Supplementary Material: Bayesian semiparametric latent variable model with DP prior for joint analysis: Implementation with nimble

Zhihua Ma[†], and Guanghui Chen[†]
[†]Department of Statistics, Jinan University, Guangzhou, China

A. Traceplots of M_0 in Section 5

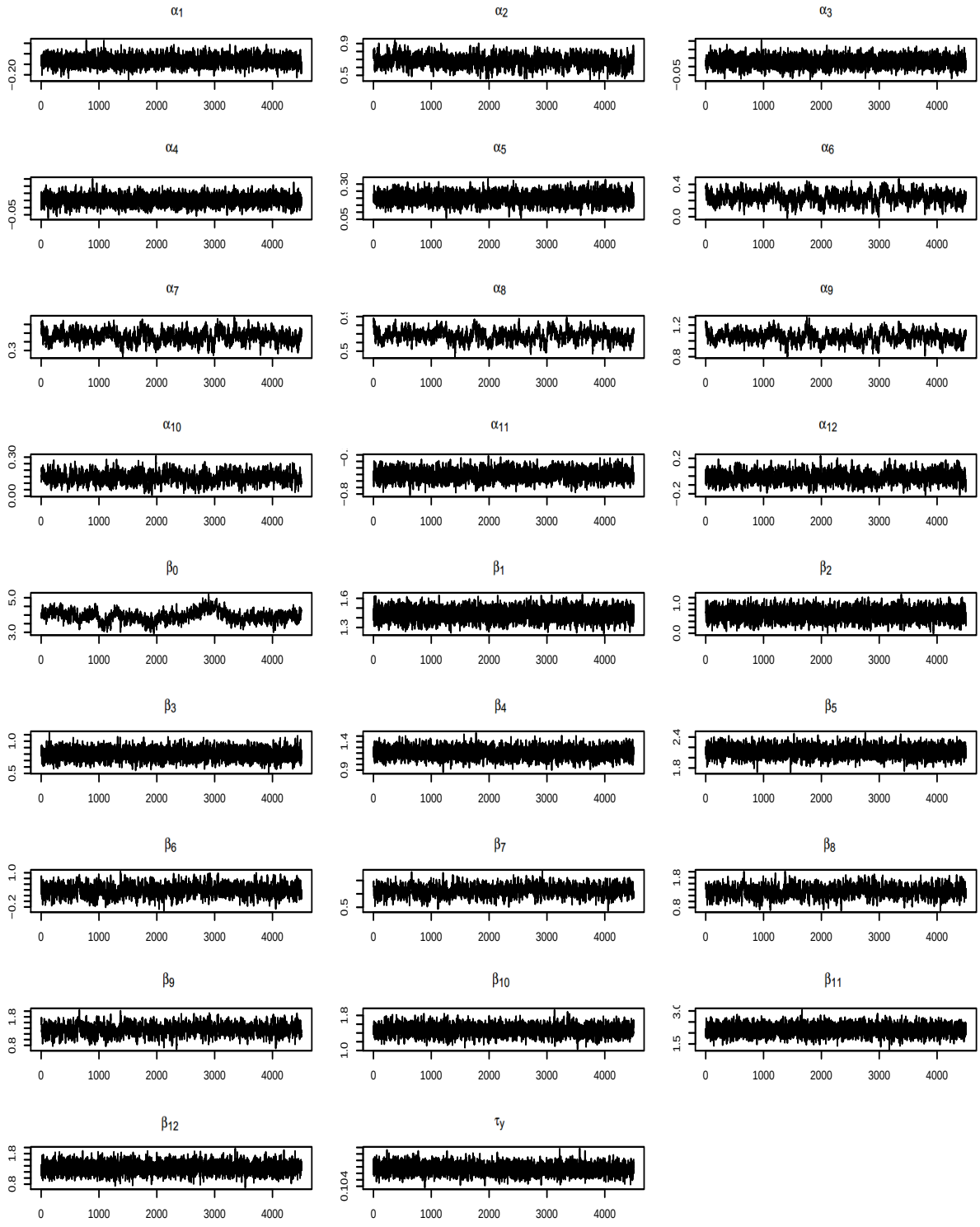

Figure 1: Traceplots of parameters of interest in model M_0 of CGSS 2013 data

B. ACF plots of M_0 in Section 5

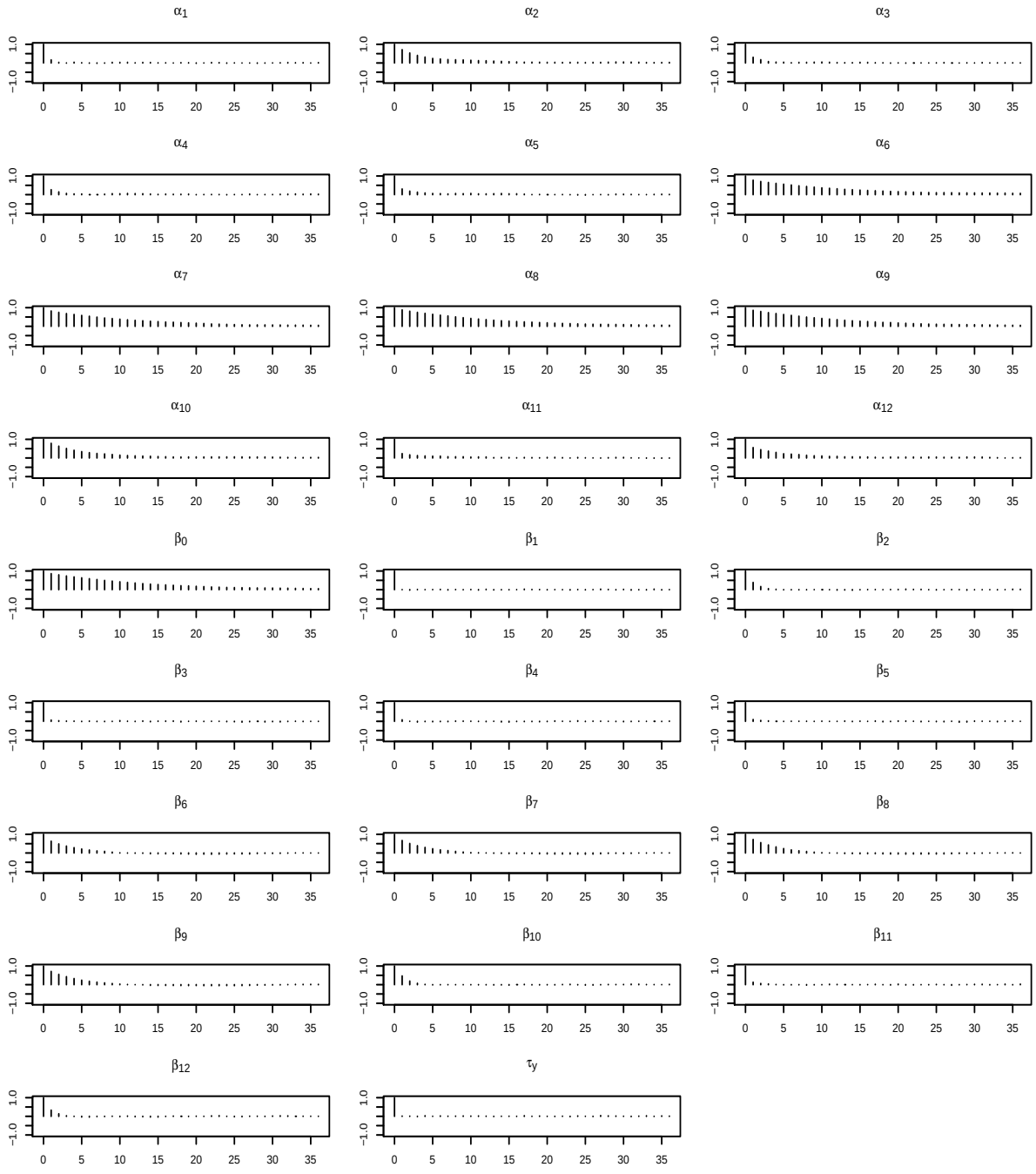

Figure 2: ACF plots of parameters of interest in model M_0 of CGSS 2013 data

C. nimble model code in Section 4.2

```
# sort function defined for nimble
Rsort <- nimbleRcall(
    function(x=double(1)){}, Rfun="sort",
    returnType = double(1), envir = .GlobalEnv
)

# nimble model code
cgssCode <- nimbleCode({
    for (i in 1:N){
      # Continuous response
      Y[i] ~ dnorm(mu_y[i], tau = tau_y)
      mu_y[i] <- b[1] + b[2] * X1[i] + b[3] * X2[i] + L[i]

      # Ordinal response
      Z[i] ~ dcat(p[i, 1:C])
      p[i,1] <- phi(theta[1] - mu_z[i])
      for (j in 2 : (C-1)){
        p[i, j] <- phi(theta[j] - mu_z[i]) - phi(theta[j-1] -
           mu_z[i])
      }
      p[i, C] <- 1-phi(theta[M-1] - mu_z[i])
      mu_z[i] <- a[1] * X1[i] + a[2] * X2[i] + L[i]

      # latent variable
      L[i] <- L1[latent[i]]
      latent[i] ~ dcat(prob[1 : J])
    }
    # stick-breaking representation
    pi[1] <- V[1]
    for (j in 2 : J){
      pi[j] <- V[j] * (1 - V[j-1]) * pi[j-1] / V[j-1]
    }
    pi_sum <- sum(pi[1 : J])
    for (j in 1 : J){
      L1[j] ~ dnorm(0, 1)
      V[j] ~ dbeta(1, kappa)
      prob[j] <- pi[j] / pi_sum
    }
    kappa ~ dgamma(1,1)

    for (j in 1 : (C-1)){
      theta_raw[j] ~ dnorm(0, tau = 0.001)
```

```
    }
    theta[1 : (C-1)] <- Rsort(theta_raw[1 : (C-1)])

    # priors for other parameters
    for (j in 1:3){
      b[j] ~ dnorm(0, tau = 0.001)
    }
    for (j in 1:2){
      a[j] ~ dnorm(0, tau = 0.001)
    }
    tau_y1 ~ dgamma(0.001, 0.001)
})
```