TDDE07 - Lab 3

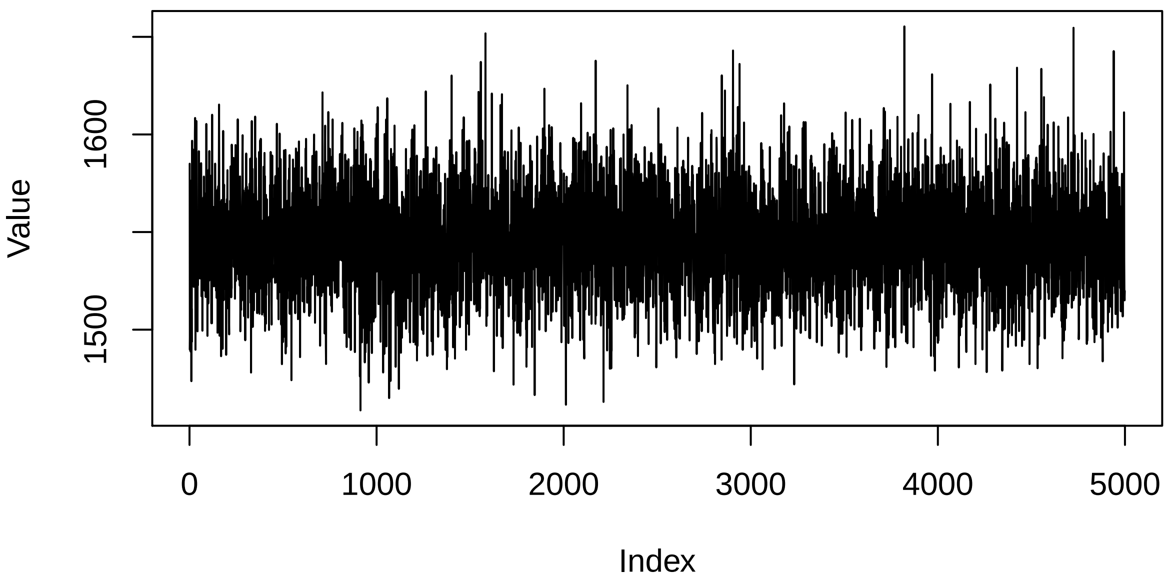
Sebastian Callh, Jacob Lundberg

4 maj 2018

1. Normal model, mixture of normal model with semi-conjugate prior

a) - Normal model

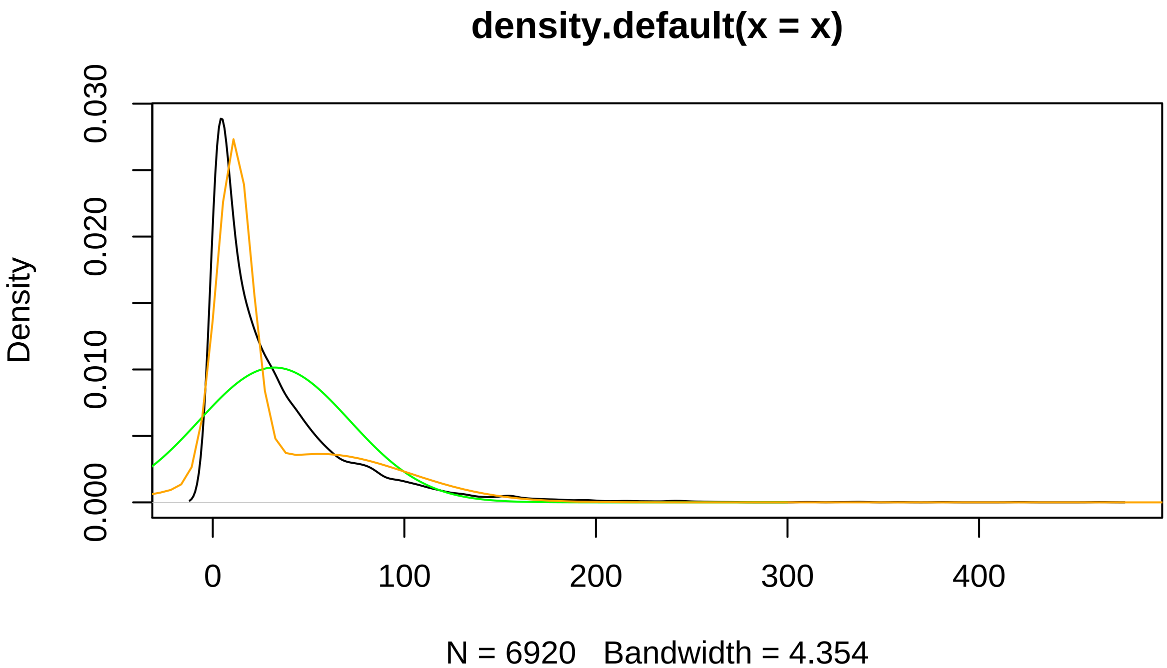
(i) - Gibbs sampler



(ii) - Analysis

b) Mixture normal model

c) Graphical comparison



2. Time-series models in Stan

a)

b)

c)

d)

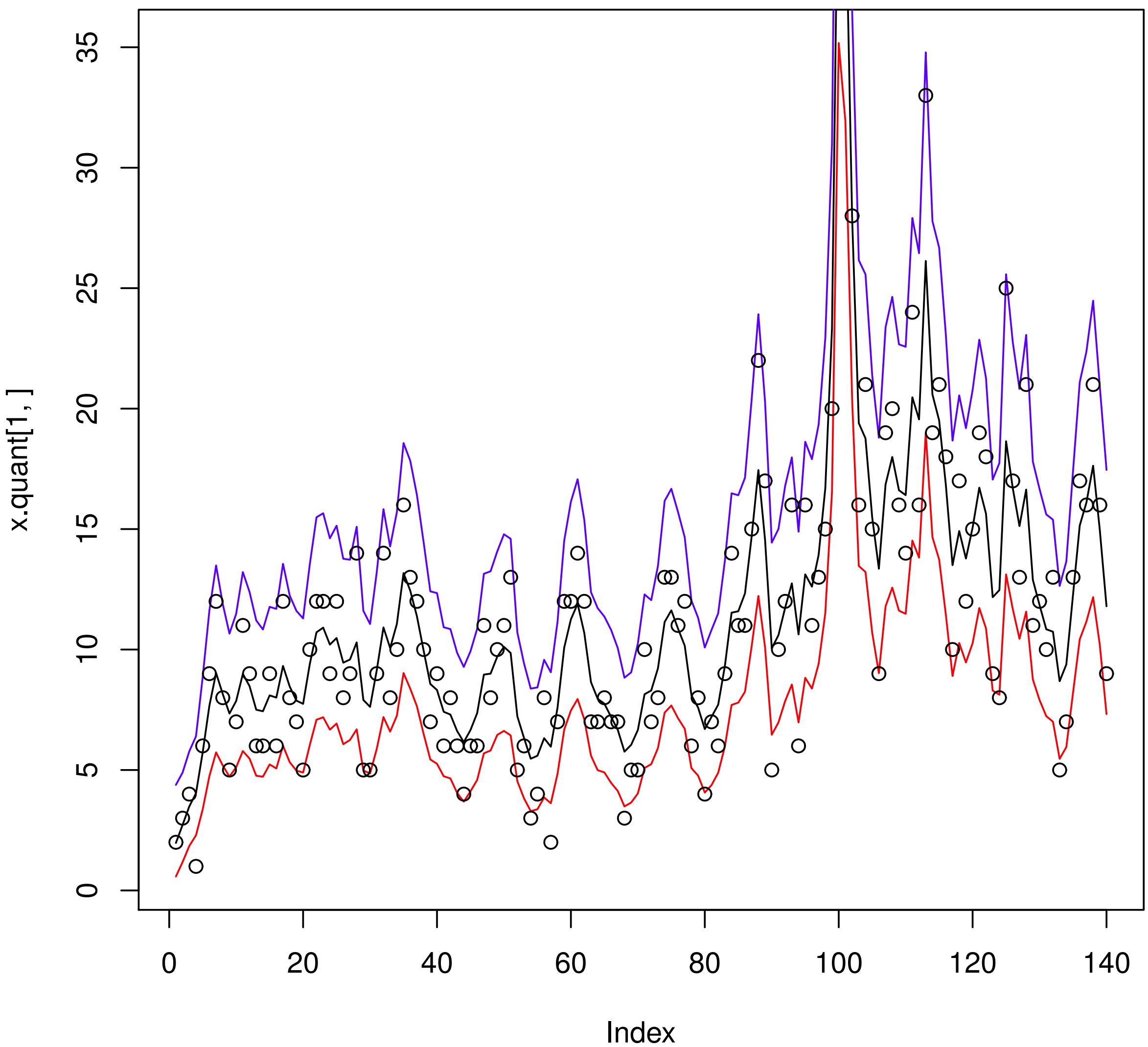


Figure 1: “Posterior without sigma prior”

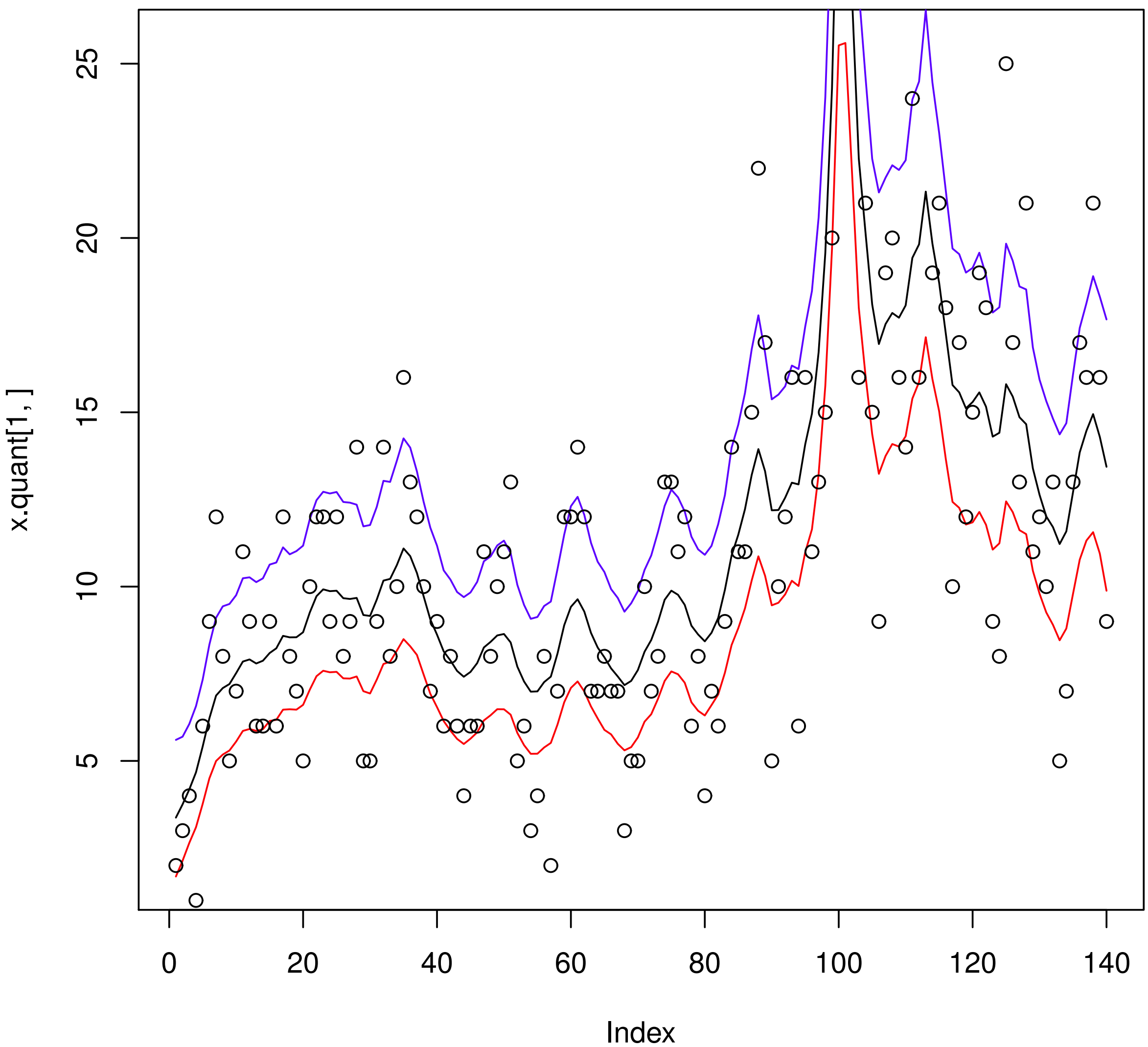


Figure 2: "Posterior with sigma 0.02 prior"

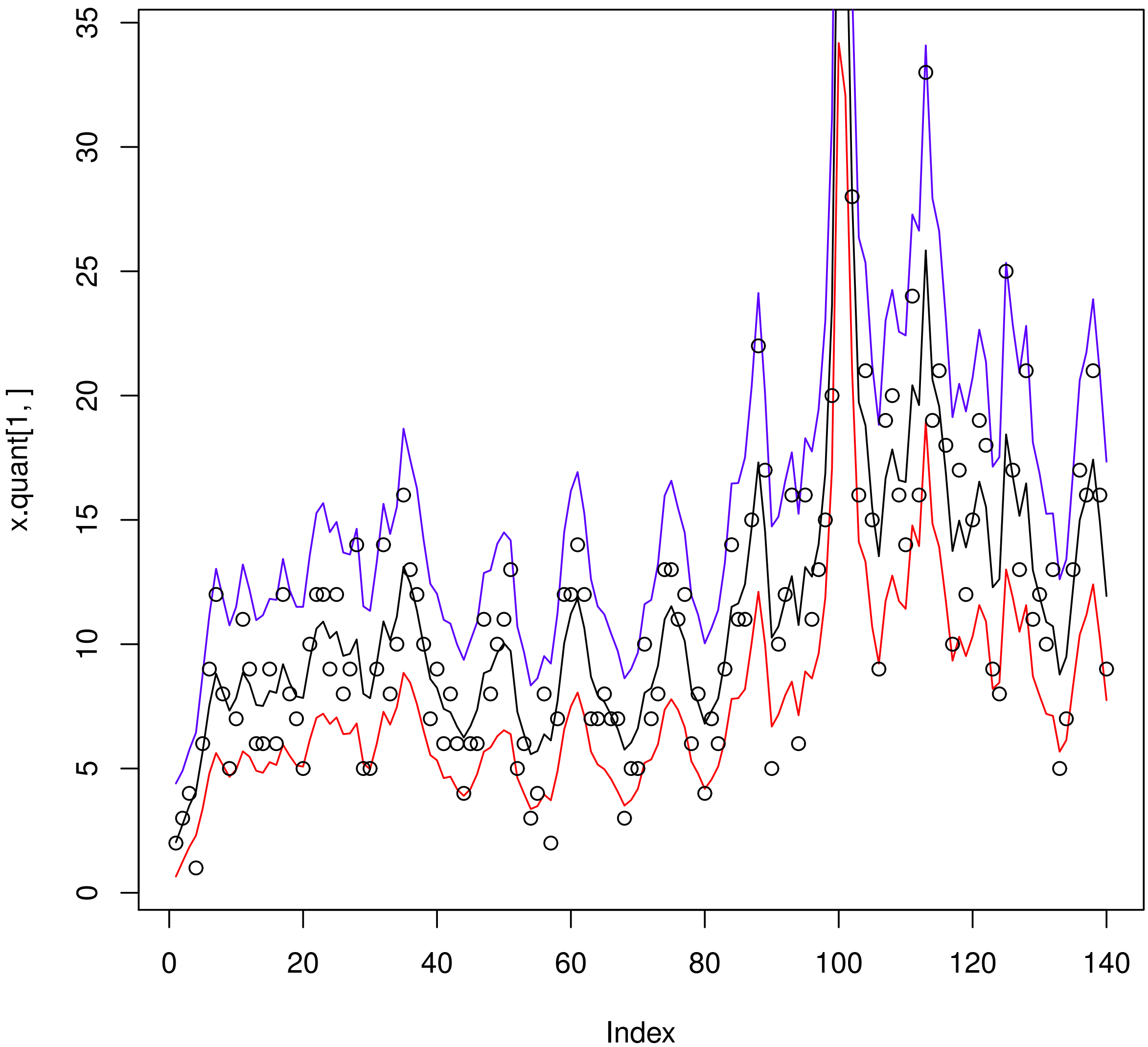


Figure 3: “Posterior with sigma prior”

Appendix

```
# 1 Normal model, mixture of normal model with semi-conjugate prior

# a) Normal model
data <- read.table("rainfall.dat", header=FALSE)
mu0    <- 20 # 0.5 cm
tau0   <- 5  # Variance of expected rainfall
nu0    <- 3  # Variance of expected value of variance
sigma0 <- 5  # Expected value of variance

nIter  <- 5000
theta0 <- c(20, 2)
x      <- data[,1]
n      <- length(x)

rinvchisq <- function(vx, sigmax, draws) {
  vx*sigmax/rchisq(draws, vx)
}

Mu.Conditional.Posterior.Draw <- function(sigma2) {
  taun <- sqrt(1/(n/sigma2 + 1/tau0^2))
  w    <- (n/sigma2) / (n/sigma2 + 1/tau0^2)
  mun  <- w*mean(x) + (1 - w) * mu0
  rnorm(1, mun, taun)
}

Sigma.Conditional.Posterior.Draw <- function(mu) {
  nun    <- n + nu0
  sigman <- (nu0*sigma0^2 + sum((x-mu)^2)) / nun
  rinvchisq(nun, sigman, 1)
}

Gibbs <- function(theta_t) {
  mu     <- Mu.Conditional.Posterior.Draw(theta_t[2])
  sigma2 <- Sigma.Conditional.Posterior.Draw(mu)
  c(mu, sigma2)
}

thetas     <- matrix(rep(0, nIter*2), nrow = nIter)
thetas[1,] <- theta0

for(i in 2:nIter) {
  thetas[i,] <- Gibbs(thetas[i-1,])
}

plot.normal.approximation <- function () {
  plot(thetas[-1,2],
       type = 'l',
       ylab = 'Value')
}

# b) Mixture normal model
```

```
# Model options
nComp <- 2                  # Number of mixture components

# Prior options
alpha     <- 10*rep(1,nComp)   # Dirichlet(alpha)
muPrior   <- rep(mu0,nComp)    # Prior mean of mu
tau2Prior <- rep(tau0,nComp)   # Prior std of mu
sigma2_0  <- rep(sigma0,nComp) # s20 (best guess of sigma2)
nu0       <- rep(nu0,nComp)    # degrees of freedom for prior on sigma2

#source("NormalMixtureGibbs.R")

# c) Graphical comparison

gibbs.mu    <- mean(thetas[,1])
gibbs.sigma <- sqrt(mean(thetas[,2]))
delta <- 0.05
grid  <- seq(-100, 300, delta)

plot.graphical.comparison <- function () {
  plot(density(x), col = 'black')
  lines(grid, dnorm(grid, gibbs.mu, gibbs.sigma), type = 'l', col = 'green')
  lines(xGrid, mixDens, type = 'l', col = "orange")
}

# 2 Time series models in Stan
library(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)

# a)

ar.process <- function (phi, init, sigma, T) {
    y     <- rep (0, T)
    y[1]  <- init
    for (t in 2:T) {
        y[t] <- mu + phi*(y[t-1] - mu) + rnorm(1, 0, sigma)
    }
    y
}

T     <- 200
mu    <- 10
phi1  <- 0.3
phi2  <- 0.95
s <- sqrt(2)
x <- ar.process(phi1, mu, s, T)
y <- ar.process(phi2, mu, s, T)

plot(x, type ='l')


# b)
```

```r
x.fit <- stan(file = "time-series.stan",
            data = list(
                x = x,
                N = T
            ))
y.fit <- stan(file = "time-series.stan",
            data = list(
                x = y,
                N = T
            ))

posterior.mean.x <- get_posterior_mean(x.fit)
posterior.mean.y <- get_posterior_mean(y.fit)

head(posterior.mean.y)
head(posterior.mean.x)
mu.x.post    <- posterior.mean.x[1, 5]
sigma.x.post <- posterior.mean.x[2, 5]
phi.x.post   <- posterior.mean.x[3, 5]
x.params <- extract(x.fit, pars = c("mu", "phi"))
y.params <- extract(y.fit, pars = c("mu", "phi"))
plot(x.params$mu, x.params$phi)
plot(y.params$mu, y.params$phi)

mu.y.post    <- posterior.mean.y[1, 5]
sigma.y.post <- posterior.mean.y[2, 5]
phi.y.post   <- posterior.mean.y[3, 5]

z.x <- ar.process(phi.x.post, mu.x.post, sigma.x.post, T)
z.y <- ar.process(phi.y.post, mu.y.post, sigma.y.post, T)

plot(y.fit)
summary(y.fit)
plot(x, type = 'l', col = 'red')
lines(z, col = 'blue')
fit
plot(fit)

d <- 0.05
grid <- seq(0, 2, d)
plot(grid, dnorm(grid, 0, 0.02))

# c)

campy.data <- as.vector(read.table("campy.dat", header = TRUE)[,1])
c.fit <- stan(file ="campy.stan",
              data = list (
                  c = campy.data,
                  N = length(campy.data)
              ))

params <- extract(c.fit, pars = c("mu", "sigma"))
x <- extract(c.fit, pars = "x")
```

```r
mean(params$sigma)
theta.t <- exp(x$x)
x.mean <- apply(theta.t, 2, mean)
x.quant <- apply(theta.t, 2, quantile, probs=c(0.025,0.975))
xb
setEPS()
postscript("posterior-with-sigma-0.02-prior.eps")
plot(x.quant[1,], type = 'l', col='red')
lines(x.quant[2,], type = 'l', col='blue')
lines(x.mean, type = 'l')
points(campy.data)
dev.off()

c.df <- as.data.frame(fit)
head(as.matrix(c.fit)[,1])

# d)

# posterior sigma = 0.2603307 with uniform
# posterior sigma = 0.2481805 with 0.15
# posterior sigma = 0.1074629 with 0.02
```