sub\_condition\_two|stat\_comp\_report

ZJH

2022-06-10

# init

library(boa)  
library(coda)  
library(assertthat)  
library(invgamma)  
library(LearnBayes)  
data("birthweight")  
library(lattice)  
library(tidyverse)

## Warning: replacing previous import 'lifecycle::last\_warnings' by  
## 'rlang::last\_warnings' when loading 'pillar'

## Warning: replacing previous import 'lifecycle::last\_warnings' by  
## 'rlang::last\_warnings' when loading 'tibble'

## Warning: replacing previous import 'lifecycle::last\_warnings' by  
## 'rlang::last\_warnings' when loading 'hms'

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.1 v dplyr 1.0.7  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x tibble::has\_name() masks assertthat::has\_name()  
## x dplyr::lag() masks stats::lag()

# distributions

dinvgamma <- function(x,shape,scale){  
 assert\_that(x!=0)  
 assert\_that(shape!=0)  
 fx <- (scale^shape\*exp(-scale/x)\*x^(-shape-1))/gamma(shape)  
 return(fx)}  
Cauchy\_plus <- function(x,b\_cauchy){  
 assert\_that(b\_cauchy>0)  
 fx <- 2/pi/(b\_cauchy\*(1+(x/b\_cauchy)^2)\*(x>0))  
 return(fx)  
}  
prob <- function(x,shape,scale,b\_cauchy){  
 fx <- dinvgamma(x^2,shape =shape,scale=scale)\*Cauchy\_plus(x,b\_cauchy=b\_cauchy)  
 assert\_that(dinvgamma(x^2,shape =shape,scale=scale)!=0)  
 assert\_that(Cauchy\_plus(x,b\_cauchy=b\_cauchy)!=0)  
 assert\_that(fx!=0)  
 return(fx)  
}

# function

draw <- function(fx,Observation,...)  
{  
 hist(Observation,nclass=25,freq = FALSE,main="")  
 curve(fx(x,...),add=TRUE,lty = 2, lwd = 2,col='red')  
}

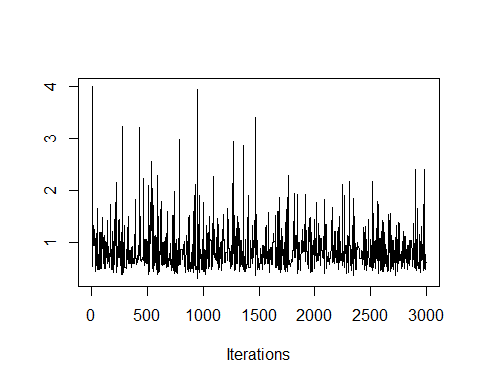
# parameters for model

b\_tau <- 1# i don't know what actually is,for cauchy plus(0,b\_tau)  
a <- 1 # for proposal distribution inverse gamma(a,b)   
b <- 1 # for proposal distribution inverse gamma(a,b)

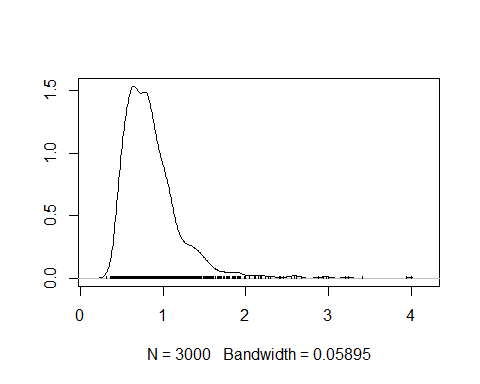
# parameters for simulation process

m <- 3000 #length of chain  
xt <- numeric(m)  
n=30  
# generate the observed sample  
x <- rinvgamma(n,shape = a,scale = b)  
# generate the independence sampler chain  
u <- runif(m)  
y <- rinvgamma(m,shape = a,scale = b) #proposal distribution  
xt[1] <- 1  
for (i in 2:m) {  
 fy <- prob(x = y[i],shape = a,scale = b,b\_cauchy = b\_tau)  
 fx <- prob(x = xt[i-1],shape = a,scale = b,b\_cauchy = b\_tau)  
 r <- prod(fy / fx) \*  
 dinvgamma(xt[i-1],shape = a,scale = b) /  
 dinvgamma(y[i],shape = a,scale = b)  
 if (u[i] <= r){  
 xt[i] <- y[i]   
 }else   
 {  
 xt[i] <- xt[i-1]  
 }  
}

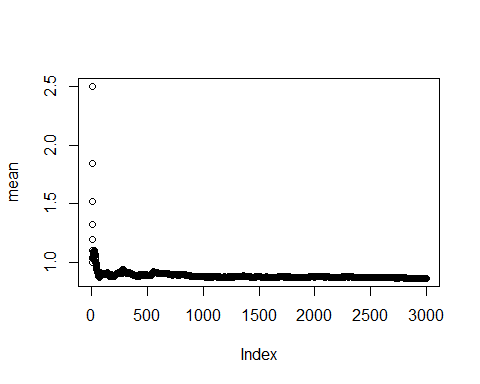
patient2 <- mcmc(xt)  
traceplot(patient2)



densplot(patient2)



meanplot <- function(patient)  
{  
mean <- patient%>% cumsum()  
mean <- mean/1:length(patient)  
plot(mean)  
}  
meanplot(patient2)



heidel.diag(patient2)

##   
## Stationarity start p-value  
## test iteration   
## var1 passed 1 0.0889   
##   
## Halfwidth Mean Halfwidth  
## test   
## var1 passed 0.862 0.0245

raftery.diag(patient2)

##   
## Quantile (q) = 0.025  
## Accuracy (r) = +/- 0.005  
## Probability (s) = 0.95   
##   
## You need a sample size of at least 3746 with these values of q, r and s

autocorr.plot(patient2)

