

HW2 Markdown

Patrick Walker

October 19, 2018

```
compare_outcomes <- function(iterations, burnIn){

  if(burnIn > iterations){return("burnIN can't be longer than iterations")}
    #the above line was added to prevent errors of non-finite Length vectors
  source(file = "C:\\\\Users\\\\Patrick's Computer\\\\Documents\\\\GitHub\\\\assignment-2-cardinalbraxiate
1\\\\HW2.R") #Opens file of Metropolis function

  a_vals <- matrix(nrow = 10, ncol = 2) #a blank matrix in which to put the mean and sd of a

  for(i in 1:10){ #for Loop for each of the iterations
    a_int <- runif(1) #random number inserted for a
    b_int <- runif(1) #random number inserted for b
    sd_int <- runif(1) #random number inserted for sd

    startvalue <- c(a_int,b_int,sd_int) #feeds the above random numbers into startvalue

    chain_final = (Metropolis(iterations, burnIn )) #defines chain as the output of the Metropolis
function

    #chain_final <- chain_final[-burnIn,] #I included this because I assume you don't want the burn
In

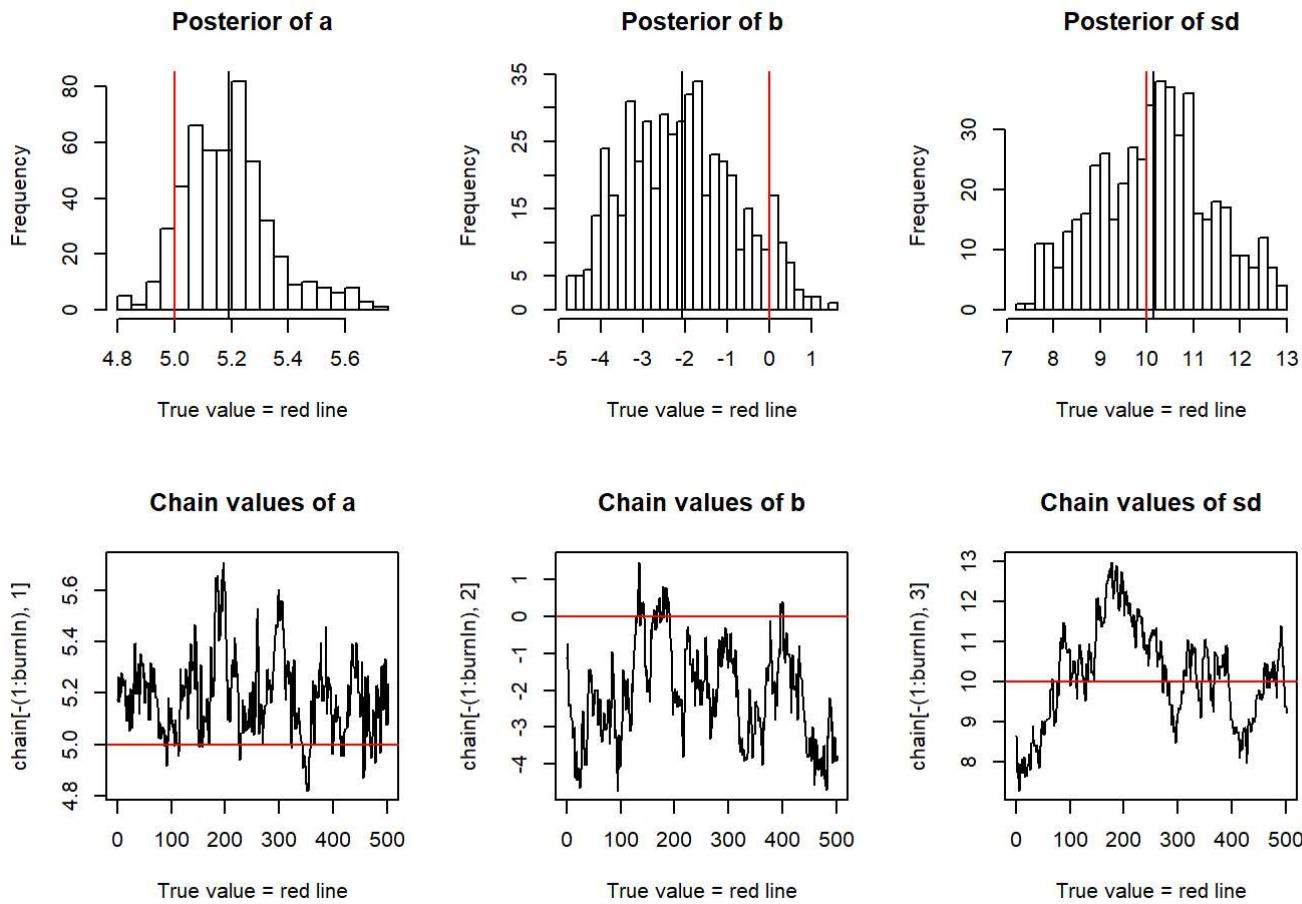
    mean_a <- mean(chain_final[,1]) #takes the mean of the first column of the chain
    std <- sd(chain_final[,1]) #takes the standard deviation of the first column of chain
    a_vals[i,1] <- mean_a #stores mean_a in ith row 1st column of a_vals
    a_vals[i,2] <- std #stores std in ith row 2nd column of a_vals
    print(a_vals[i,]) #prints ith row of a_vals

  }
  return(a_vals)
}

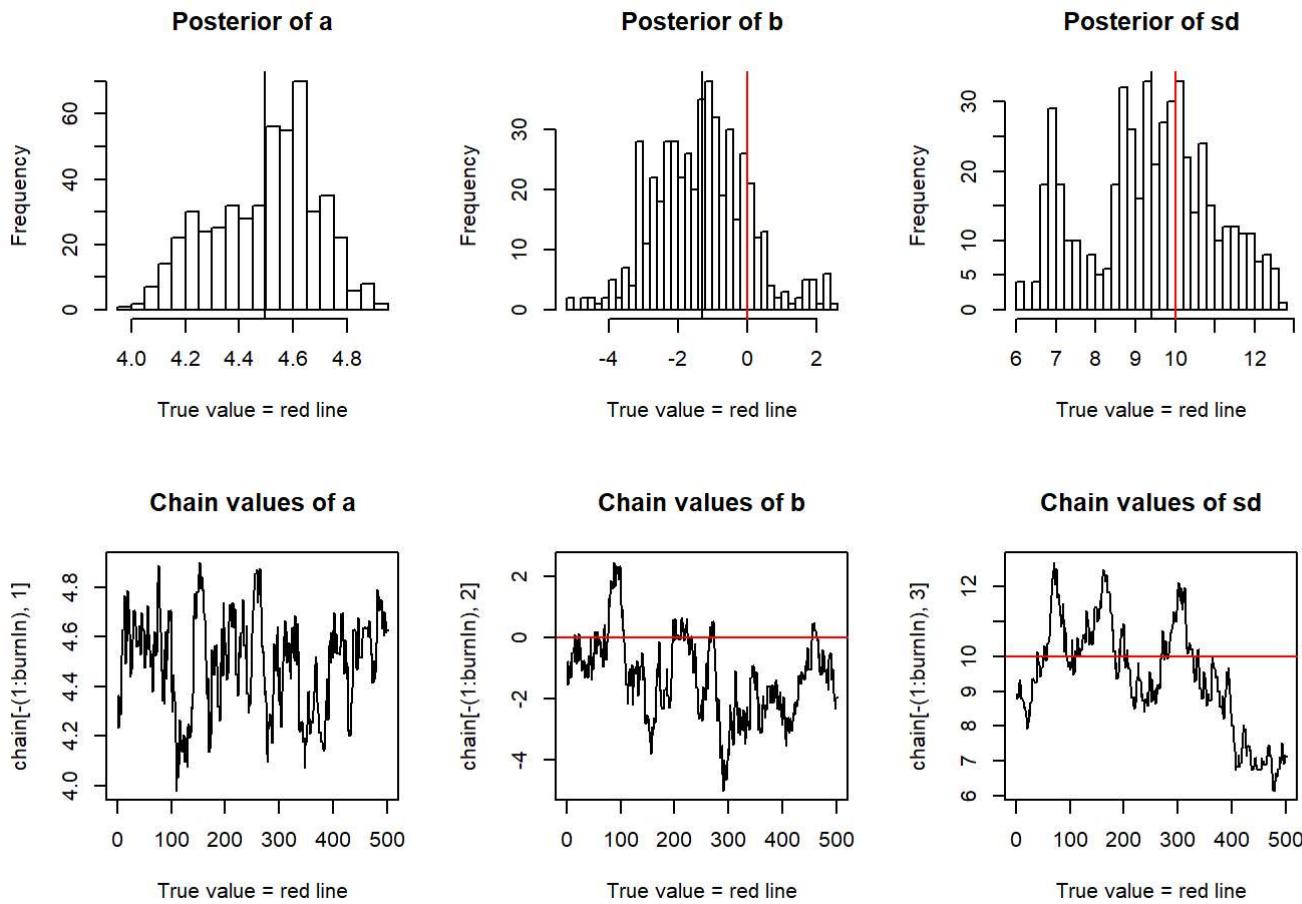
print("1000 iterations, 500 burnIn")
```

```
## [1] "1000 iterations, 500 burnIn"
```

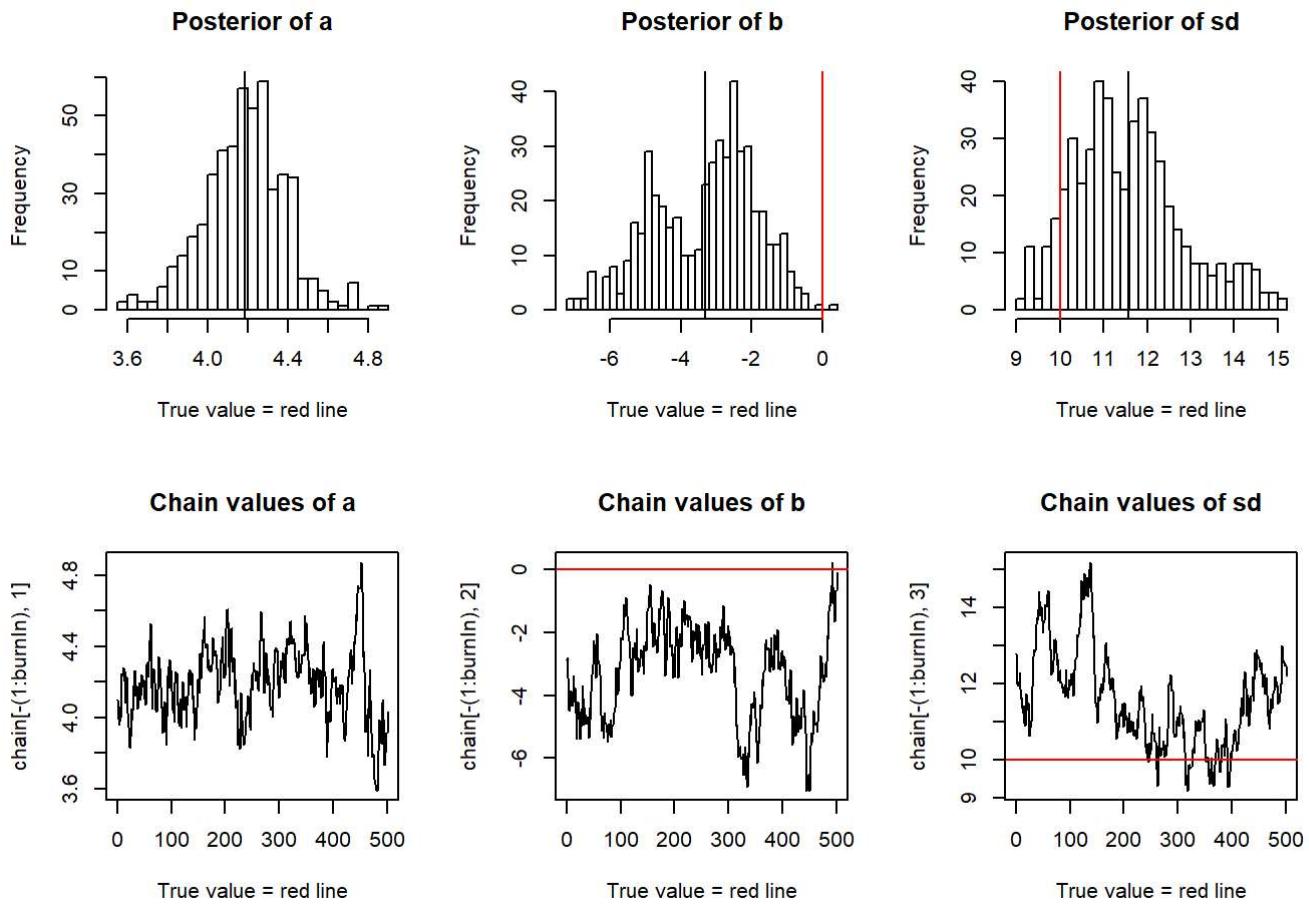
```
a_vals2 <- compare_outcomes(1000, 500)
```



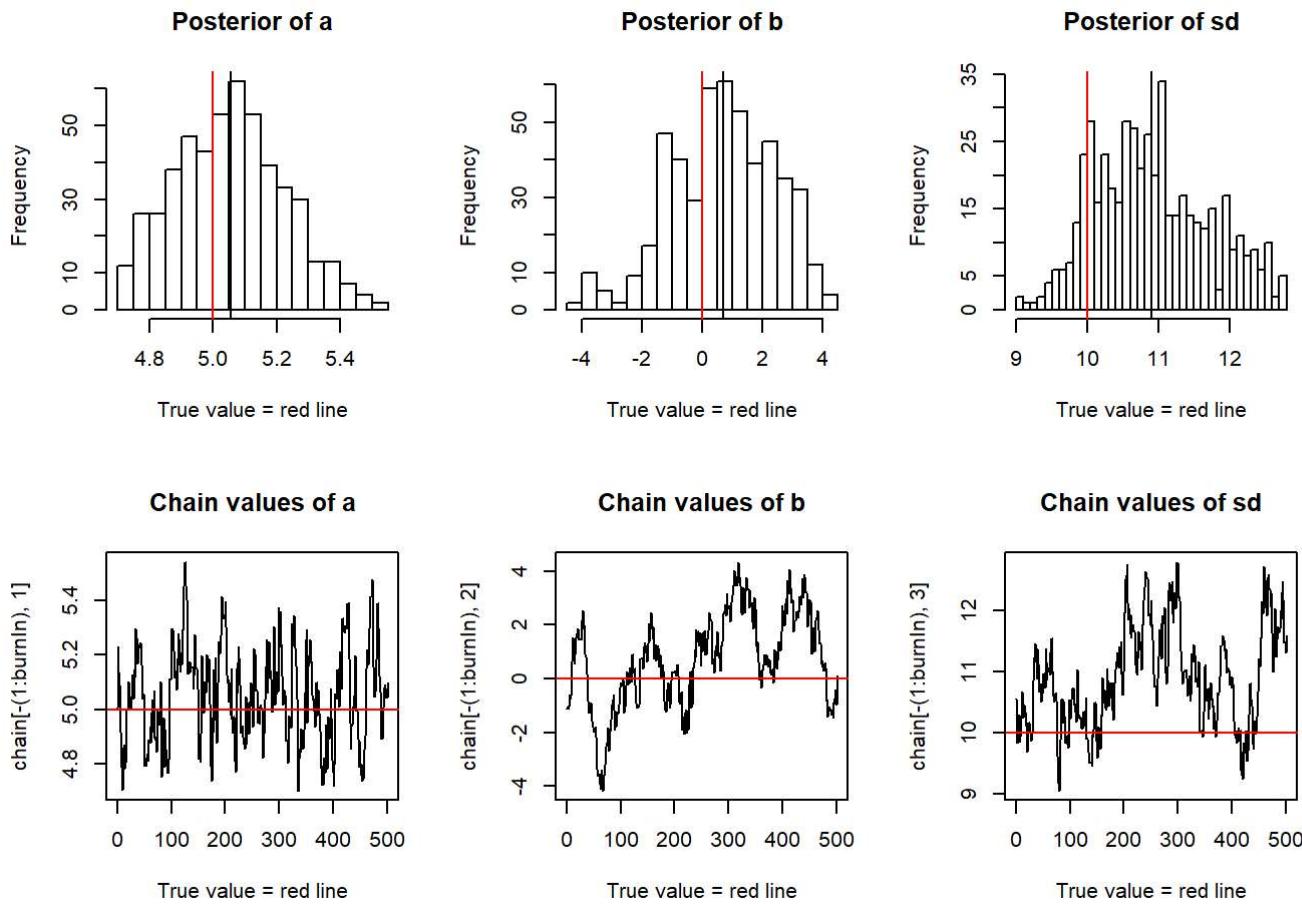
```
## [1] 5.198823 0.228925
```



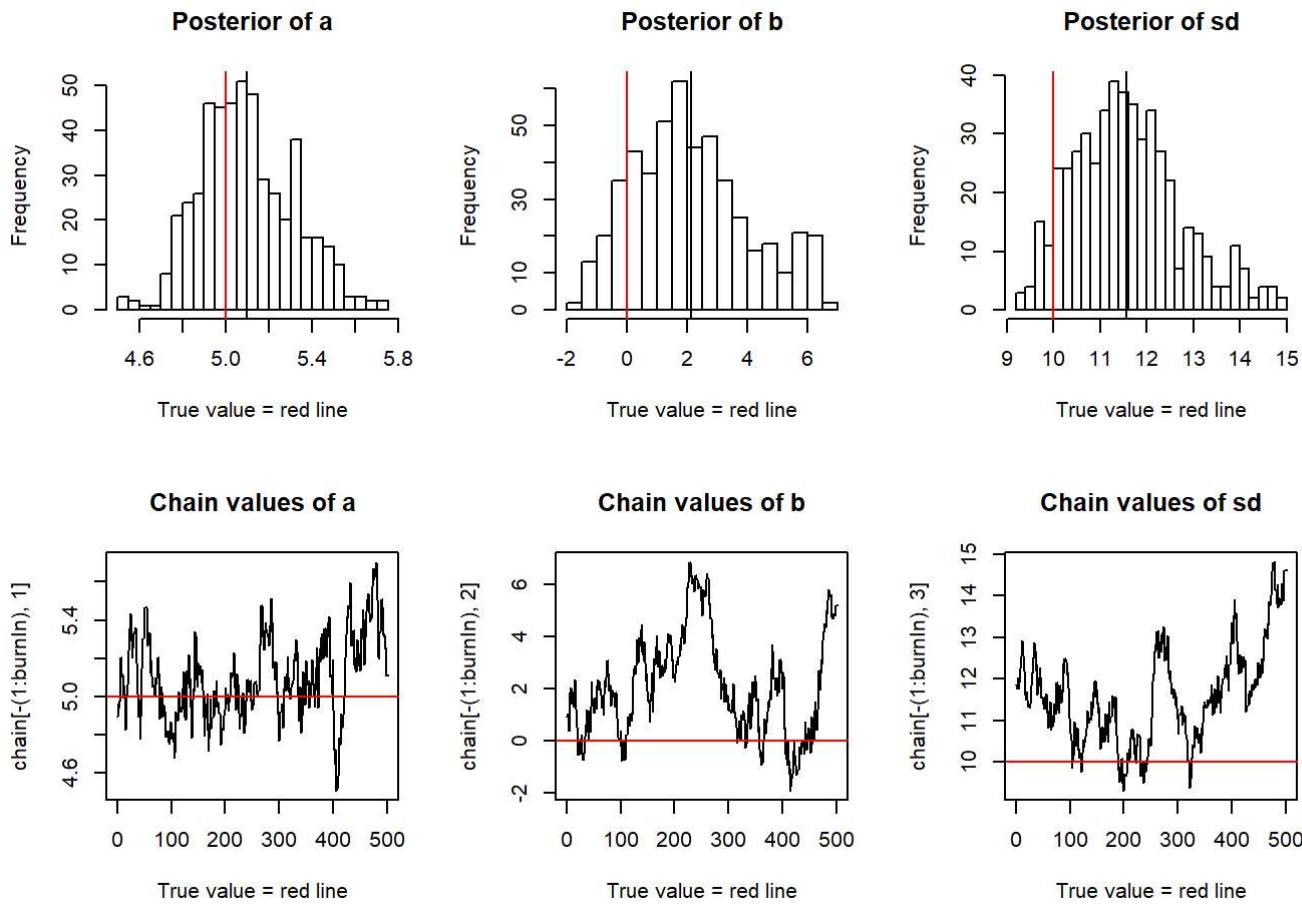
```
## [1] 4.5262809 0.2047069
```



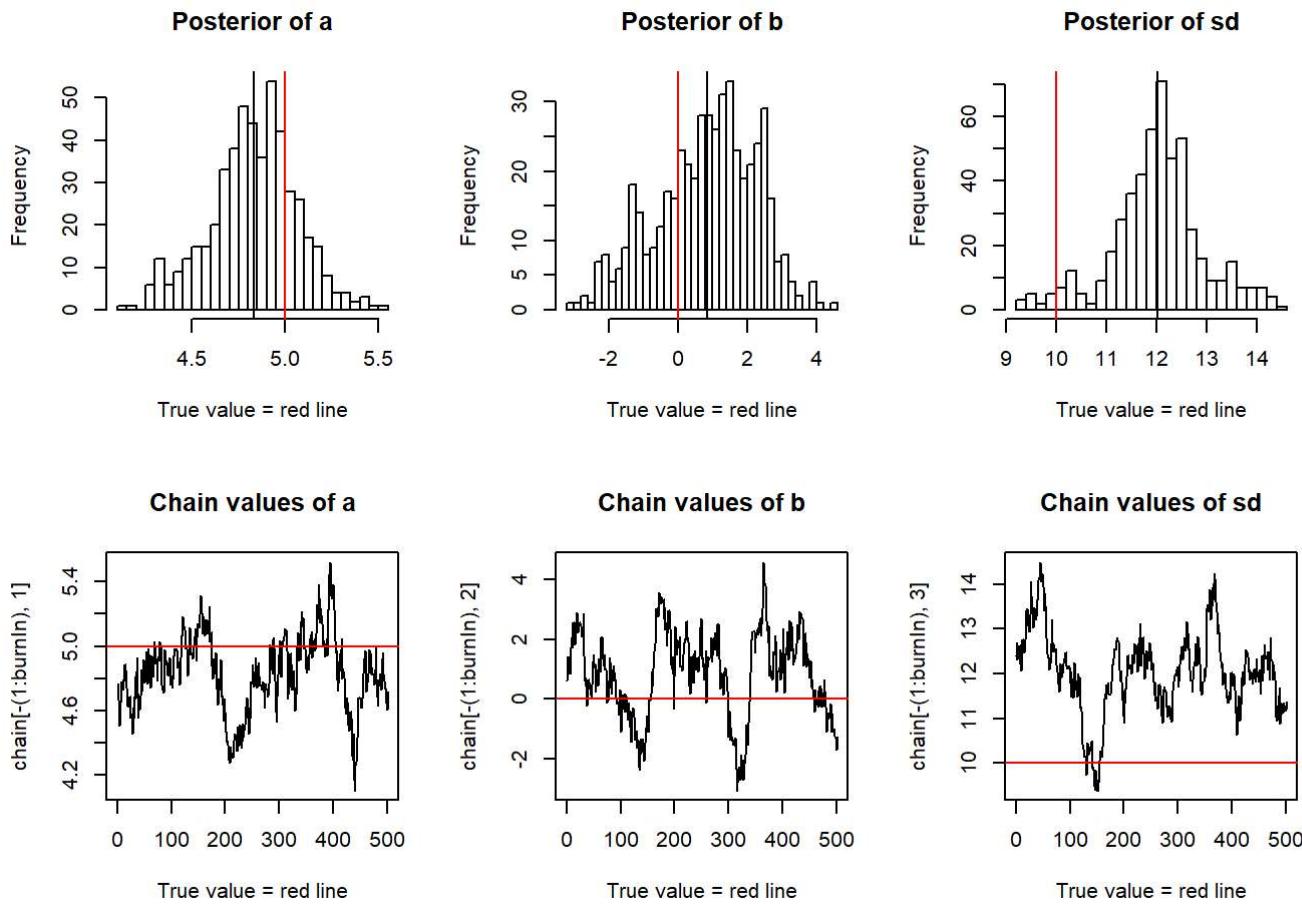
```
## [1] 4.1901209 0.1852082
```



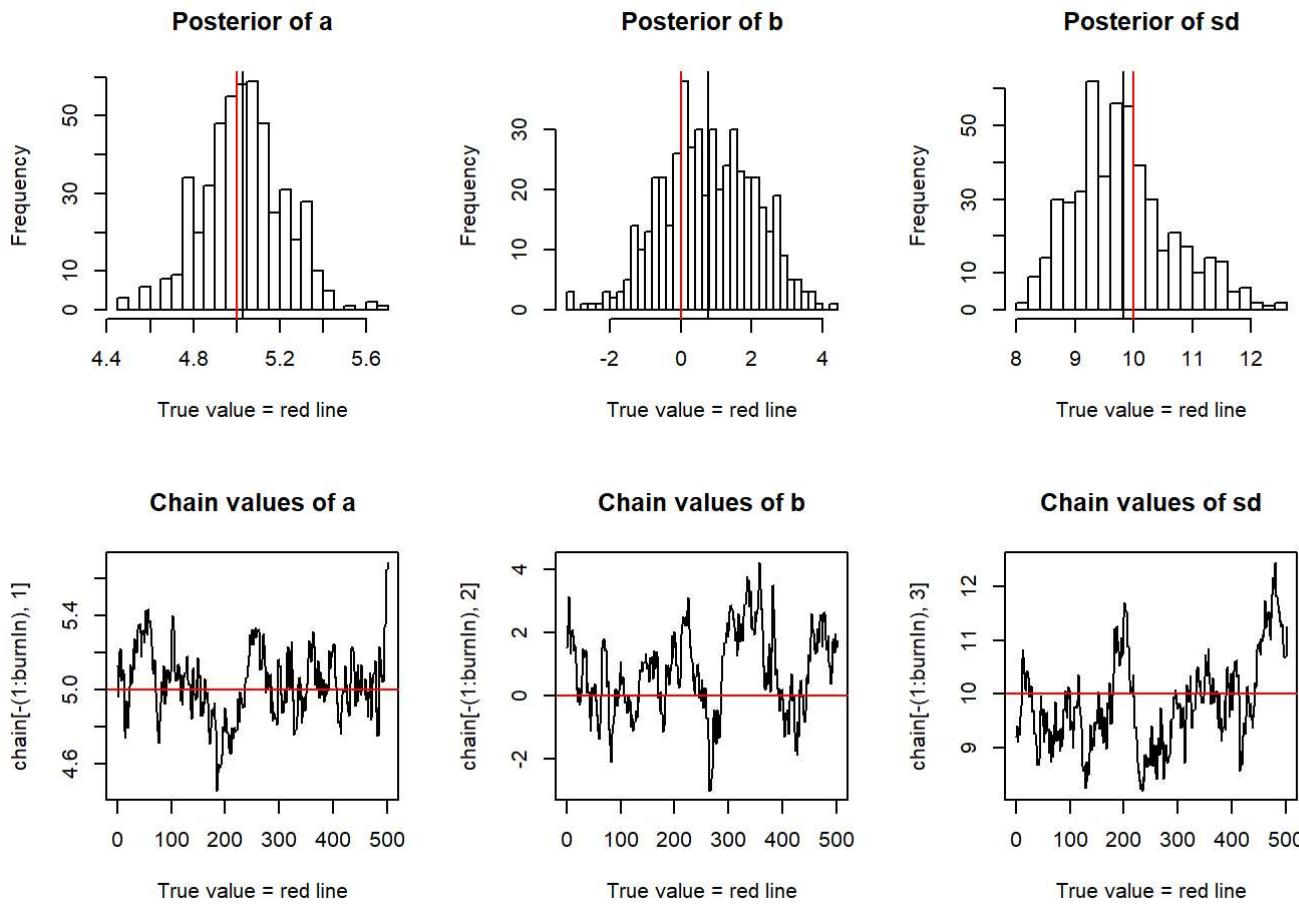
```
## [1] 5.0101783 0.2519691
```



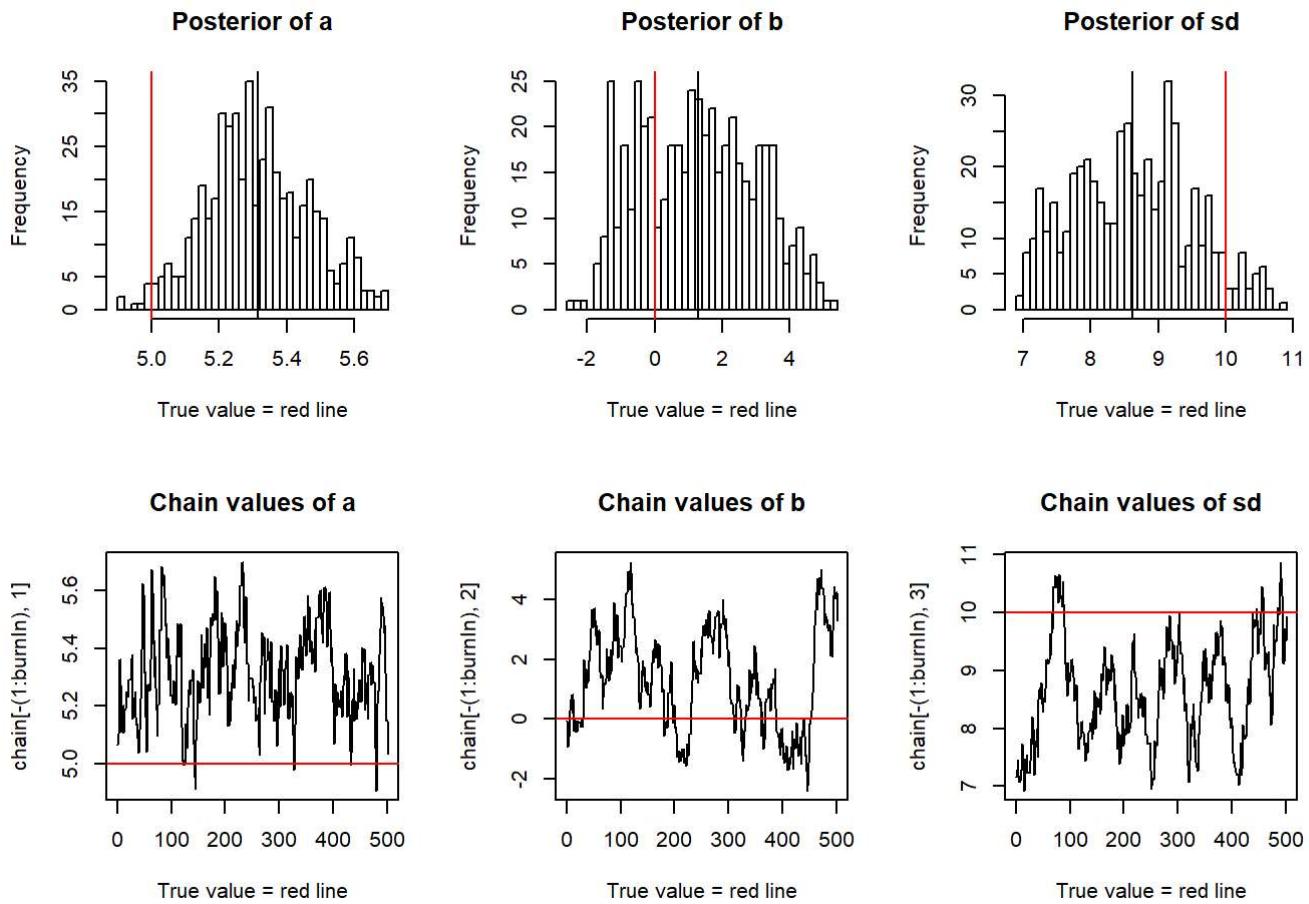
```
## [1] 5.0381471 0.2408744
```



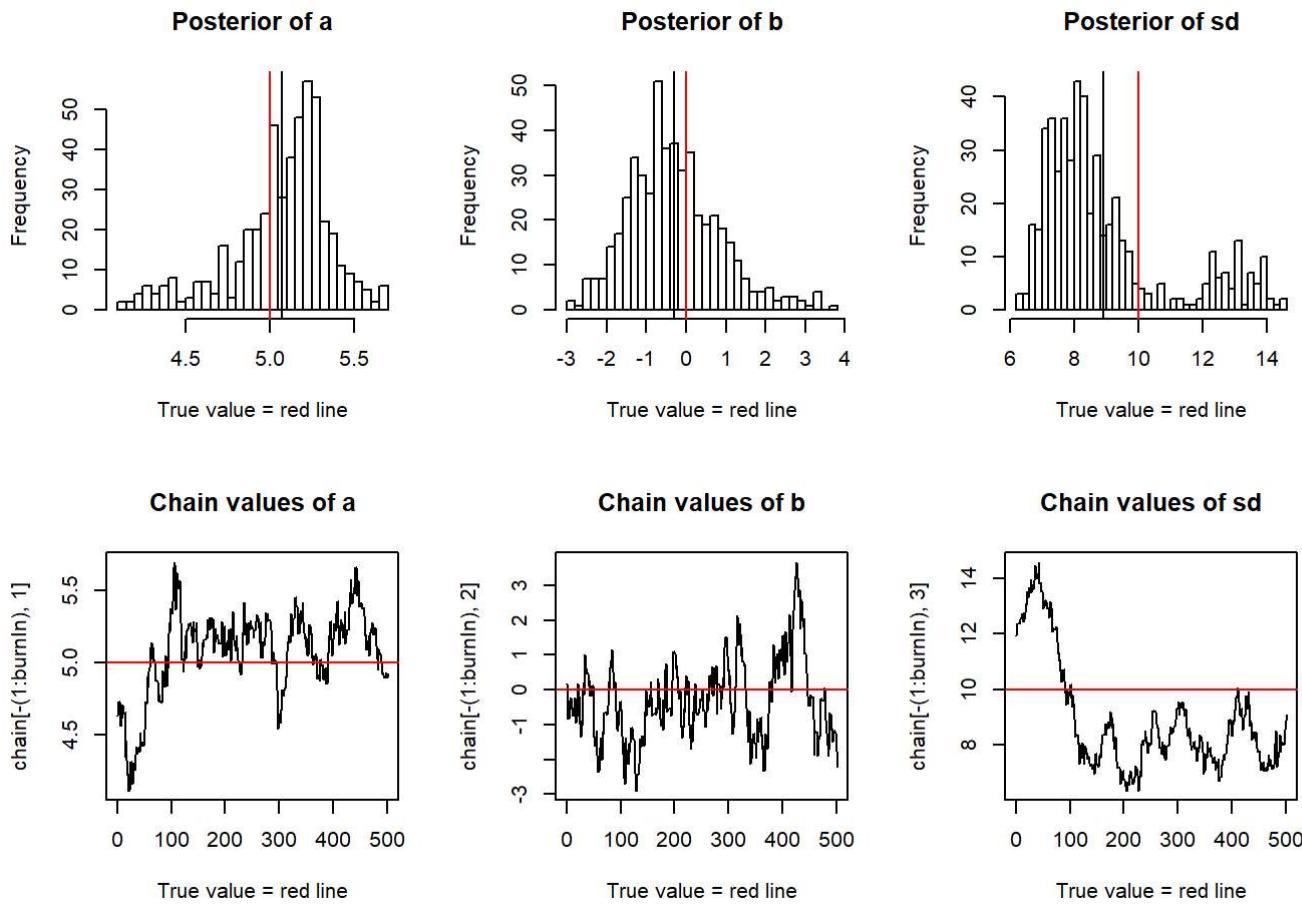
```
## [1] 4.7916426 0.2432614
```



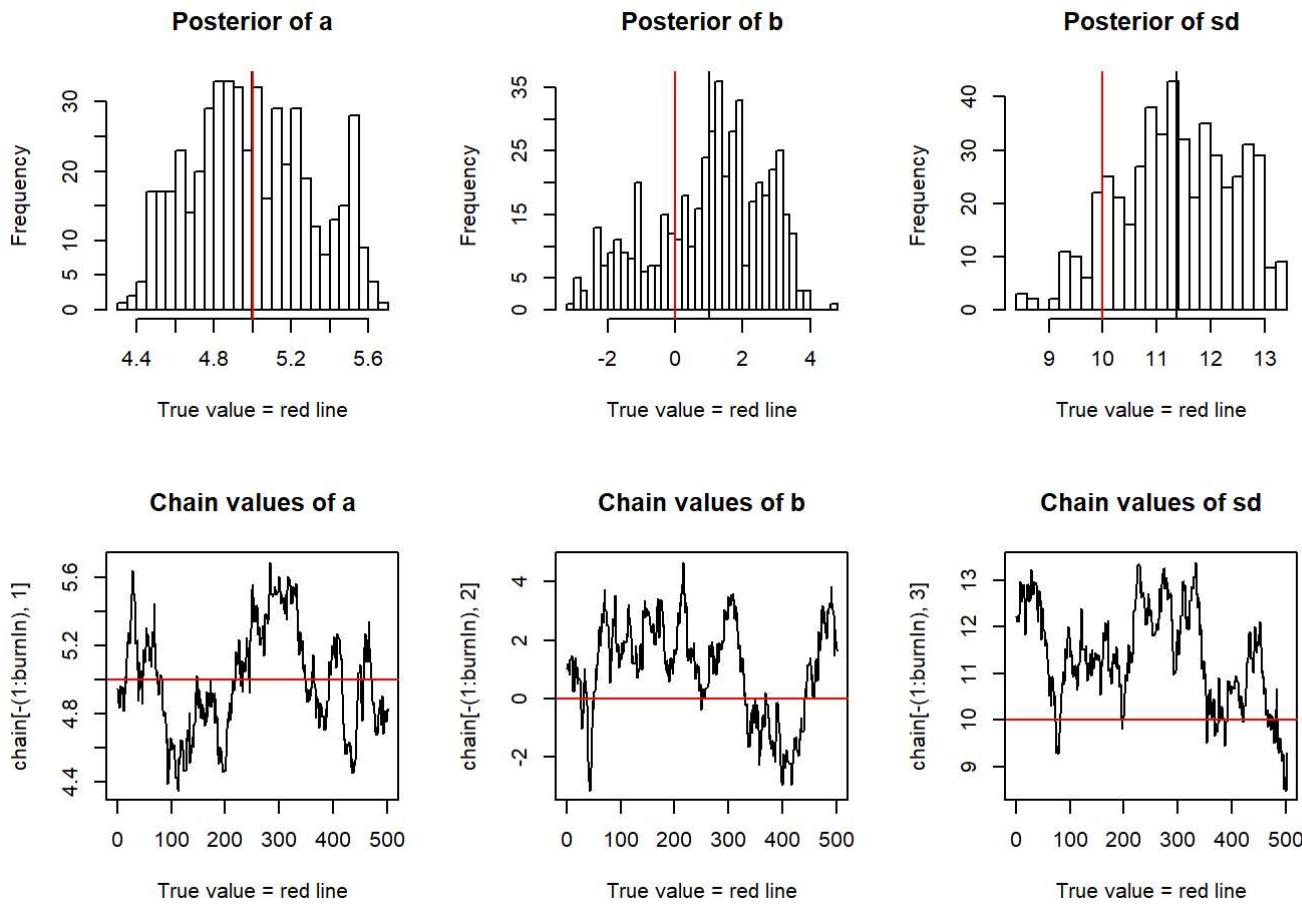
```
## [1] 4.993545 0.222838
```



```
## [1] 5.2507758 0.2535137
```



```
## [1] 5.0272210 0.3027354
```



```
## [1] 4.9128232 0.2959408
```

```
avg_avg <- mean(a_vals2[,1])
avg_std <- sqrt(mean((a_vals2[,2])^2))
#print(a_vals2)
paste("average of the averages", "average of the standard deviations")
```

```
## [1] "average of the averages average of the standard deviations"
```

```
paste(avg_avg, avg_std)
```

```
## [1] "4.89395582280147 0.245449442584897"
```

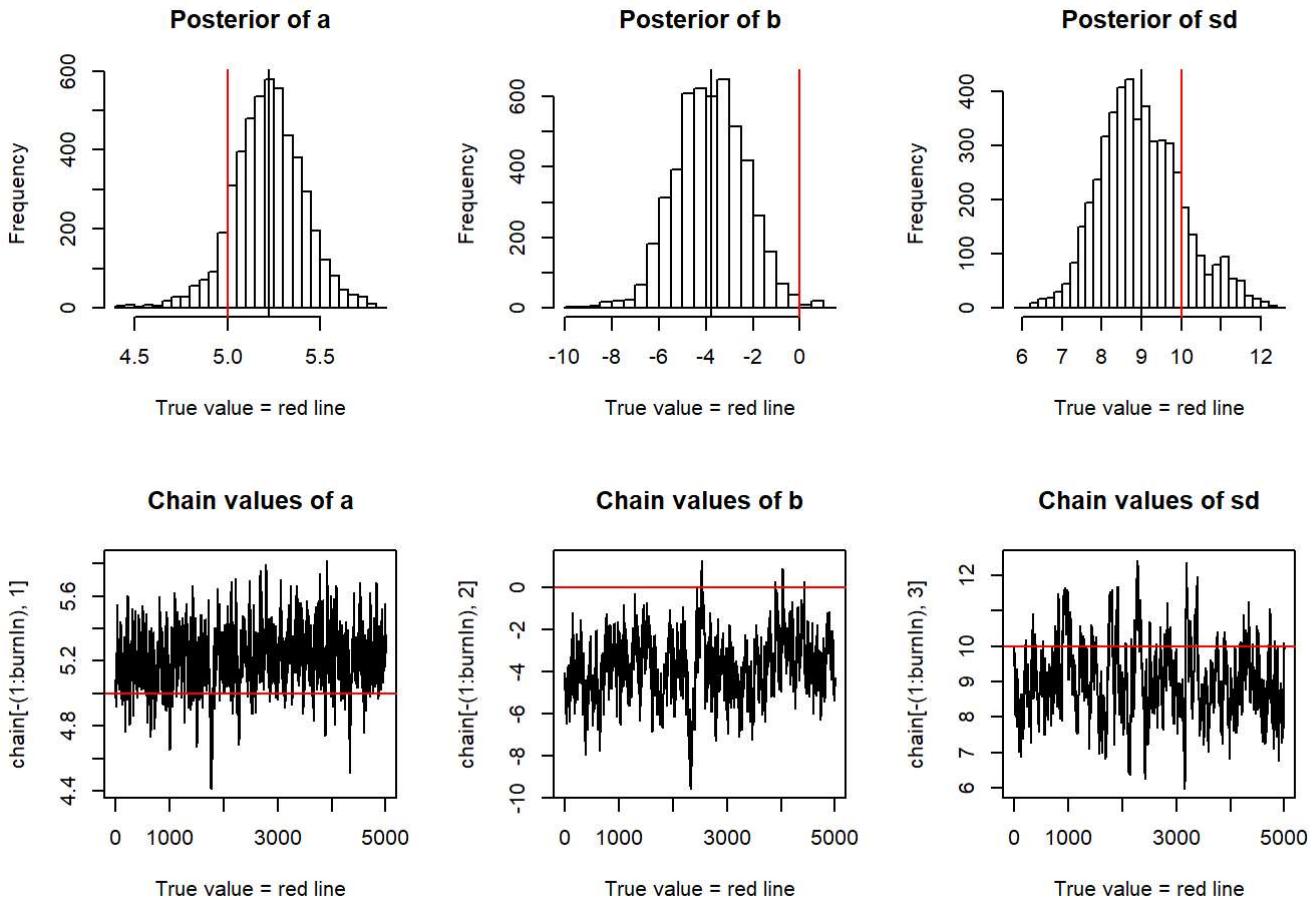
```
print("*****")
```

```
## [1] "*****"
```

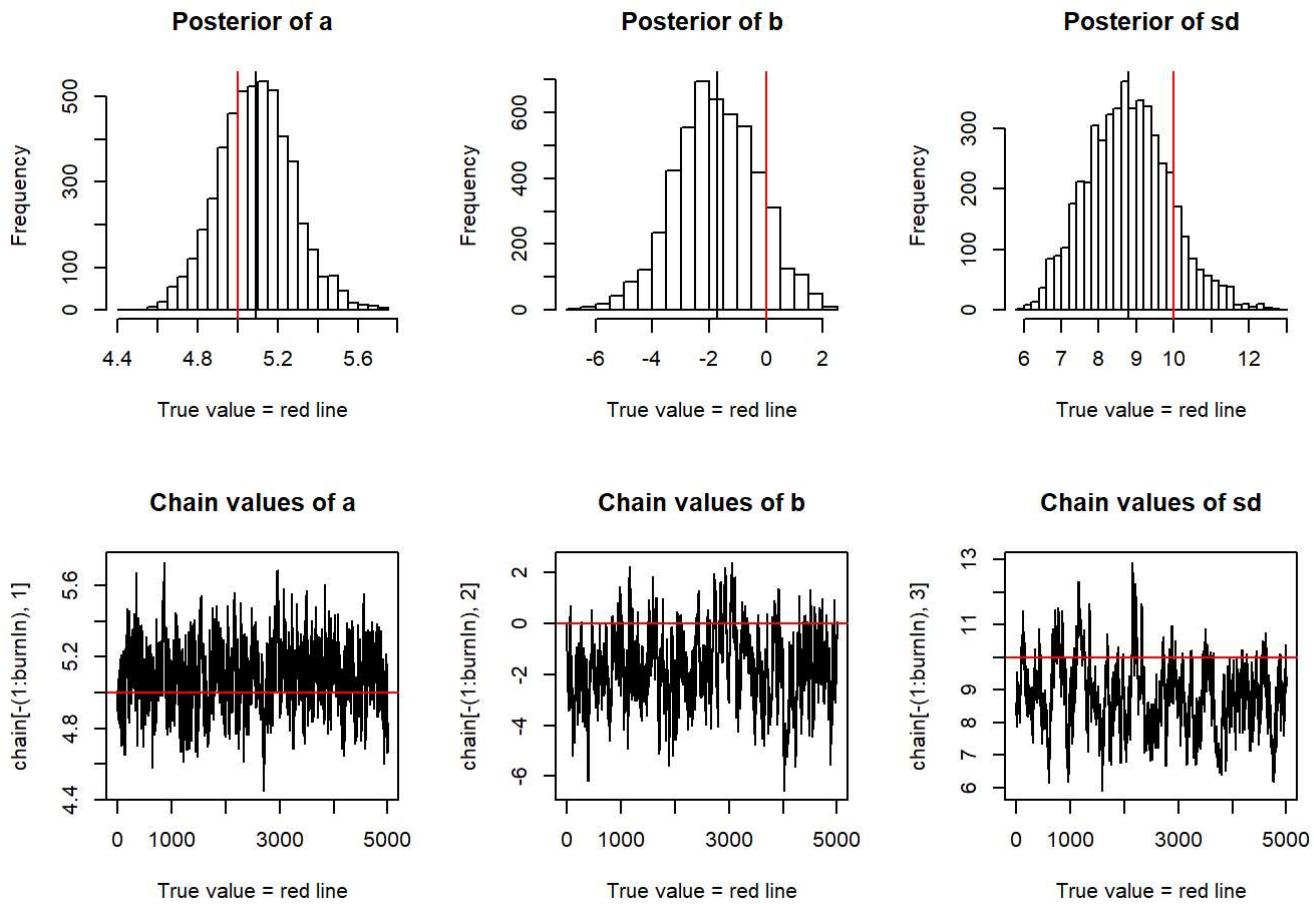
```
print("10000 iterations, 5000 burnIn")
```

```
## [1] "10000 iterations, 5000 burnIn"
```

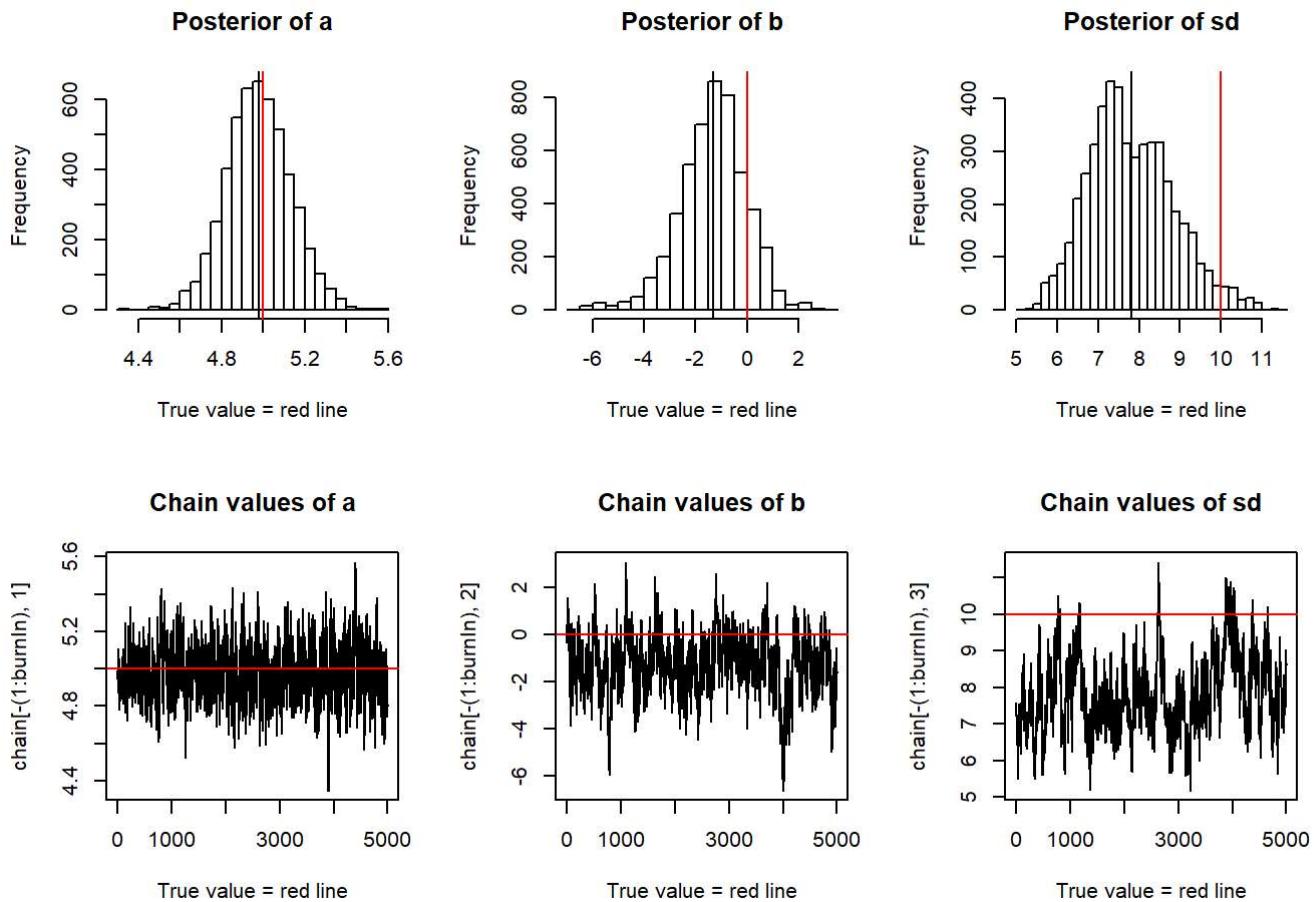
```
a_vals2 <- compare_outcomes(10000, 5000)
```



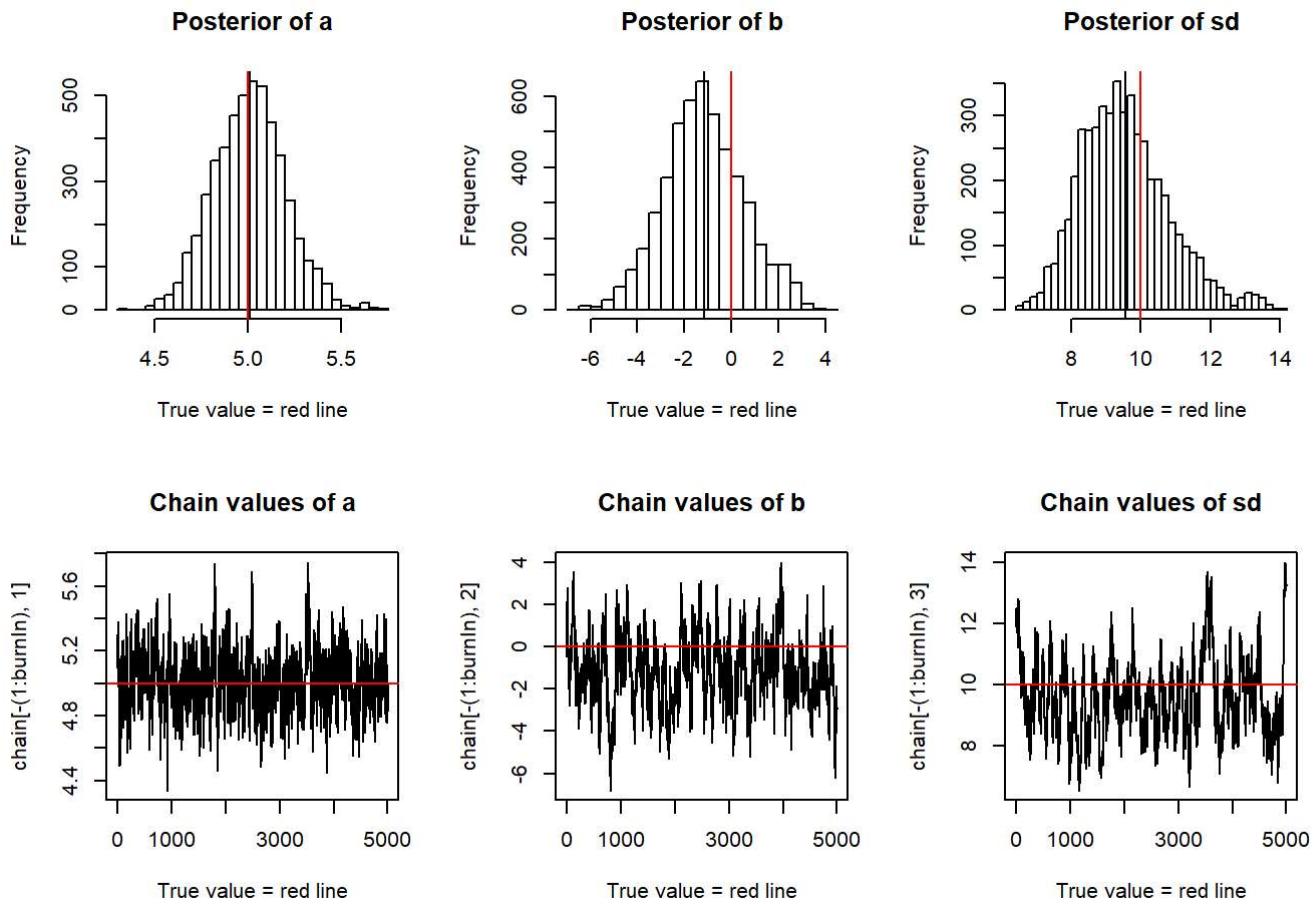
```
## [1] 5.222411 0.193915
```



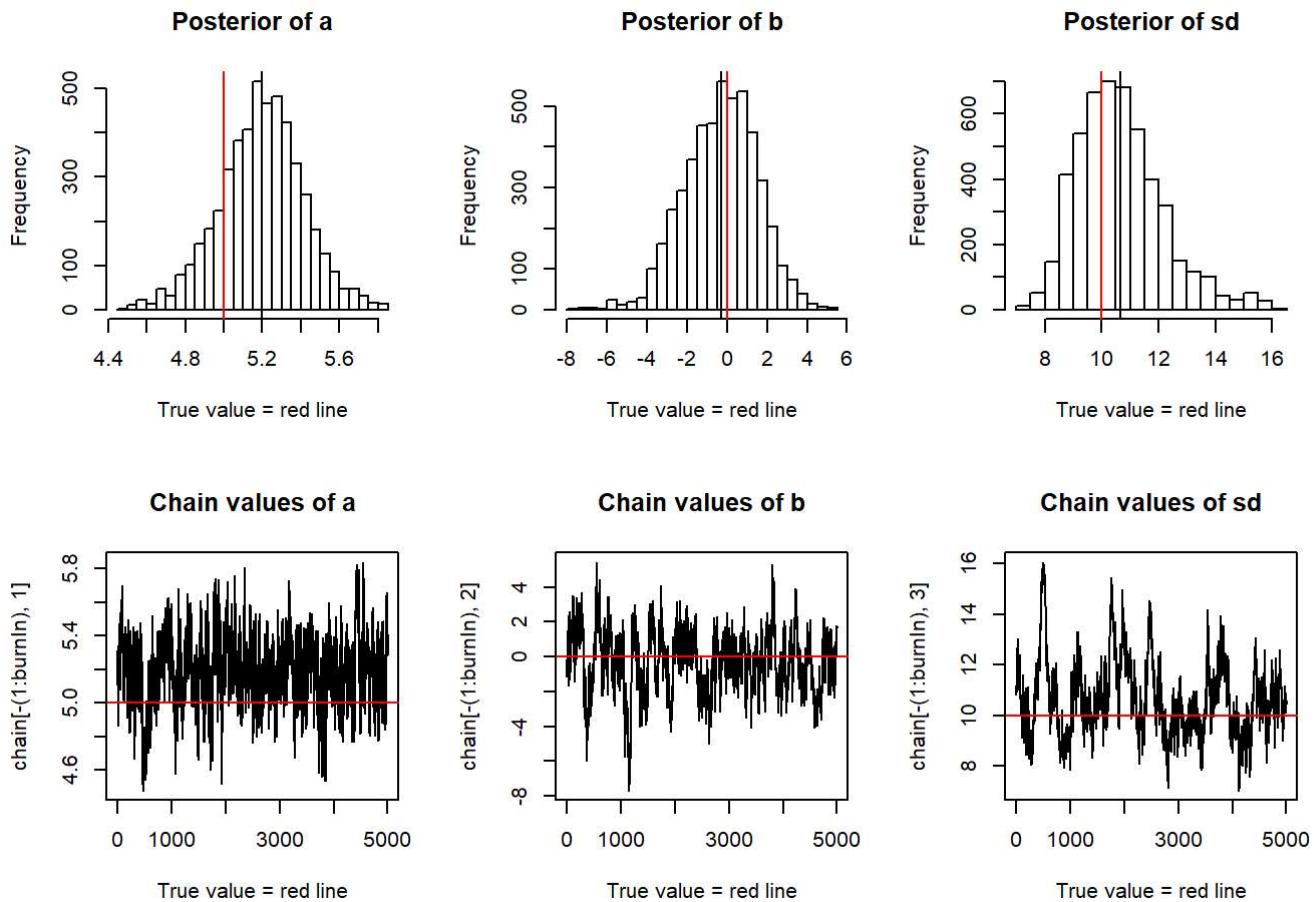
```
## [1] 5.0902570 0.1916809
```



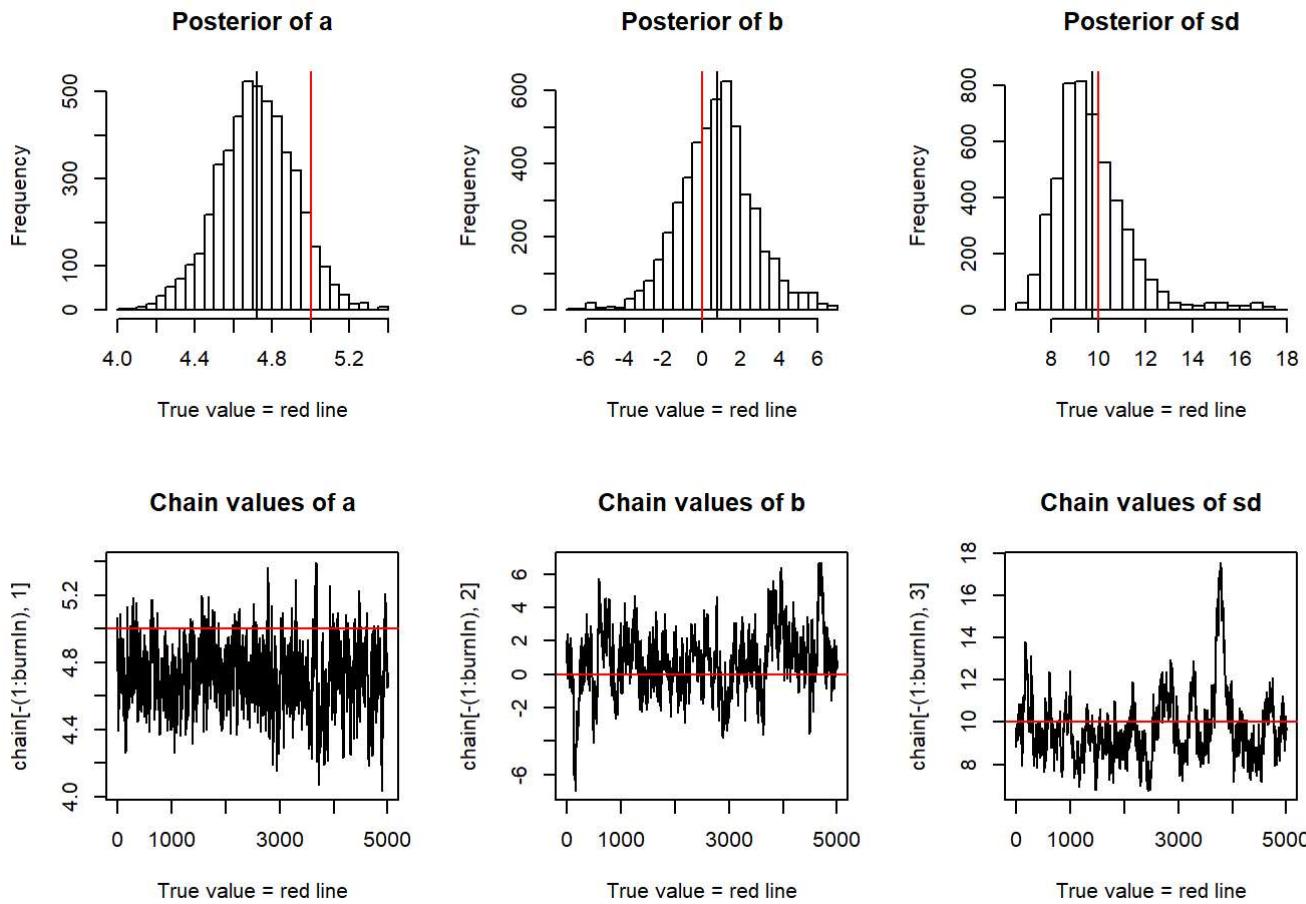
```
## [1] 4.9614464 0.1650334
```



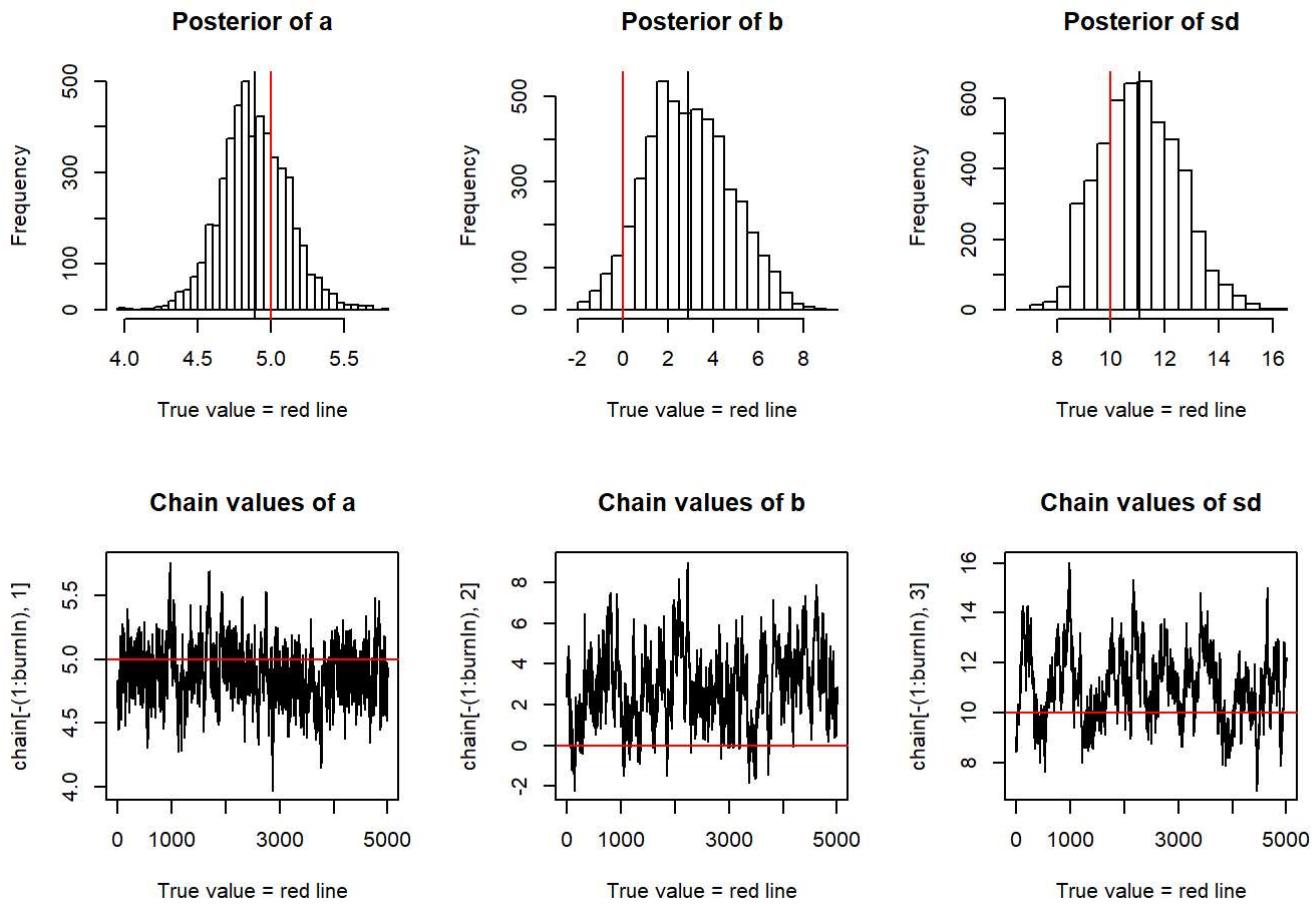
```
## [1] 5.0128368 0.1954781
```



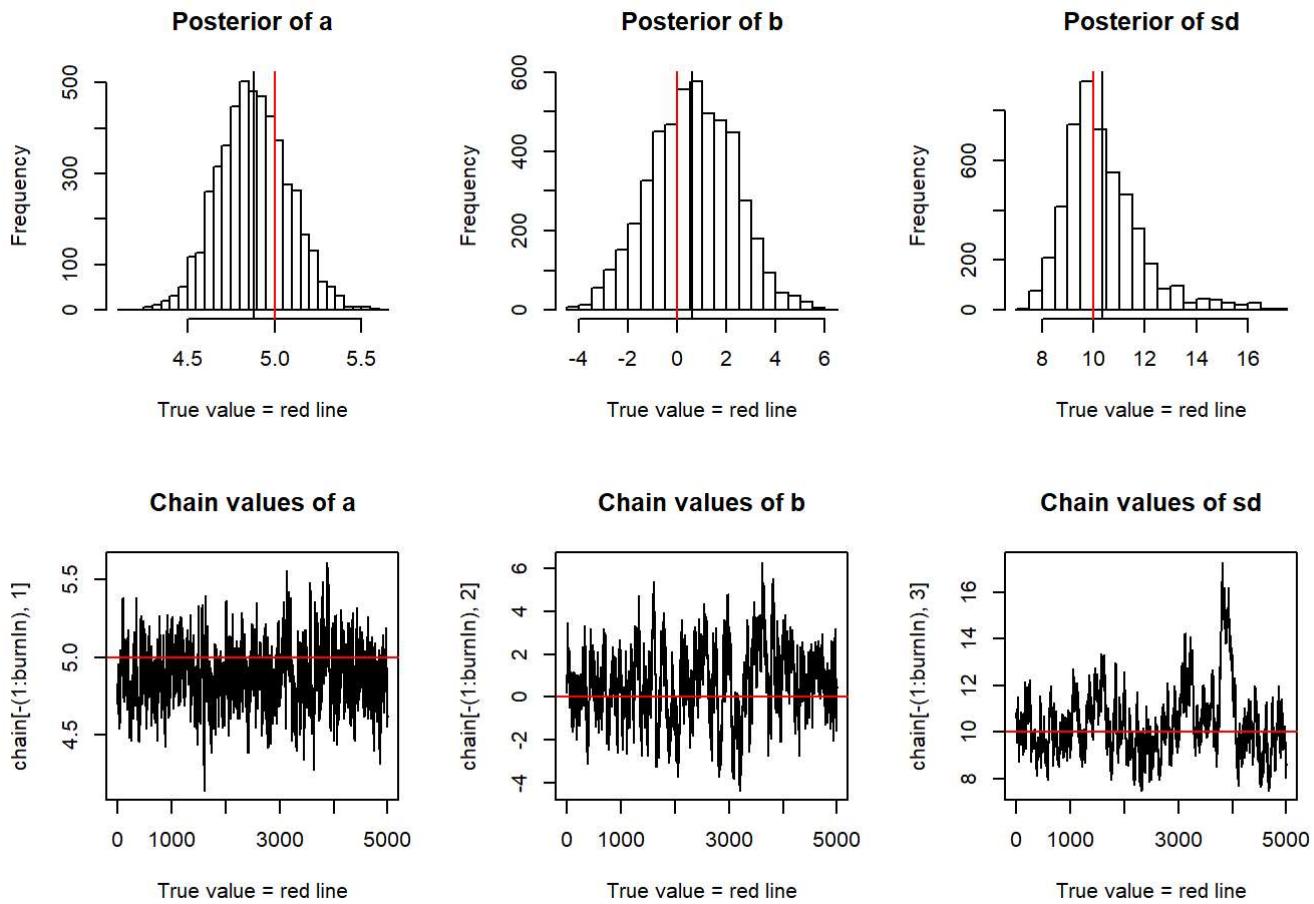
```
## [1] 5.1936619 0.2272689
```



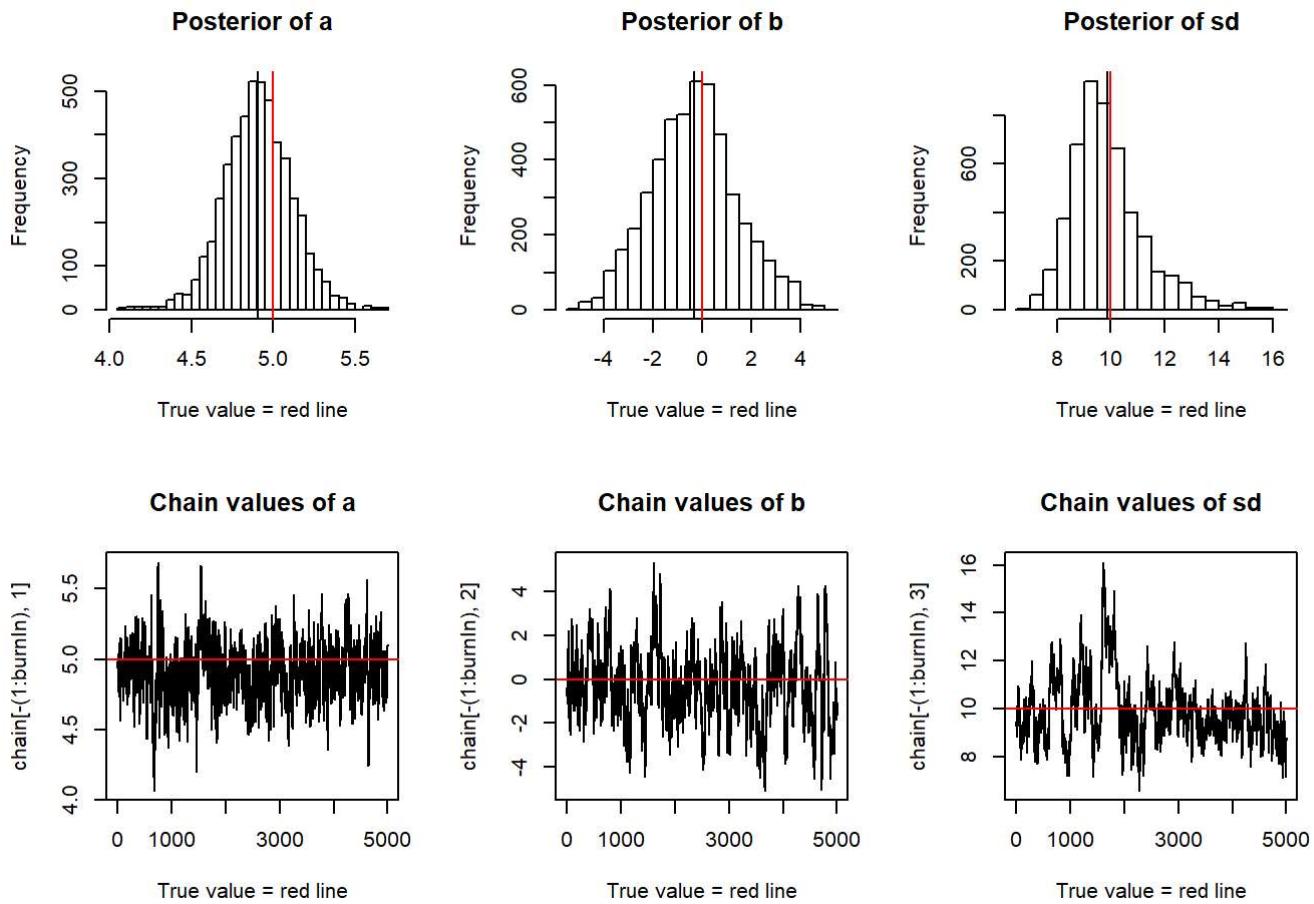
```
## [1] 4.7244901 0.1935424
```



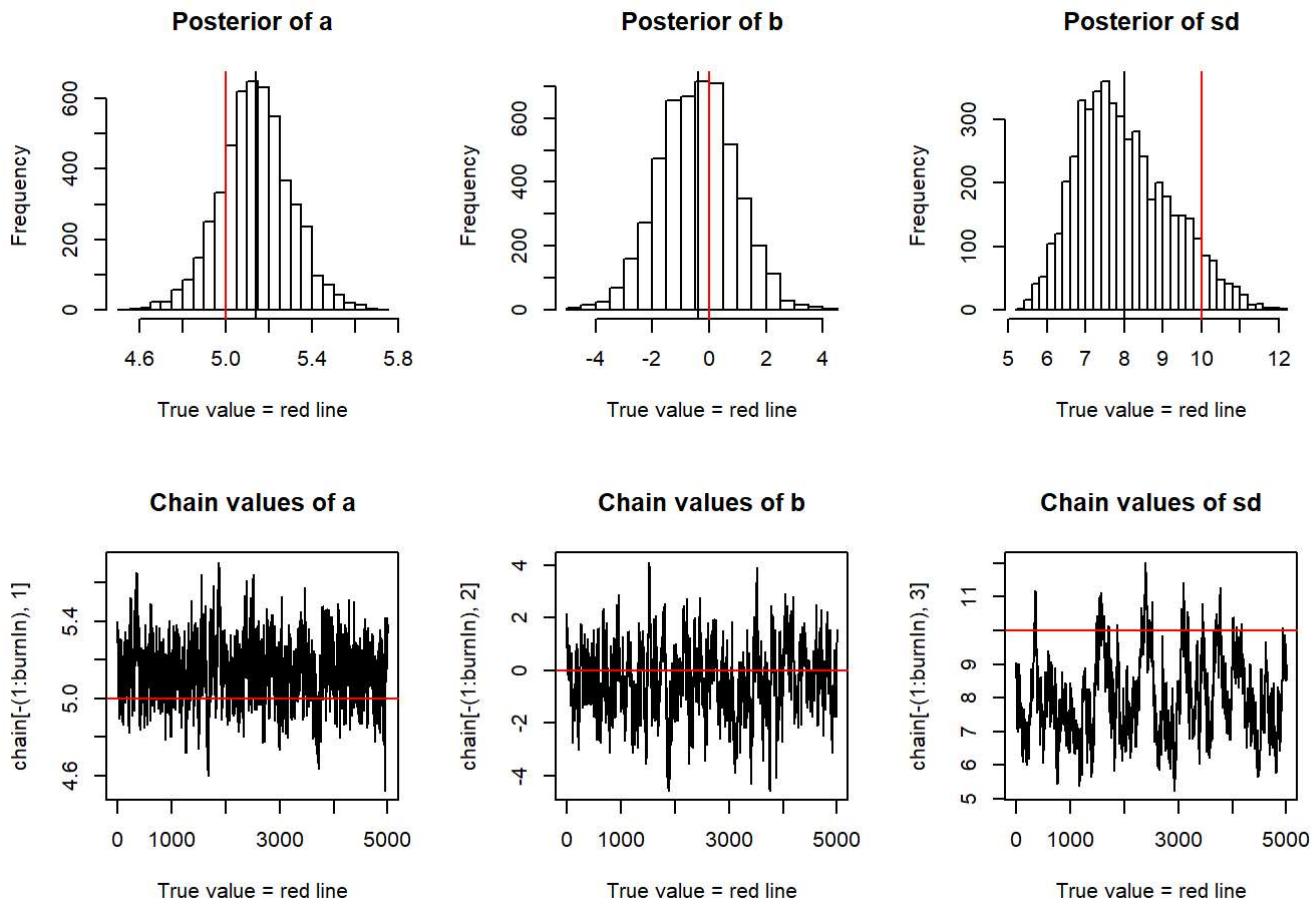
```
## [1] 4.8662764 0.2385324
```



```
## [1] 4.8951728 0.2157685
```



```
## [1] 4.8860795 0.2048756
```



```
## [1] 5.1321350 0.1690884
```

```
avg_avg <- mean(a_vals2[,1])
avg_std <- sqrt(mean((a_vals2[,2])^2))
#print(a_vals2)
paste("average of the averages", "average of the standard deviations")
```

```
## [1] "average of the averages average of the standard deviations"
```

```
paste(avg_avg, avg_std)
```

```
## [1] "4.99847669946547 0.200728857398037"
```

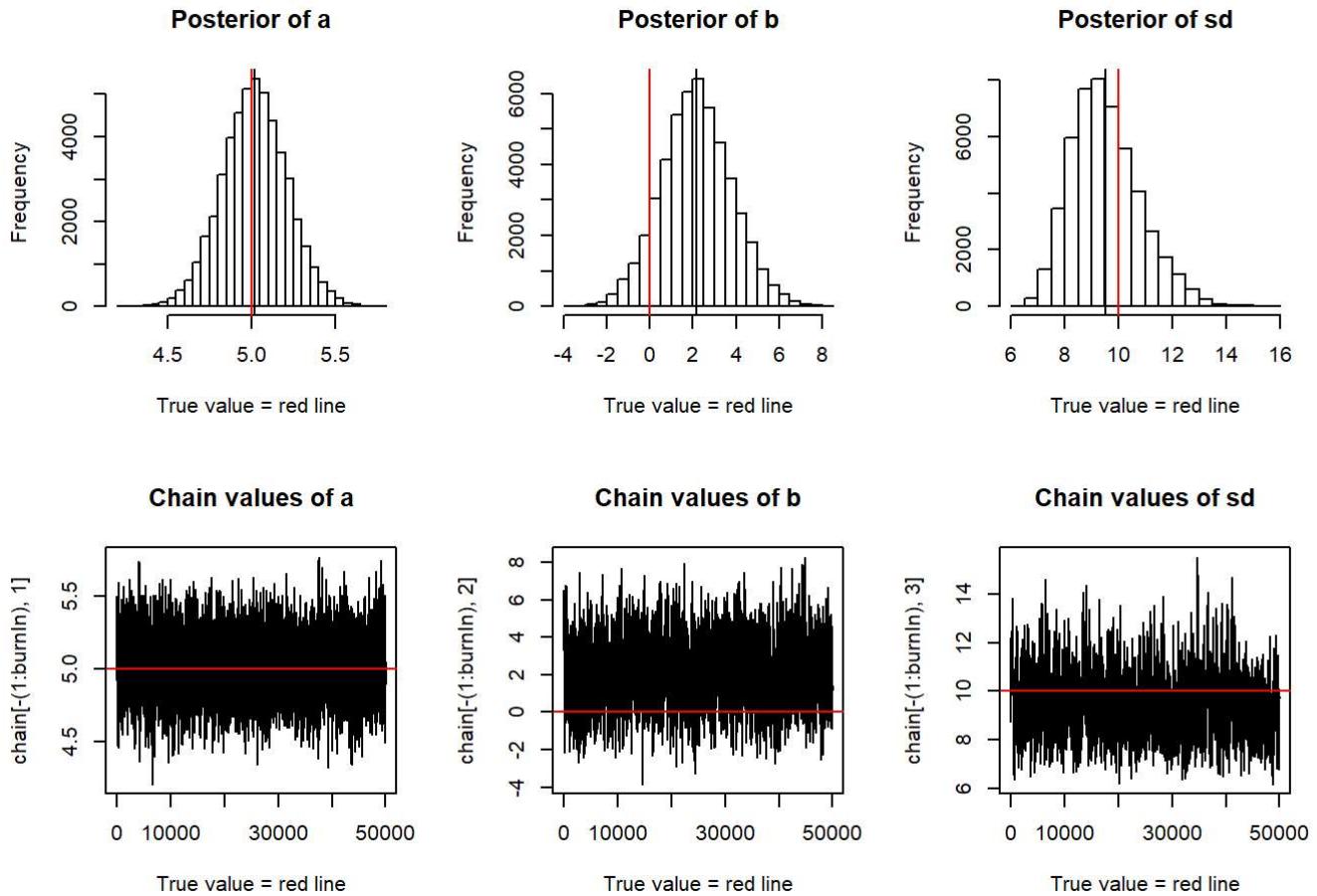
```
print("*****")
```

```
## [1] "*****"
```

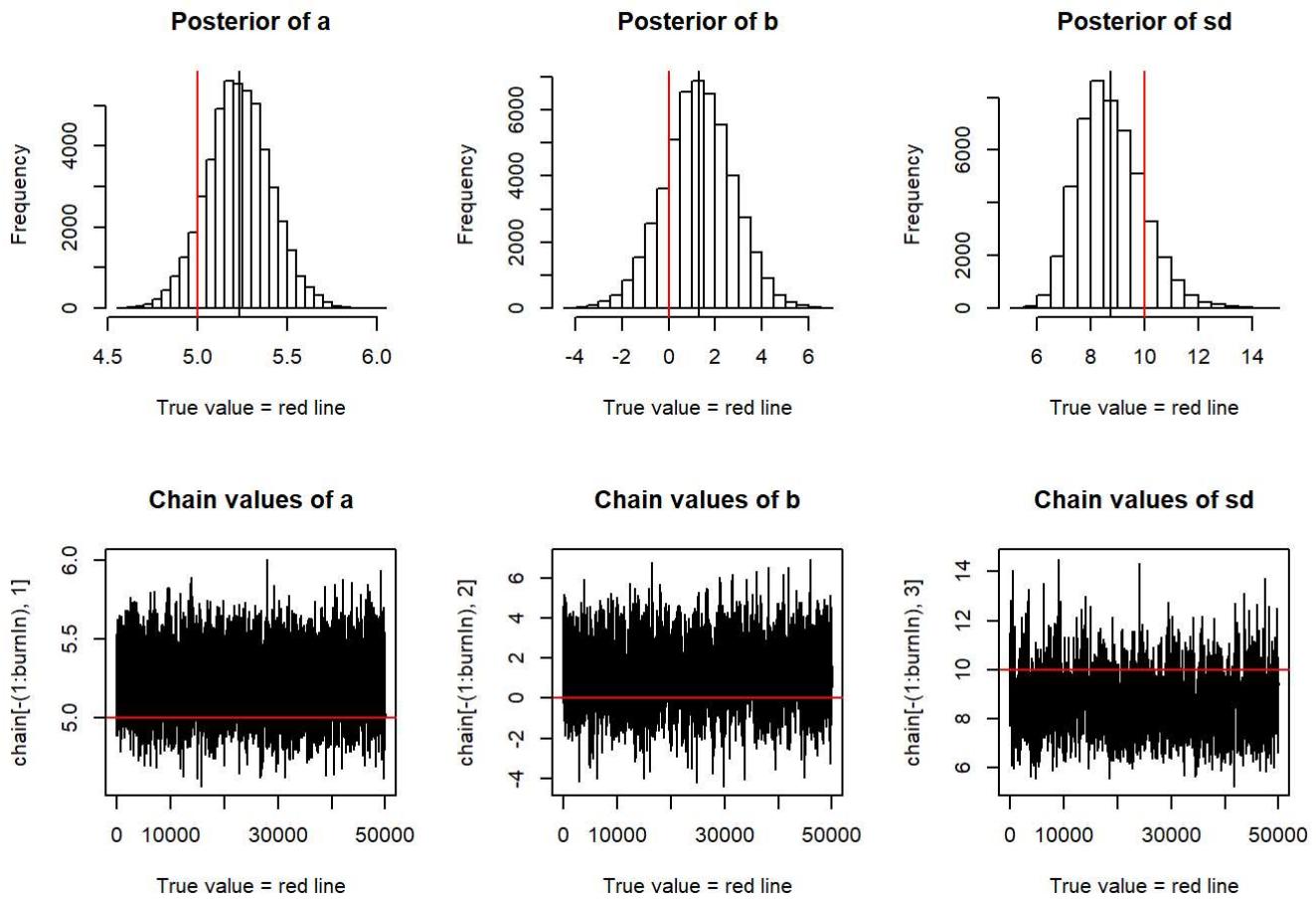
```
print("100000 iterations, 50000 burnIn")
```

```
## [1] "100000 iterations, 50000 burnIn"
```

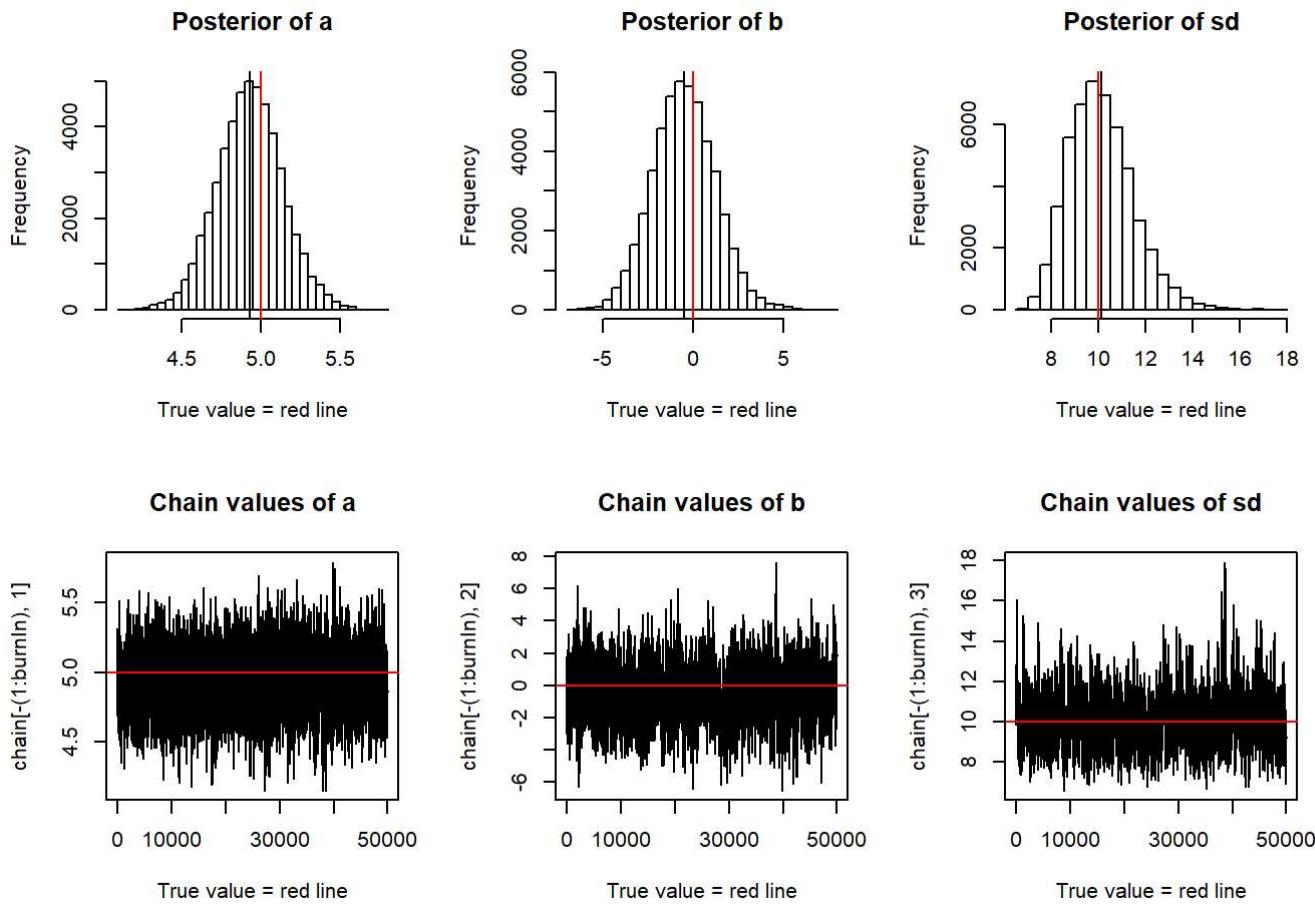
```
a_vals2 <- compare_outcomes(100000, 50000)
```



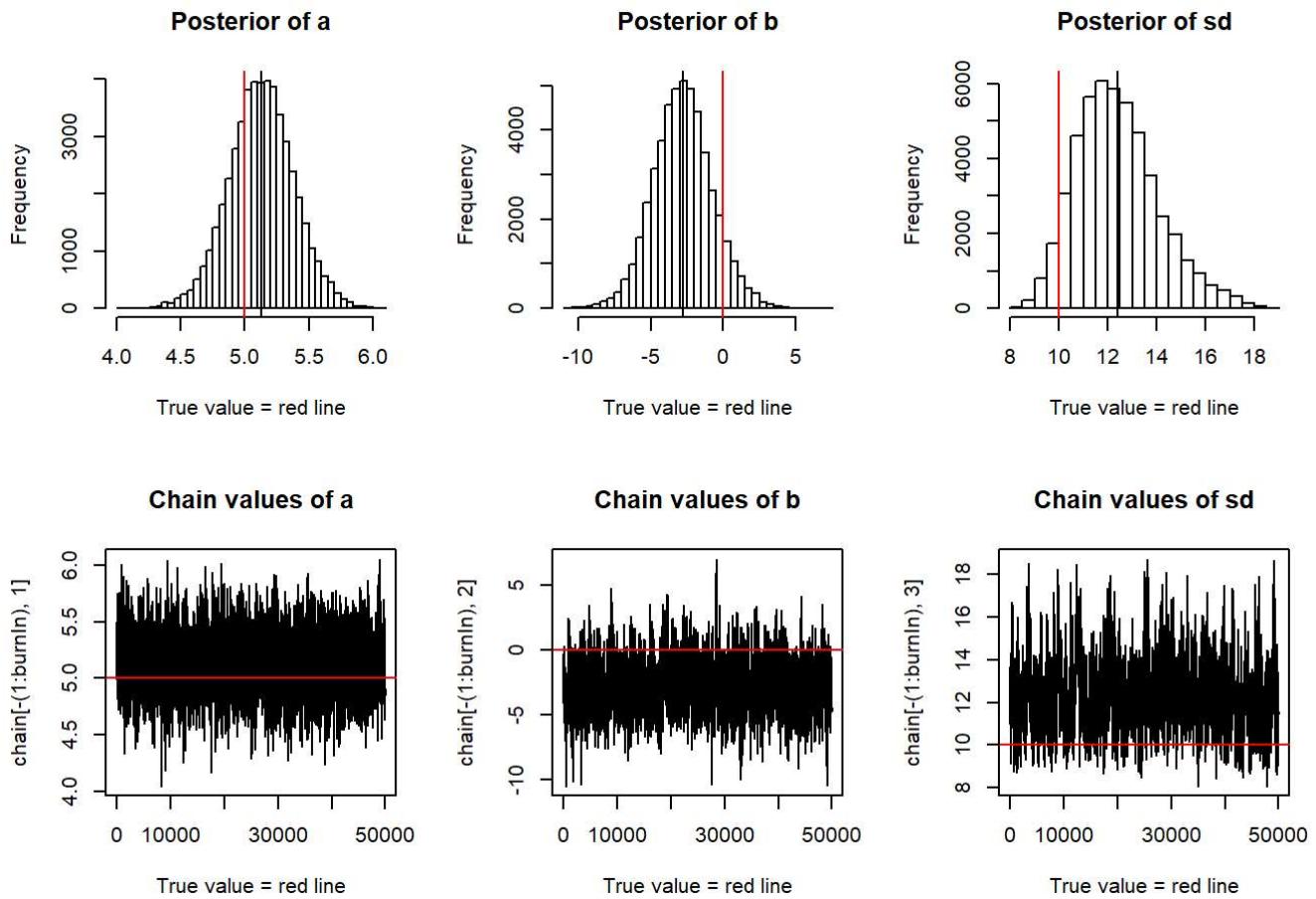
```
## [1] 5.0188898 0.1938202
```



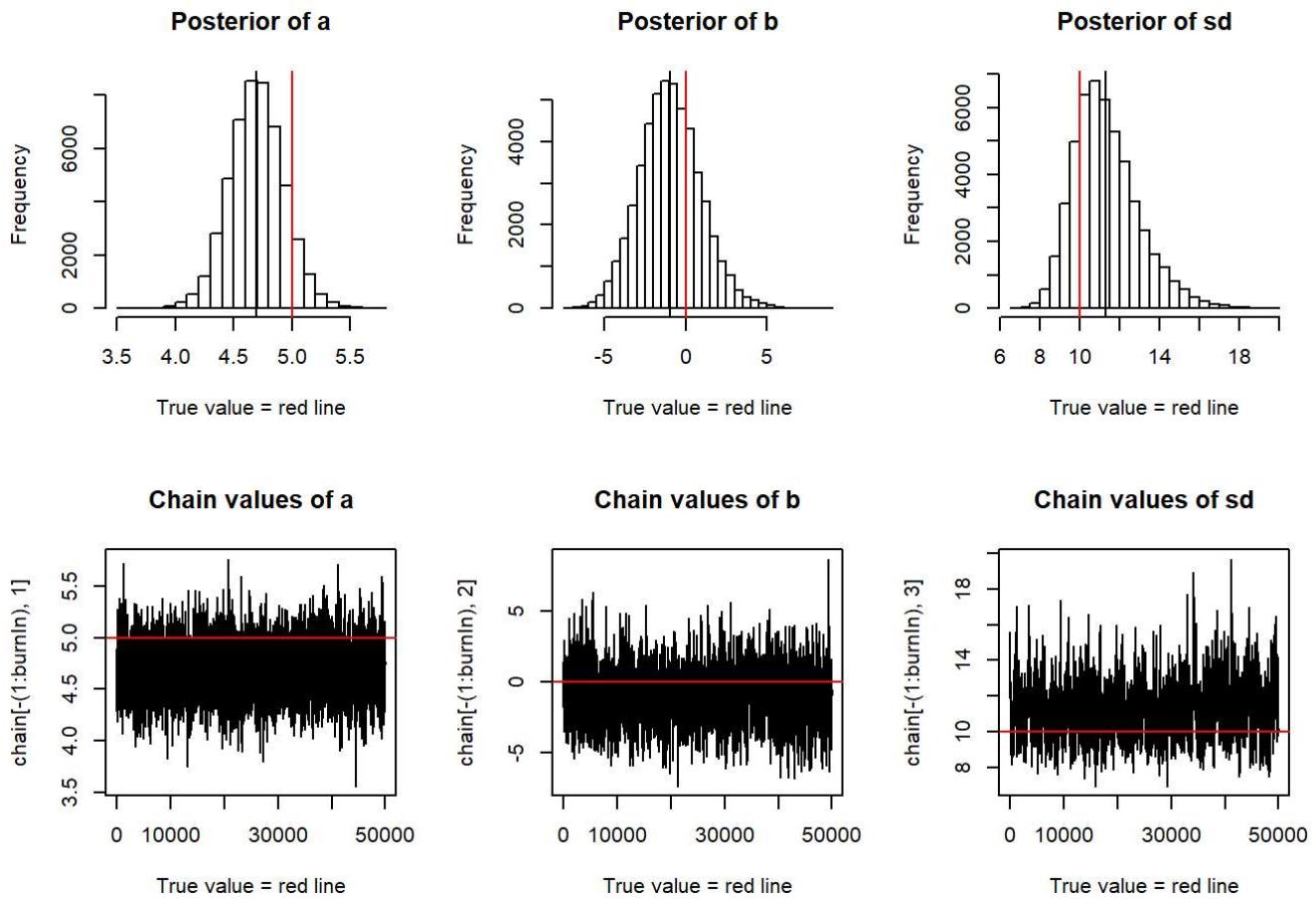
```
## [1] 5.2270376 0.1791635
```



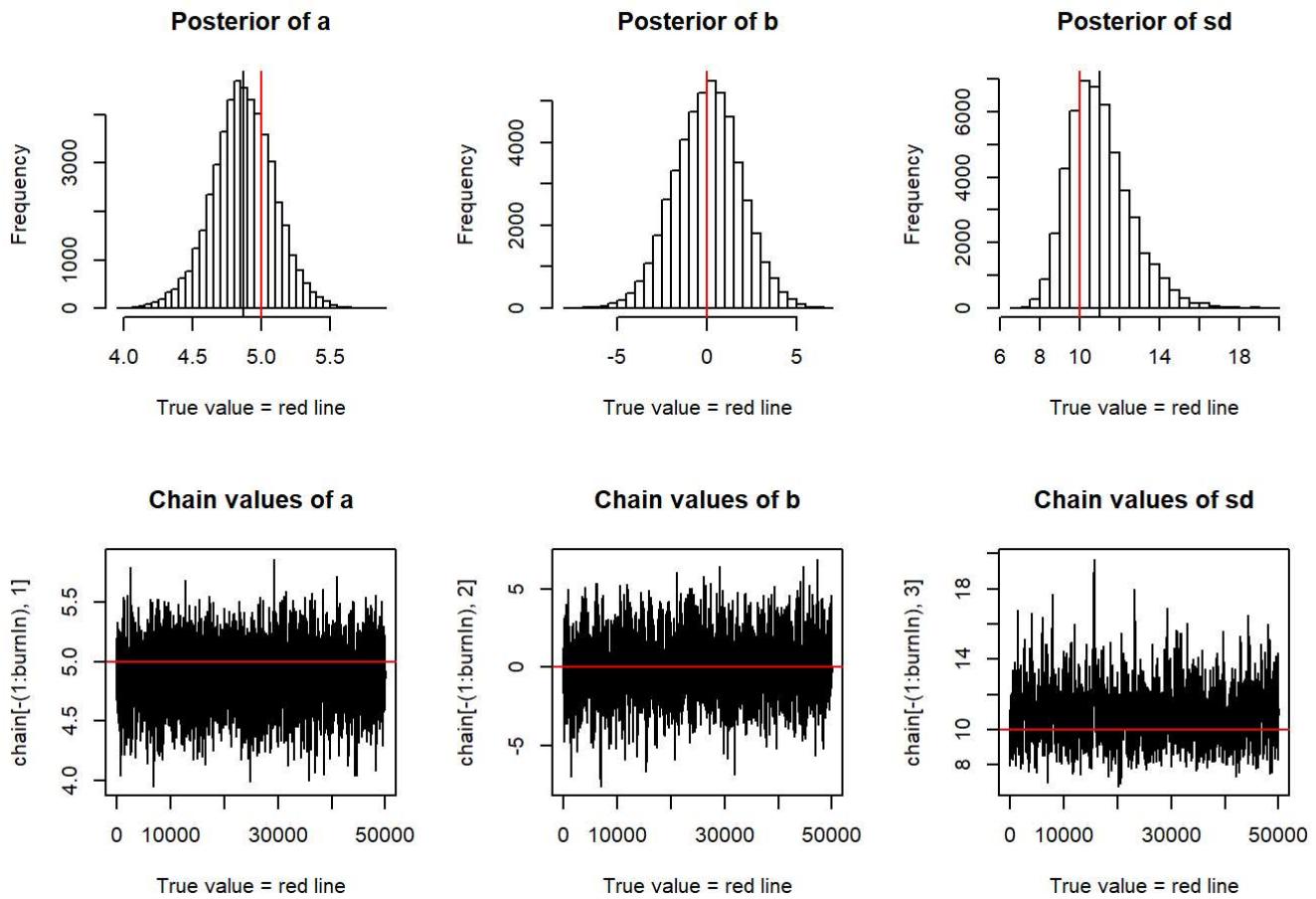
```
## [1] 4.9316676 0.2040731
```



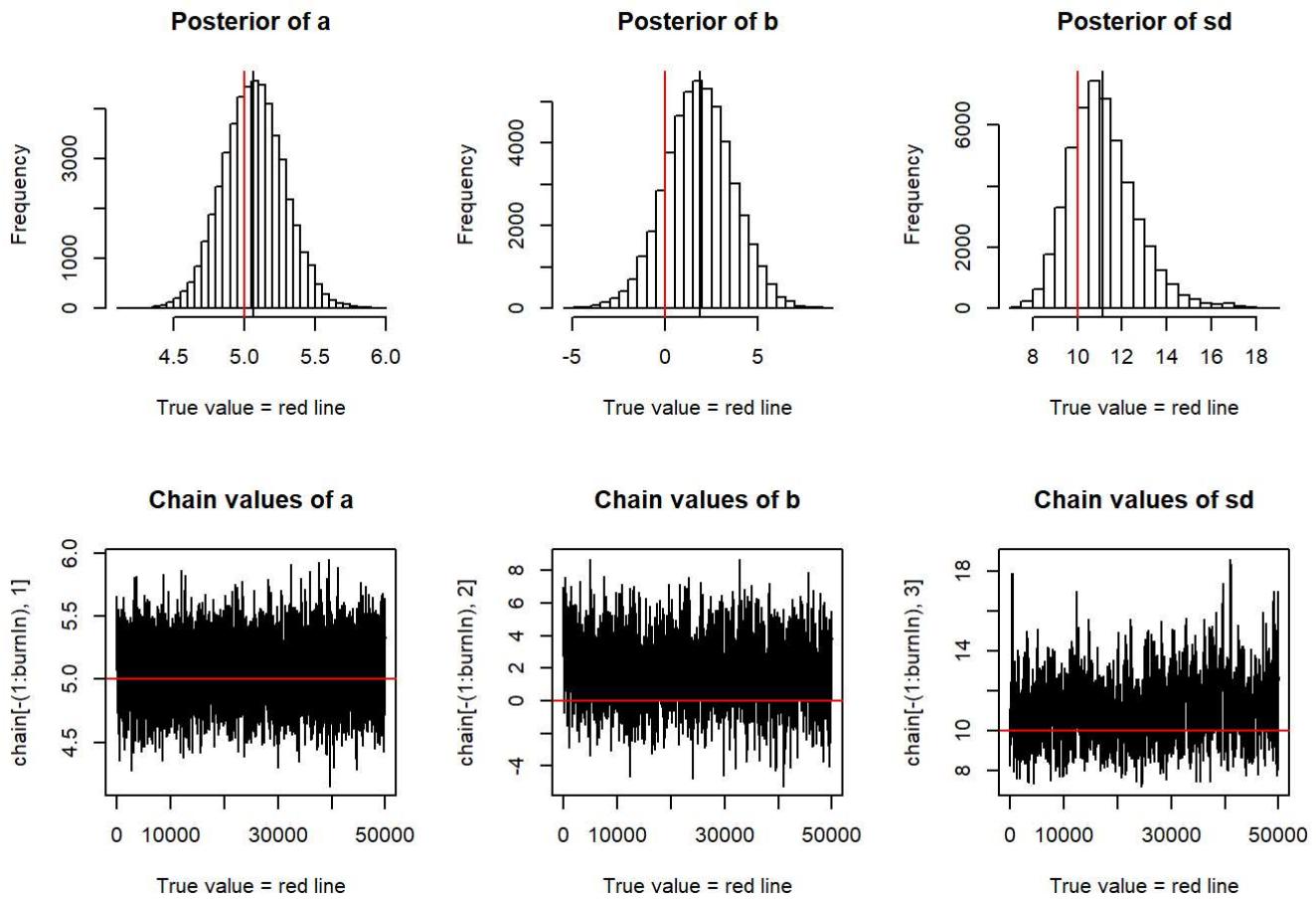
```
## [1] 5.1309959 0.2562959
```



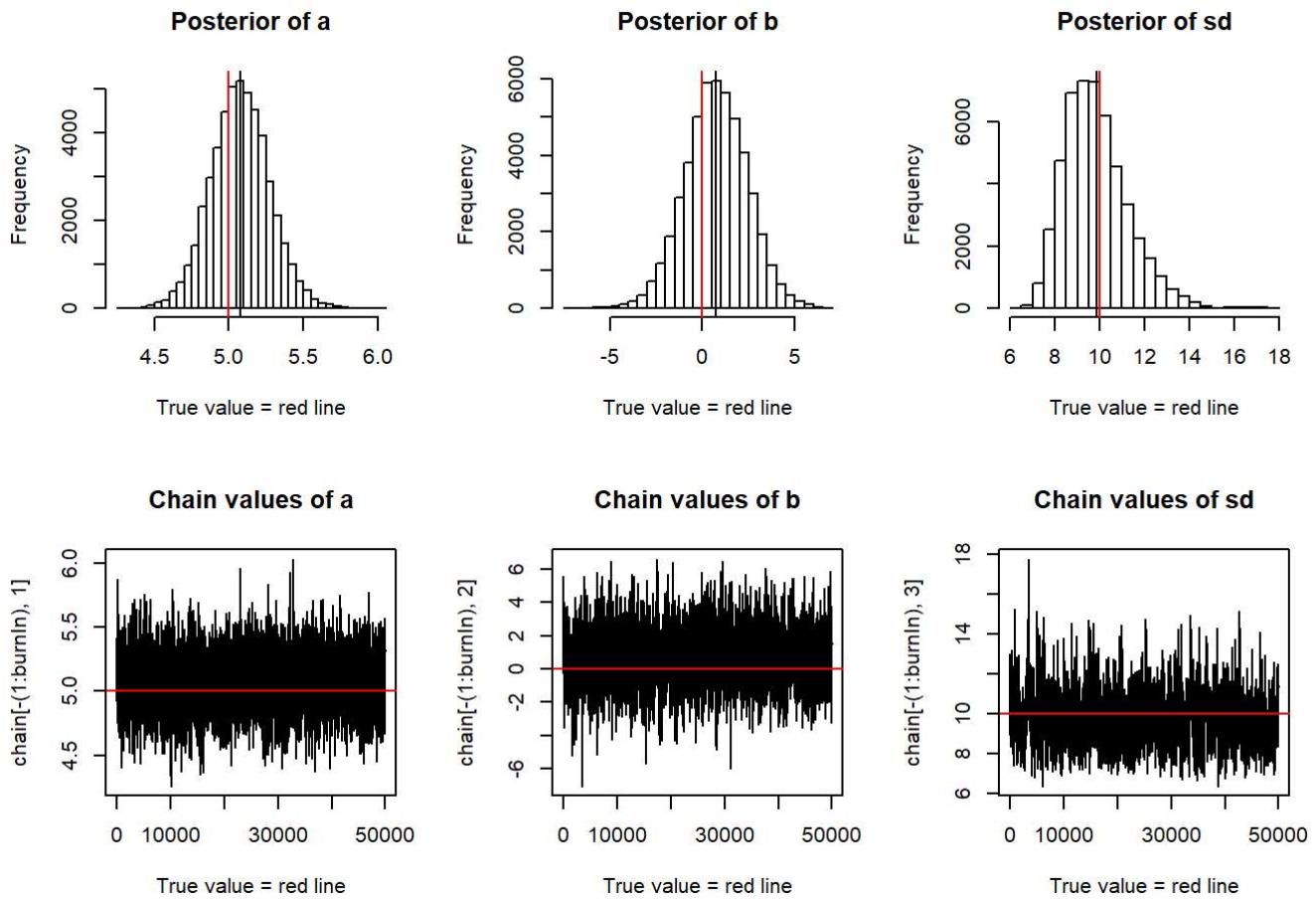
```
## [1] 4.6942212 0.2310126
```



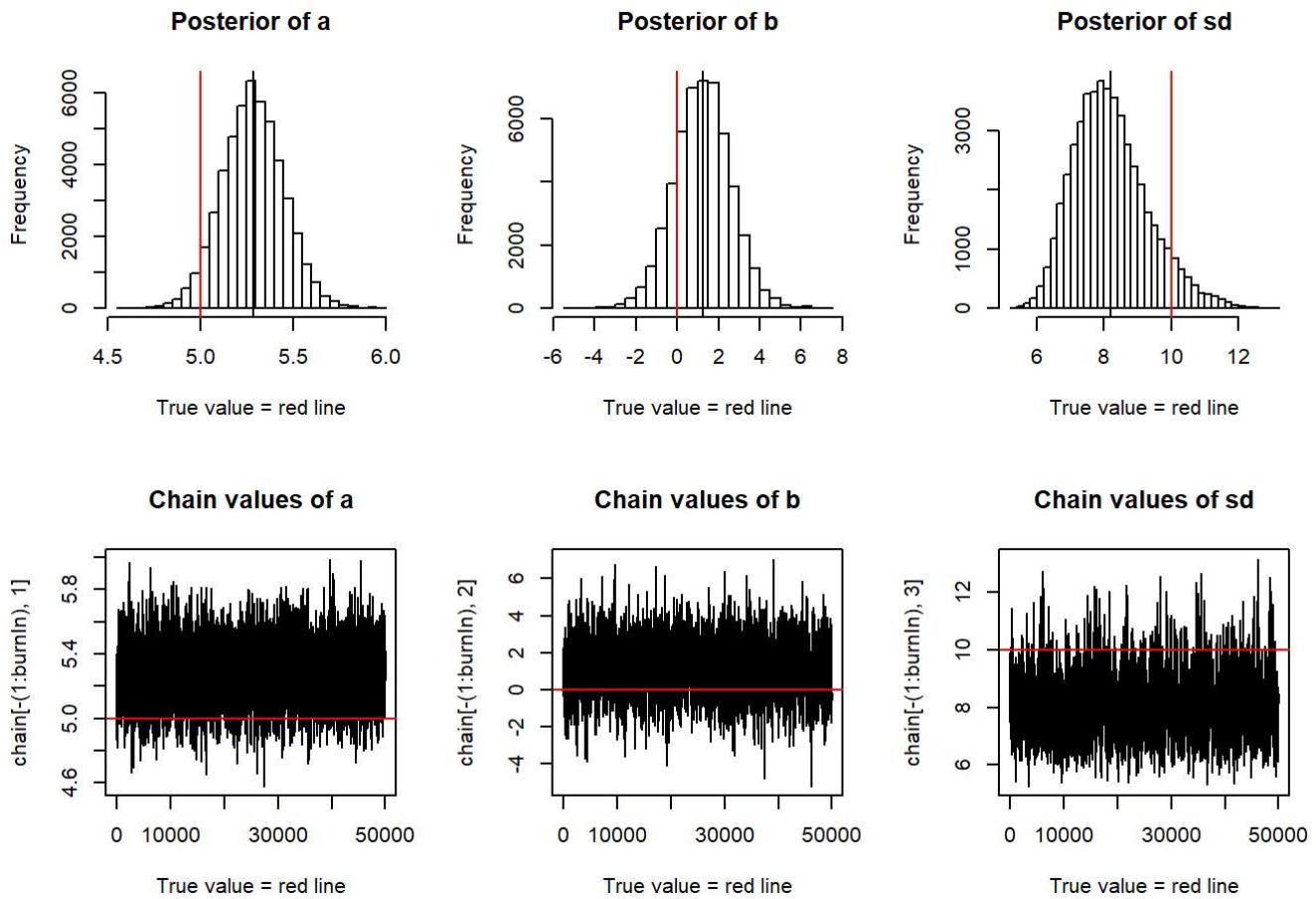
```
## [1] 4.8709382 0.2223533
```



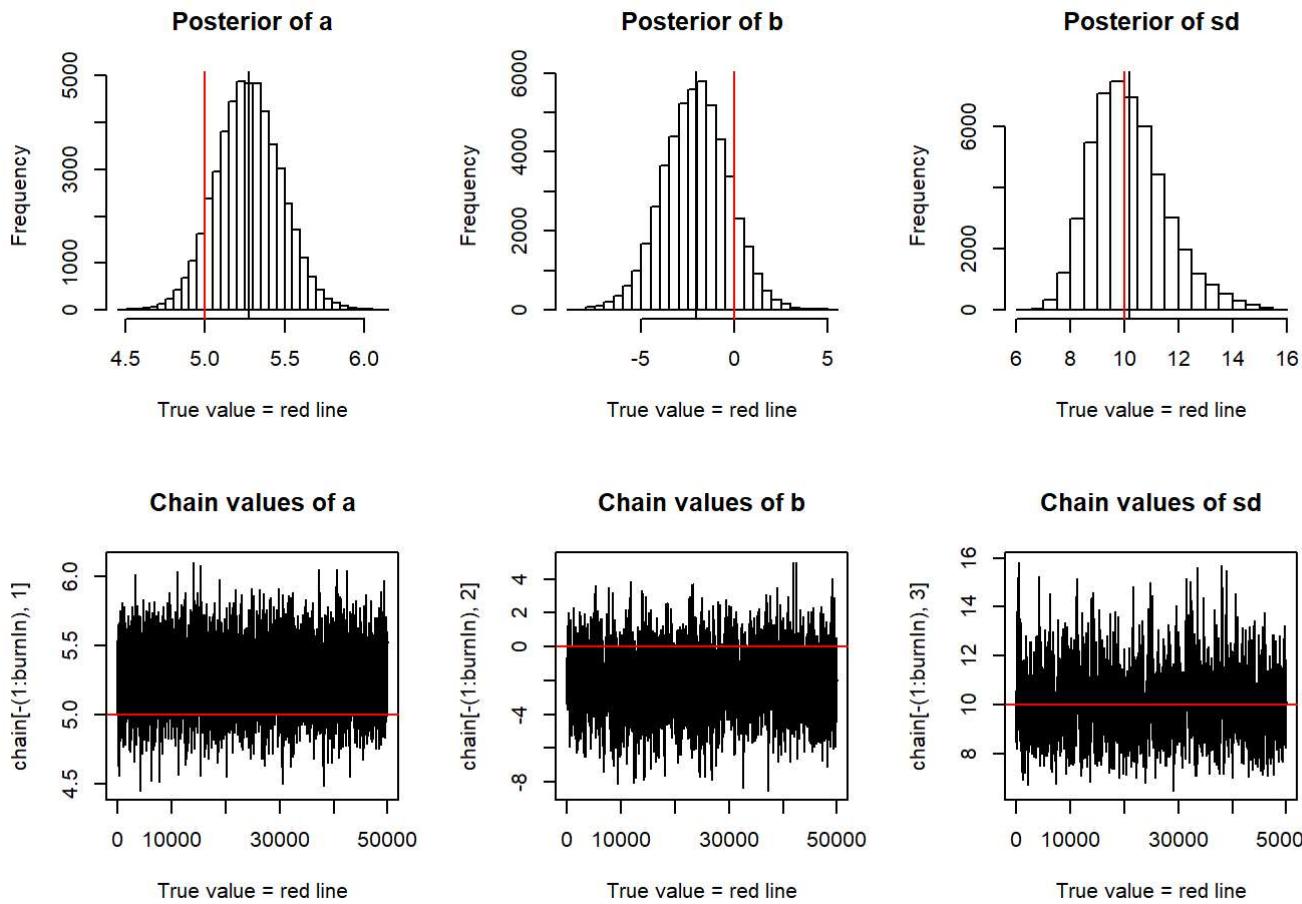
```
## [1] 5.071898 0.222090
```



```
## [1] 5.0806936 0.1986843
```



```
## [1] 5.2869204 0.1668838
```



```
## [1] 5.2751339 0.2054998
```

```
avg_avg <- mean(a_vals2[,1])
avg_std <- sqrt(mean((a_vals2[,2])^2))
#print(a_vals2)
paste("average of the averages", "average of the standard deviations")
```

```
## [1] "average of the averages average of the standard deviations"
```

```
paste(avg_avg, avg_std)
```

```
## [1] "5.05883959949988 0.209448524767978"
```

I do not notice any particular changes in either the estimate or the standard deviation of *a*. I do notice that the histogram for *a* becomes more sharply peaked as well as the fact that the true estimate is better centered towards the middle of the plot of chain. This effect is observed as one transitions both between 1000 iterations with 500 burnln to 10000 iterations with 5000 burnln, as well as between the latter and 100000 with 50000 burnln.