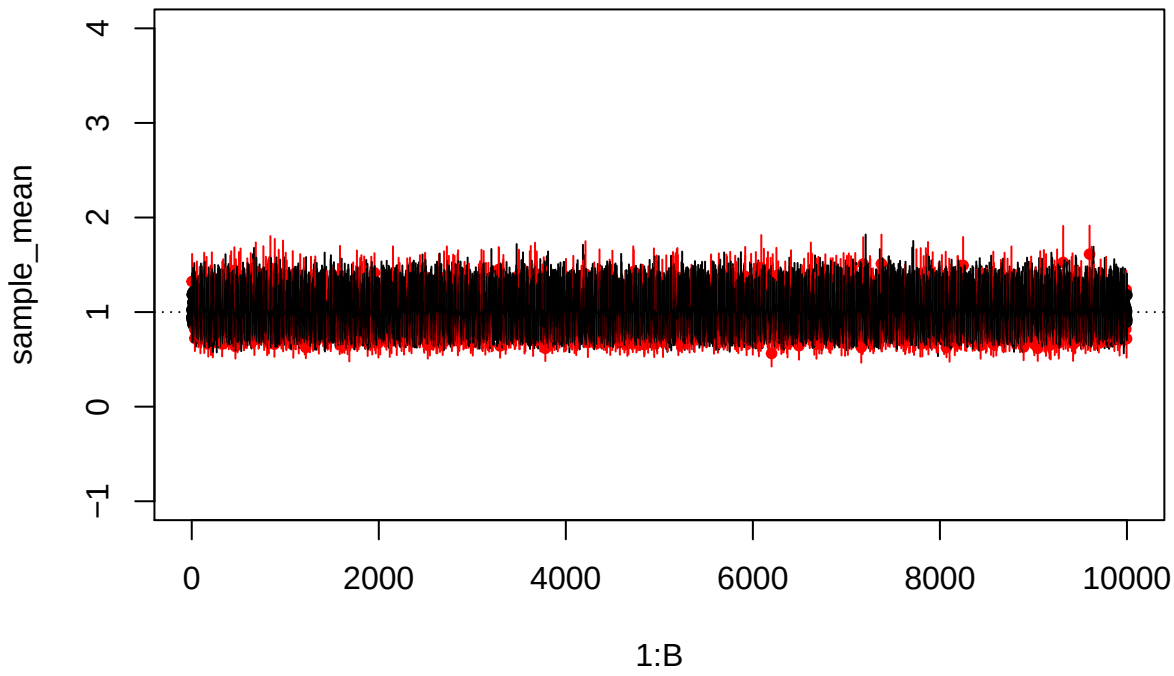

```
## [1] "fraction of true mean = 0"
```



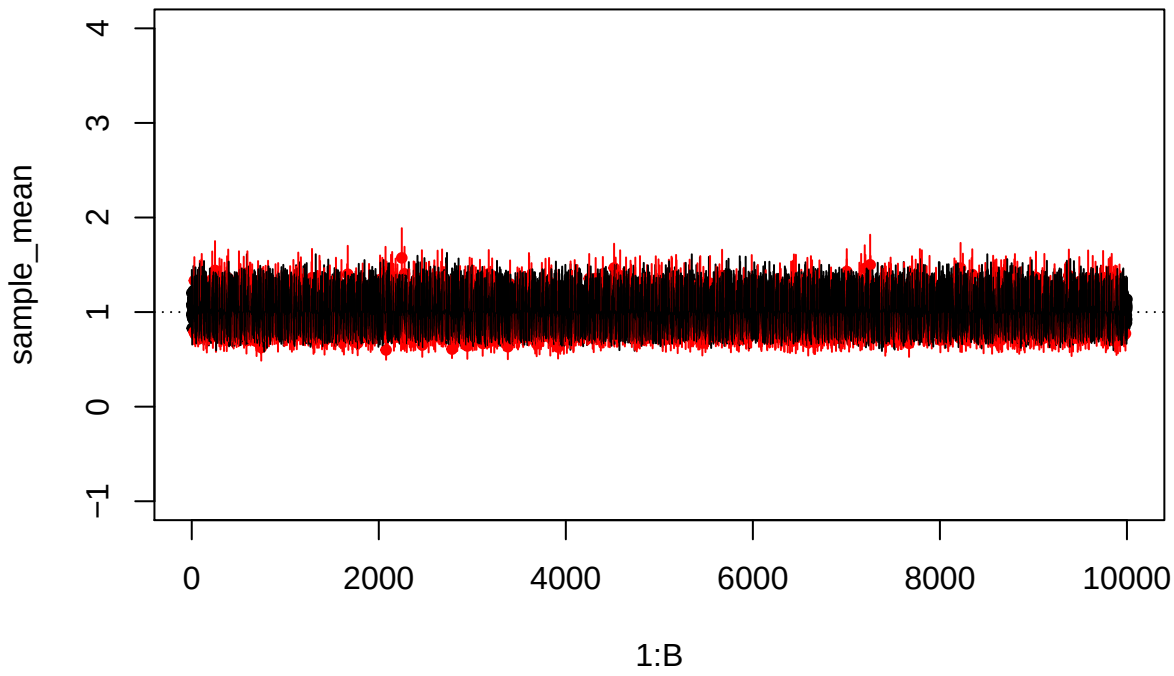

```
## [1] "fraction of true mean = 0"
```



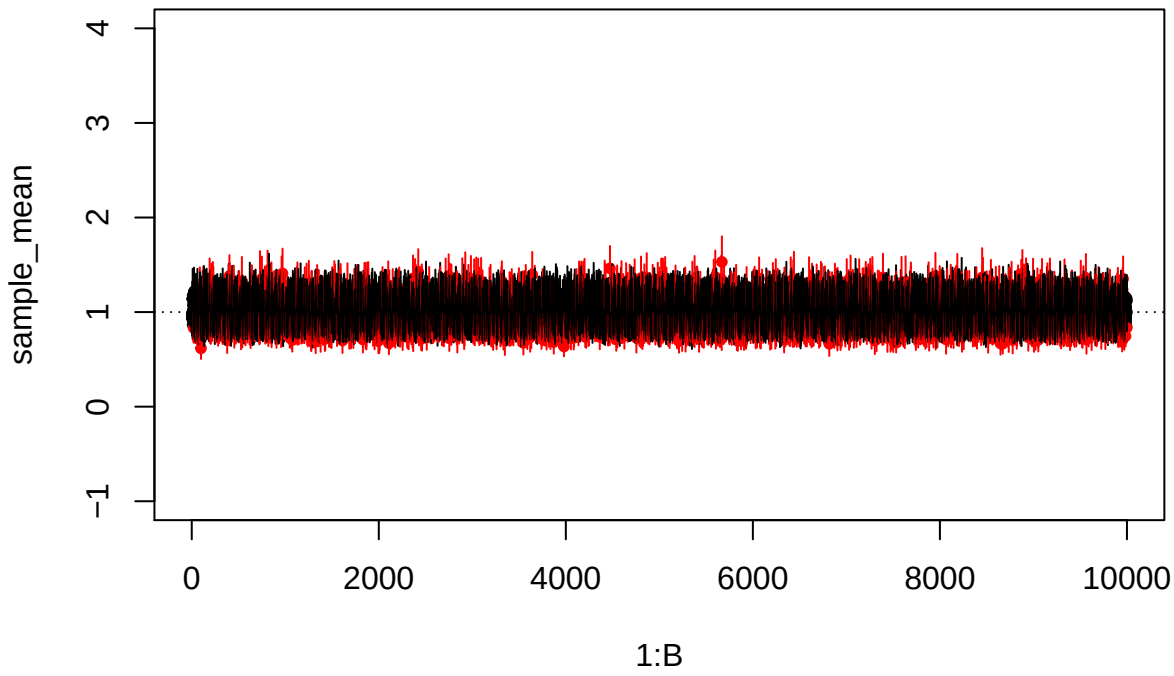

```
## [1] "fraction of true mean = 0"
```



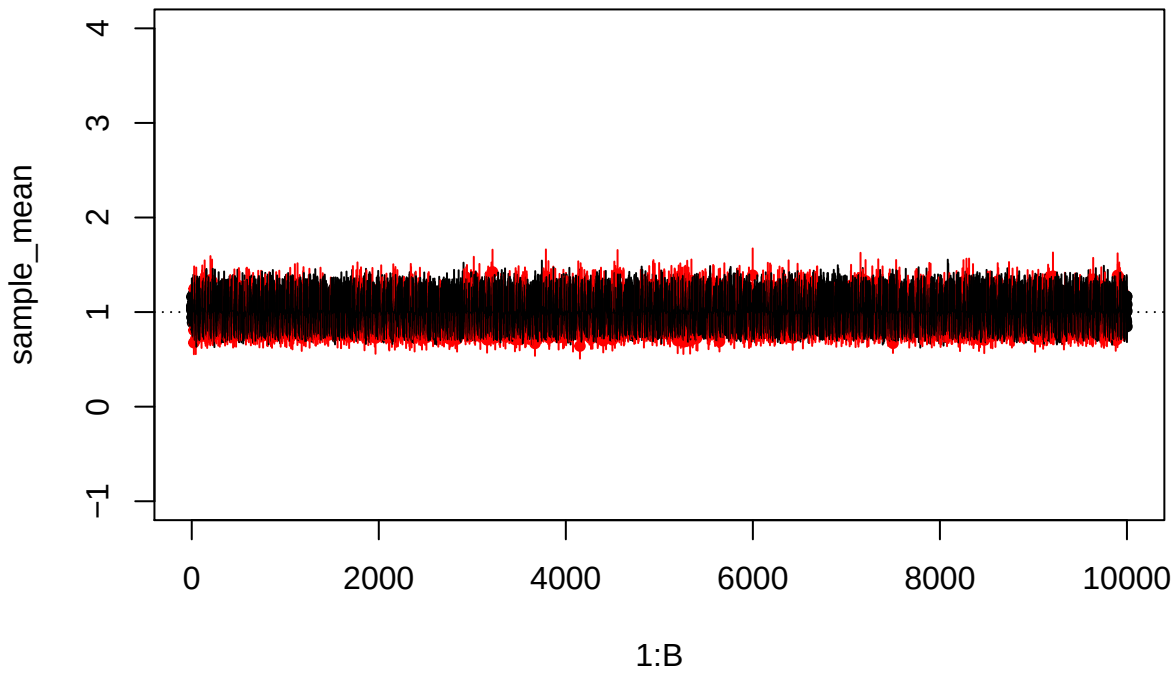

```
## [1] "fraction of true mean = 0"
```



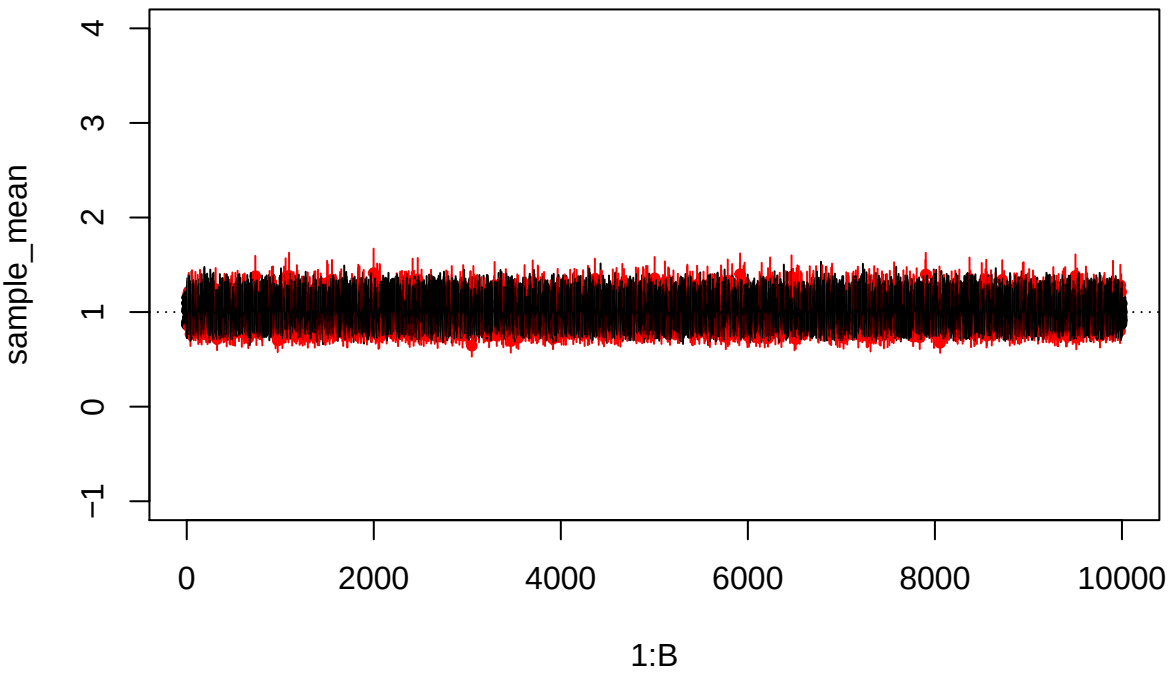

```
## [1] "fraction of true mean = 0"
```




```
## [1] "fraction of true mean = 0"
```




```
## [1] "fraction of true mean = 0"
```


sample_mean
4
3
2
1
0
−1
0
2000
4000
6000
8000
10000
1:B