Computational Statistics

Lab 4

Emil K Svensson and Rasmus Holm

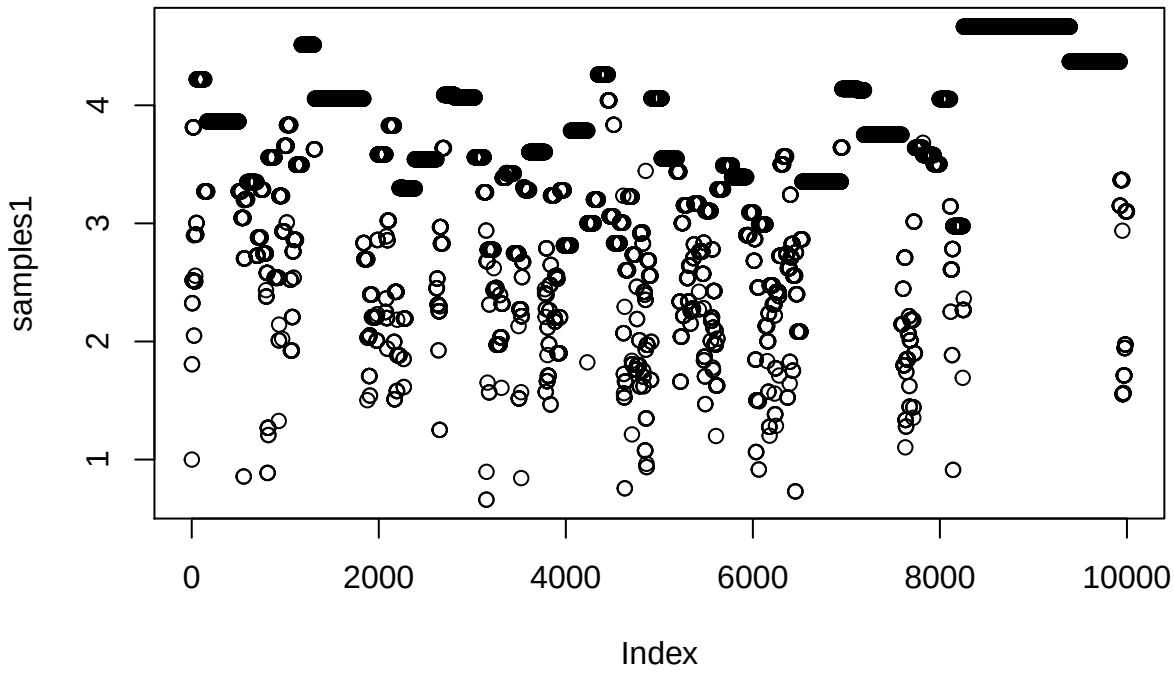
2017-02-15

```
pifunc <- function(x) {
    x^5 * exp(-x)
}

lognormalfuncs <- list(propsample=function(x) { rlnorm(1, meanlog=x, sdlog=1) },
                       propdensity=function(x, y) { dlnorm(x, meanlog=y, sdlog=1) },
                       targetdensity=pifunc)

chisquarefuncs <- list(propsample=function(x) { rchisq(1, df=floor(x + 1)) },
                       propdensity=function(x, y) { dchisq(x, df=floor(y + 1)) },
                       targetdensity=pifunc)

metropolis_hastings <- function(X0, iters, funcs) {
    x <- X0
    values <- rep(0, iters)

    alpha <- function(x, y) {
        numerator <- funcs$targetdensity(y) * funcs$propdensity(x, y)
        denominator <- funcs$targetdensity(x) * funcs$propdensity(y, x)
        numerator / denominator
    }

    for (i in 1:iters) {
        y <- funcs$propsample(x)
        u <- runif(1)

        if (u < alpha(x, y)) {
            x = y
        }

        values[i] <- x
    }

    values
}

iters <- 10000
X0 <- 1

x <- 0:20
y <- sapply(x, function(x) pifunc(x))
```

```
set.seed(123456)
samples1 <- metropolis_hastings(X0=X0, iters=iters, funcs=lognormalfuncs)
mean(samples1)
```

```
## [1] 3.503922
```

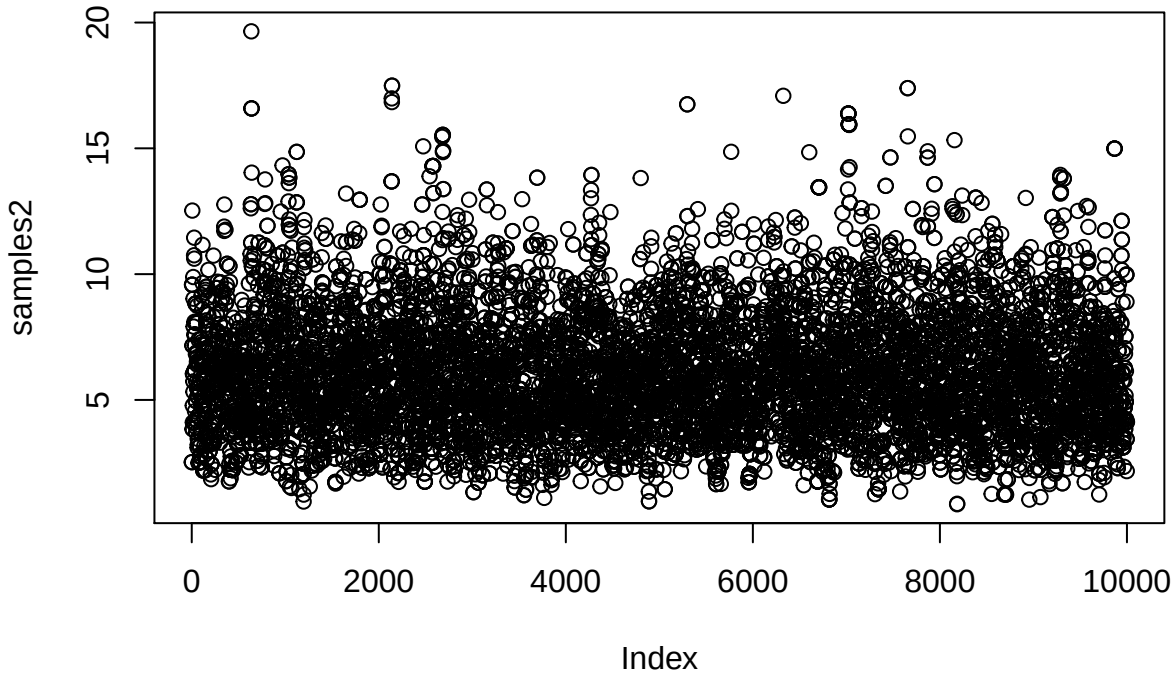
```
plot(samples1)
```



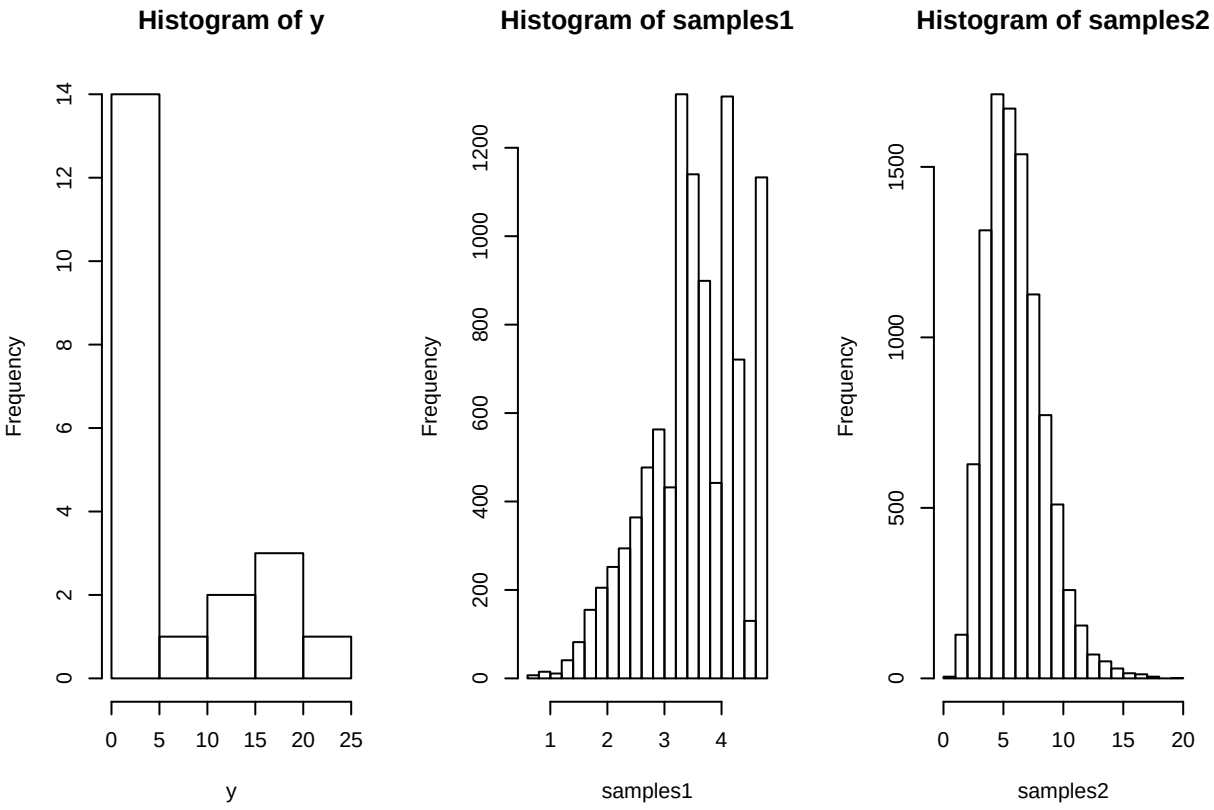

```
set.seed(123456)
samples2 <- metropolis_hastings(X0=X0, iters=iters, funcs=chisquarefuncs)
mean(samples2)
```

```
## [1] 6.015022
```

```
plot(samples2)
```




```r
x <- 0:20
y <- sapply(x, function(x) pifunc(x))

oldpar <- par(mfrow = c(1, 3))

hist(y)
hist(samples1)
hist(samples2)
```


Histogram of y
Frequency
0
2
4
6
8
10
12
14
0
5
10
15
20
25
y
Histogram of samples1
Frequency
0
200
400
600
800
1000
1200
1
2
3
4
samples1
Histogram of samples2
Frequency
0
500
1000
1500
0
5
10
15
20
samples2