Group 29: Markov Chain Monte Carlo

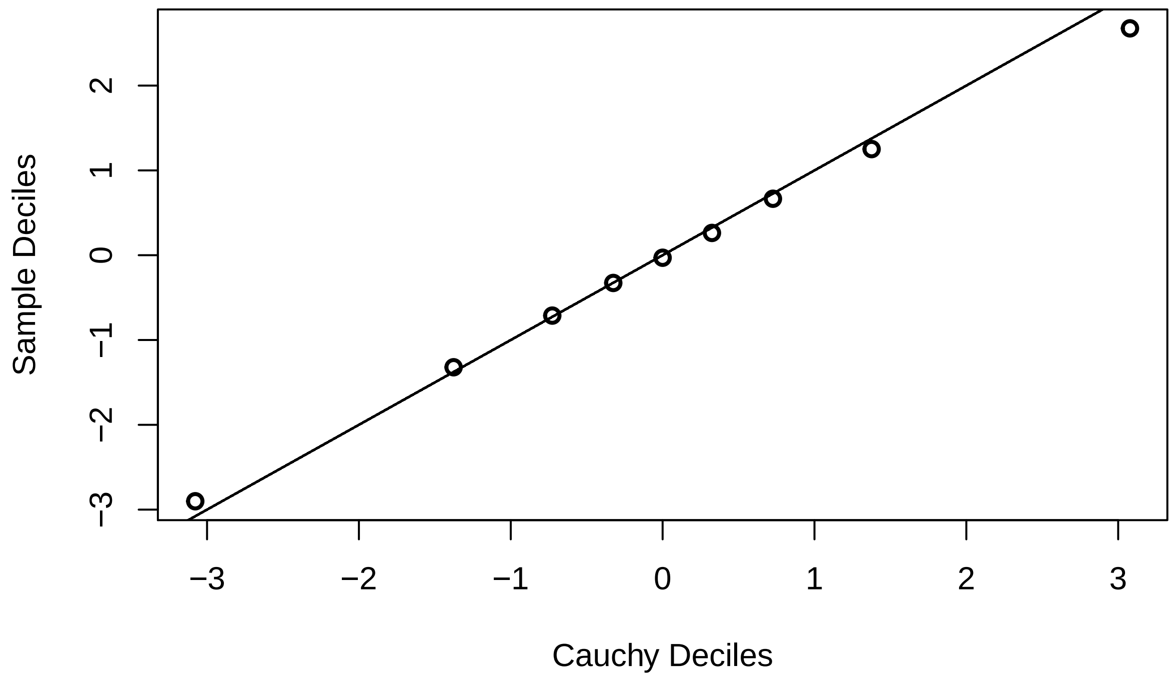
Nikhit Rao (nmrao2)

12/13/2019

Below is code from lecture/homework for Markov Chain Monte Carlo.

```
f <- function(x){1/pi/(1 + x^2)}          #The standard cauchy density
m2 <- 10000
b <- 1001                                 #The burn in period
x <- numeric(m2)
x[1] <- rnorm(1, 0, 2)                    #Starting from a normal distribution with mean 0
k2 <- 0
u <- runif(m2)
for (i in 2:m2) {
  xt <- x[i-1]
  y <- rnorm(1, xt, 2)                    #Using normal as the proposal distribution
  num <- f(y) * dnorm(xt, y, 2)
  den <- f(xt) * dnorm(y, xt, 2)
  if (u[i] <= num/den) x[i] <- y else {
    x[i] <- xt
    k2 <- k2+1                            #y is rejected
  }
}
index <- b:m2
y <- x[index]
a <- seq(0.1,0.9,0.1)                     #Introducing the decile probabilities
QR <- qcauchy(a)                          #deciles of standard cauchy
Q <- quantile(y, a)                       #deciles of the generated sample

qqplot(QR, Q, main="", xlab="Cauchy Deciles", ylab="Sample Deciles", lwd=2)
lines(x,x,xlim=c(-3.5,3.5), ylim=c(-3.5,3.5))
```



Below is code using SimDesign for Markov Chain Monte Carlo.

```
library(SimDesign)

f <- function(x){1/pi/(1 + x^2)}

Design <- data.frame(N = c(10000))

Generate <- function(condition, fixed_objects=FALSE) {
  dat <- numeric(condition$N)
  dat[1] <- rnorm(1, 0, 2)

  for (i in 2:condition$N) {
    u <- runif(1)
    xt <- dat[i-1]
    y <- rnorm(1, xt, 2)
    num <- f(y) * dnorm(xt, y, 2)
    den <- f(xt) * dnorm(y, xt, 2)
    if (u <= num/den) dat[i] <- y else {
      dat[i] <- xt
    }
  }

  dat
}

Analyse <- function(condition, dat, fixed_objects=NULL) {
```

```r
  a <- seq(0.1,0.9,0.1)
  Q <- quantile(dat[ceiling(condition$N*.1):condition$N], a)
  ret <- Q
  ret
}

Summarise <- function(condition, results, fixed_objects=NULL) {
  ret <- c(mcmc=results)
  ret
}

final <- runSimulation(design=Design, replications=1, generate=Generate, analyse=Analyse, summarise=Summ
```

```
##
##
Design row: 1/1;   Started: Fri Dec 13 06:23:53 2019;   Total elapsed time: 0.00s
##
## Simulation complete. Total execution time: 0.64s
```

```r
y <- final[1, 2:10]
y = c(y$mcmc1, y$mcmc1, y$mcmc2, y$mcmc3, y$mcmc4, y$mcmc5, y$mcmc6, y$mcmc7, y$mcmc8, y$mcmc9)
a <- seq(0.1,0.9,0.1)                          #Introducing the decile probabilities
QR <- qcauchy(a)                               #deciles of standard cauchy

qqplot(QR, y, main="", xlab="Cauchy Deciles", ylab="Sample Deciles", lwd=2)
lines(final,final,xlim=c(-3.5,3.5), ylim=c(-3.5,3.5))
```