Bayesova statistika

Domača naloga 3

Sara Bizjak | 27202020

December 2021

1. naloga

Koda Gibbsovega vzorčevalnika. Kjer je kaj dodano ali spremenjeno iz kode iz vaj, je zraven komentar `DODANO` oz. `SPREMENJENO`. Bolj pregledna koda (s klici za vse naslednje naloge) je dostopna v datoteki `DN3_koda_sarabizjak.R`.

```
# Uvoz podatkov:
setwd("~/Documents/ISRM magistrski/Bayesova statistika/BS/Domace naloge/DN3")
source("podatki_sole.R")
str(pod)

library(ggplot2)
ggplot(pod, aes(x = school, y = mathscore, group = school)) +
        stat_summary(fun.ymin = min, fun.ymax = max, fun.y = mean) +
        labs(title = "Povprecja (pika) in razpon rezultatov po solah")

# Preureditev podatkov:
library(dplyr)
pod.sole = pod %>%
    group_by(school) %>%
    summarise(povprecje = mean(mathscore),
                          n=length(mathscore),
                          varianca = var(mathscore))

# Nastavitev parametrov:

# Parametri (hiper)apriornih porazdelitev, isti kot na vajah:
sigma20 = 100
nu0 = 1
eta20 = 100
kappa0 = 1
```

```
mu0 = 50
tau20 = 25

# Parametri iz domace naloge:
a = 2
b = 1 / 10
alpha = 2
k.max = 1000

### Pripravimo si kolicine, ki jih bomo potrebovali iz podatkov
x = pod
m = length(pod.sole$school)
n = pod.sole$n
x.povpr = pod.sole$povprecje
x.var = pod.sole$varianca

### Dolocimo si zacetne vrednosti
muGroups = x.povpr
mu = mean(muGroups)
eta2 = var(muGroups)

# DODANO: shranimo sigmo za vsako skupino loceno
sigma2Groups = x.var

### Pripravimo si prostor za shranjevanje
n.iter = 5000

muGroups.all = matrix(nrow = n.iter, ncol = m)
mu.all = rep(NA, n.iter)
eta2.all = rep(NA, n.iter)

# DODANO:
sigma2Groups.all = matrix(nrow=n.iter, ncol= m)
sigma20.all = rep(NA, n.iter)
nu0.all = rep(NA, n.iter)

### Na prvo mesto si shranimo zacetne vrednosti (nepotrebno)
muGroups.all[1, ] = muGroups
mu.all[1] = mu
eta2.all[1] = eta2
```

```
# DODANO: shranimo zacetne vrednosti novih parametrov
sigma2Groups.all[1,] = sigma2Groups
sigma20.all[1] = sigma20
nu0.all[1] = nu0

### Pozenemo Gibbsov vzorcevalnik

set.seed(1)
for (s in 1 : n.iter) {

    # Vzorcimo muGroups
    for (j in 1 : m) {
    # SPREMENJENO: sigma2 v enacbi je zamenjana s sigma2Groups[j],
    #              ostalo je enako
    muGroups[j] = rnorm(1,
                        mean = (x.povpr[j] * n[j] / sigma2Groups[j] + mu / eta2)
                        sd = sqrt(1 / (n[j] / sigma2Groups[j] + 1 / eta2)))
    }

    # DODANO: Vzorcimo sigma2Groups namesto sigma2, delamo po skupinah...
    #         po formuli iz domace naloge
    for(j in 1 : m){
    sigma2Groups[j] = 1 / rgamma(1,
                                 (nu0 + n[j]) / 2,
                                 (nu0 * sigma20 + sum((x[x[, 1] == j, 2] - mu
    }

    # DODANO: Vzorcimo sigma20... po formuli iz domace
    sigma20 = rgamma(1,
                     a + m * nu0 / 2,
                     b + nu0 * sum(1 / sigma2Groups) / 2)

    # Vzorcimo mu, ne spreminjamo nic
    mu <- rnorm(1,
                mean = (mean(muGroups) * m / eta2 + mu0 / tau20) / (m / eta2 + 1
                sd = sqrt(1 / (m / eta2 + 1 / tau20)))

    # Vzorcimo eta2, ne spreminjamo nic
    ss   <- kappa0 * eta20 + sum((muGroups - mu)^2)
    eta2 <- 1 / rgamma(1, (kappa0 + m) / 2, ss / 2)
```

```
    # DODANO: Vzorcimo nu0, koda kopirana iz navodil za domaco nalogo
    k <- 1:k.max
    logp.nu0 <- m * (0.5 * k * log(k*sigma20/2) - lgamma(k/2)) +
    (k/2-1) * sum(log(1/sigma2Groups)) +
    - k * (alpha + 0.5 * sigma20 * sum(1/sigma2Groups))
    nu0 <- sample(k, 1, prob = exp(logp.nu0 - max(logp.nu0)))

    # Shranimo nove parametre
    muGroups.all[s,] = muGroups
    mu.all[s] = mu
    eta2.all[s] = eta2

    # DODANO:
    sigma2Groups.all[s,] = sigma2Groups
    sigma20.all[s] = sigma20
    nu0.all[s] = nu0
}
```

---

2. naloga: Proučevanje konvergence

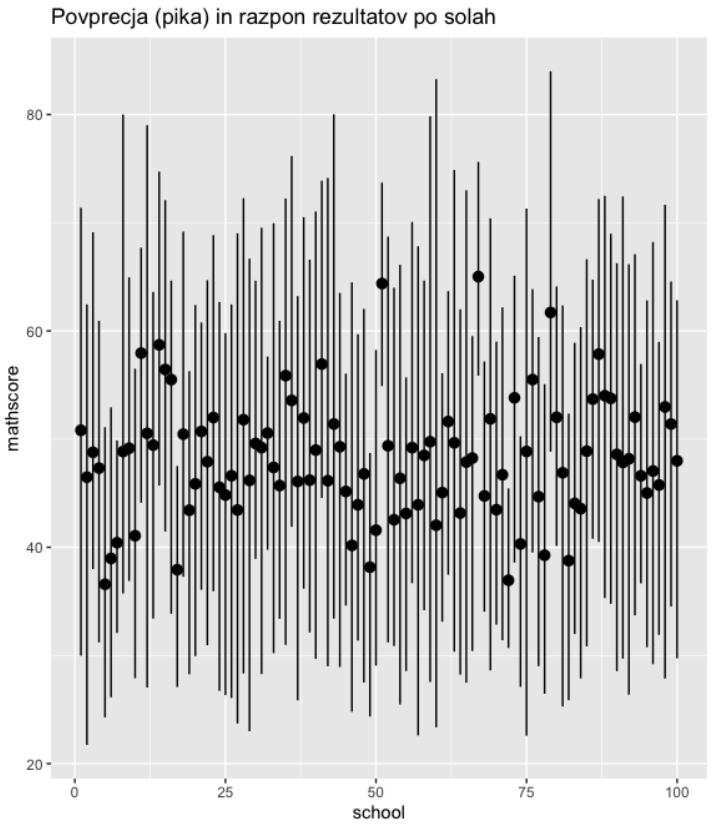

Figure 1: Podatki šol.

Trace plots.
Poglejmo si najprej trace plots za hiperparametre.

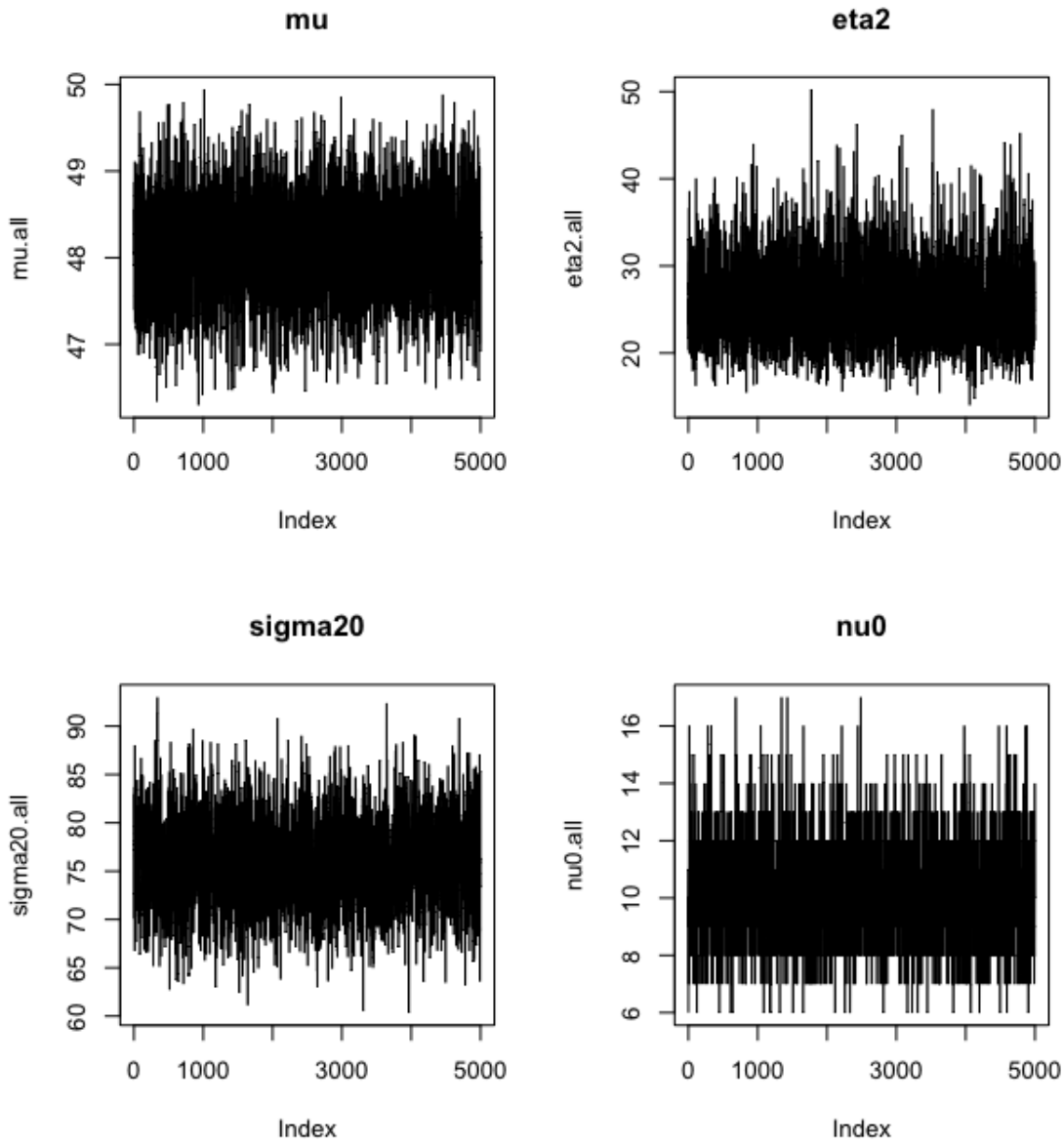

Figure 2: Trace plots za hiperparametre.

Videti je, da je kovergenca dosežena, zaporedje pa se že takoj giba v območju porazdelitve, tako da *burn-in* ni potreben. Rezultat je pričakovan, saj smo za začetno vrednost zaporedja izbrali vzorčno povprečje posamezne skupine (oz. vzorčno varianco), kar je najbolj možna smiselna začetna vrednost.

Poglejmo si še iste grafe z izrisanimi prvimi 500 členi zaporedij.

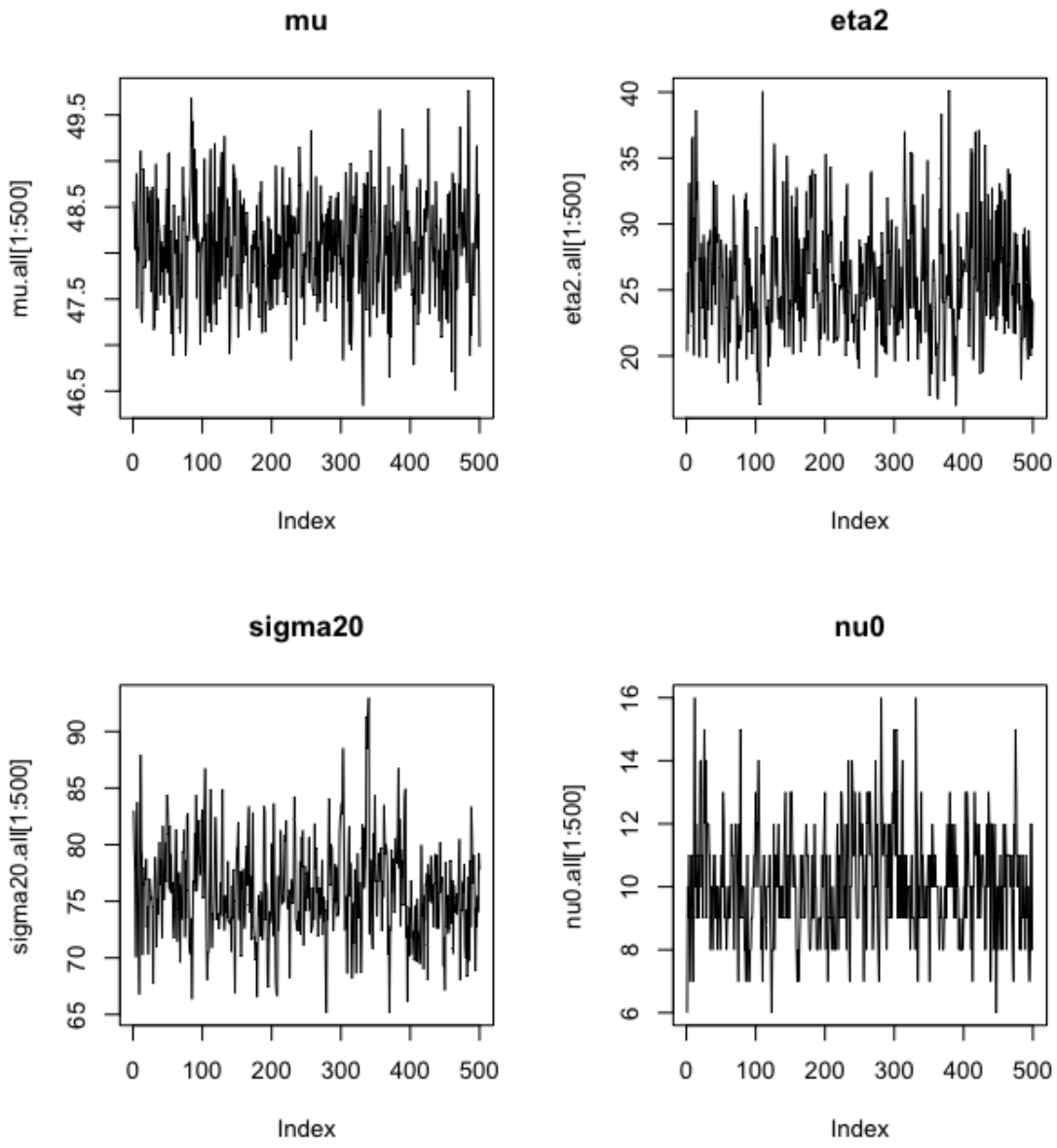

Figure 3: Trace plots za hiperparametre z izrisanimi prvimi 500 členi zaporedij.

Slednji grafi s 500 členi zaporedij samo še potrdijo prejšnjo trditev o *burn-in* parametru, katerega res ne potrebujemo. Iz grafa za $nu0$ se zazdi, da so vrednosti diskretne. Izpis dejanskih vrednsti domnevo potrdi.

Poglejmo še ostale parametre, (naključno) si izberemo šole $j = 1, 10, 50, 80$.

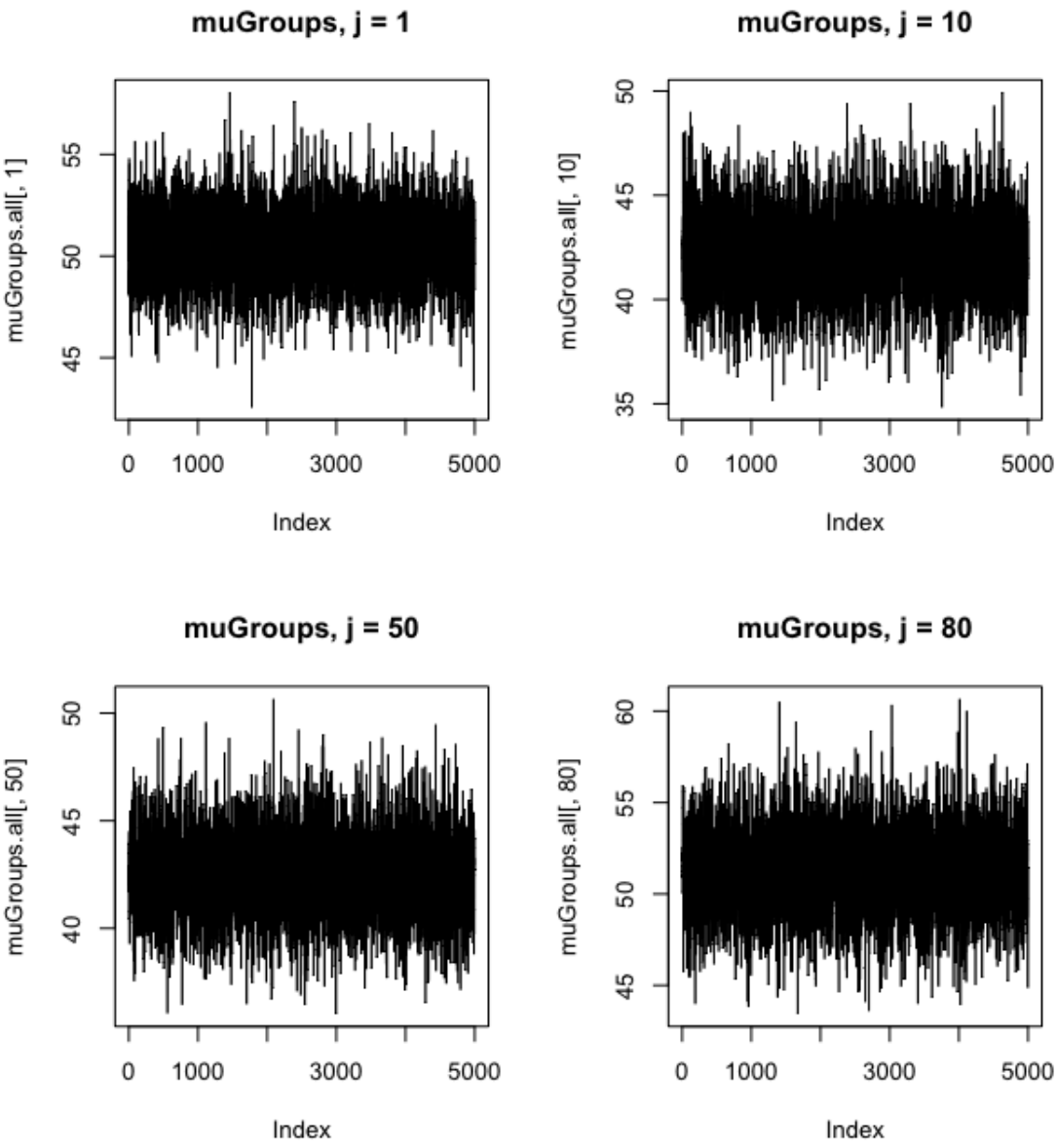

Figure 4: Trace plots za parameter `muGroups`.

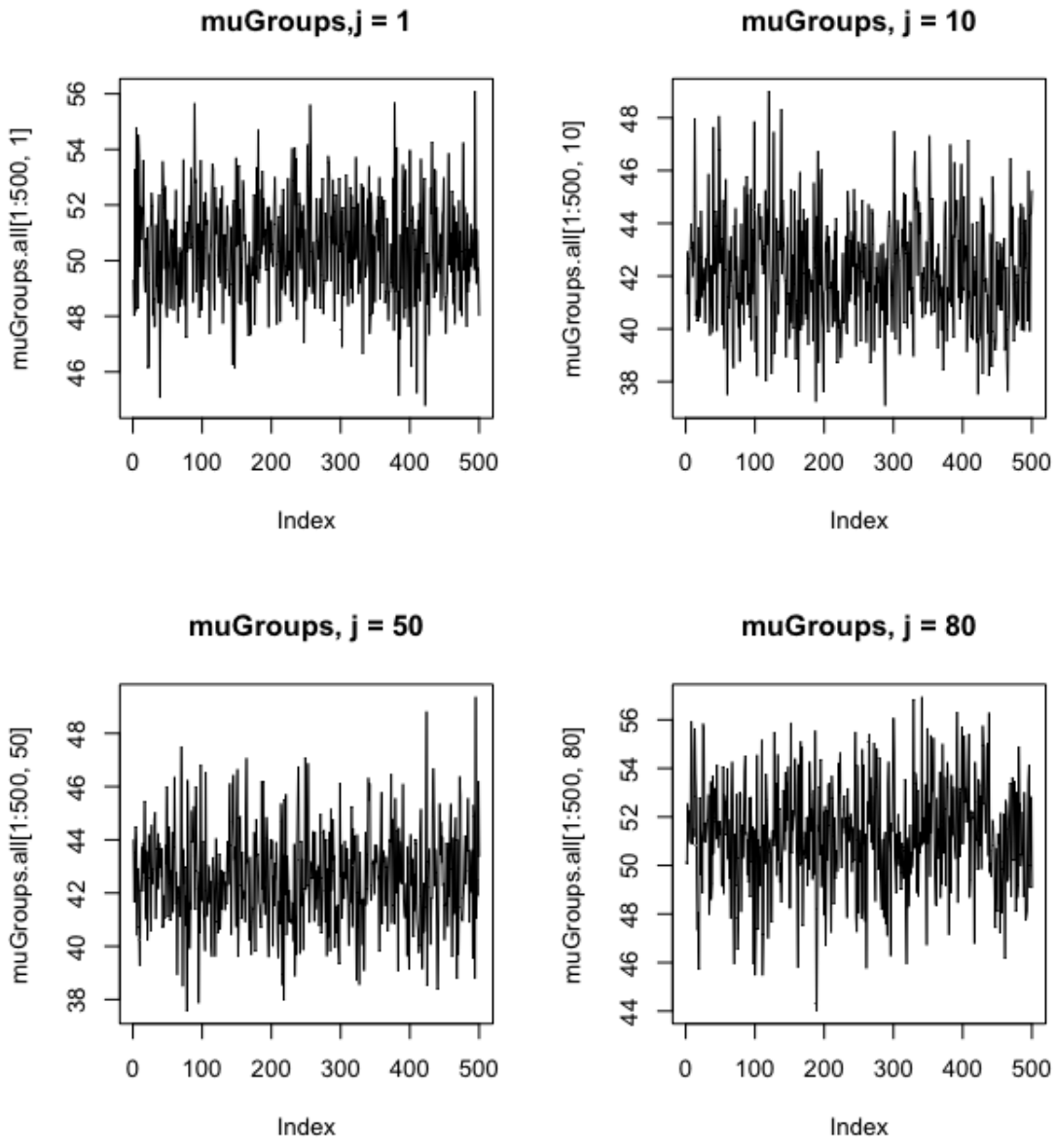

Figure 5: Trace plots za parameter `muGroups` z izrisanimi prvimi 500 členi zaporedja.

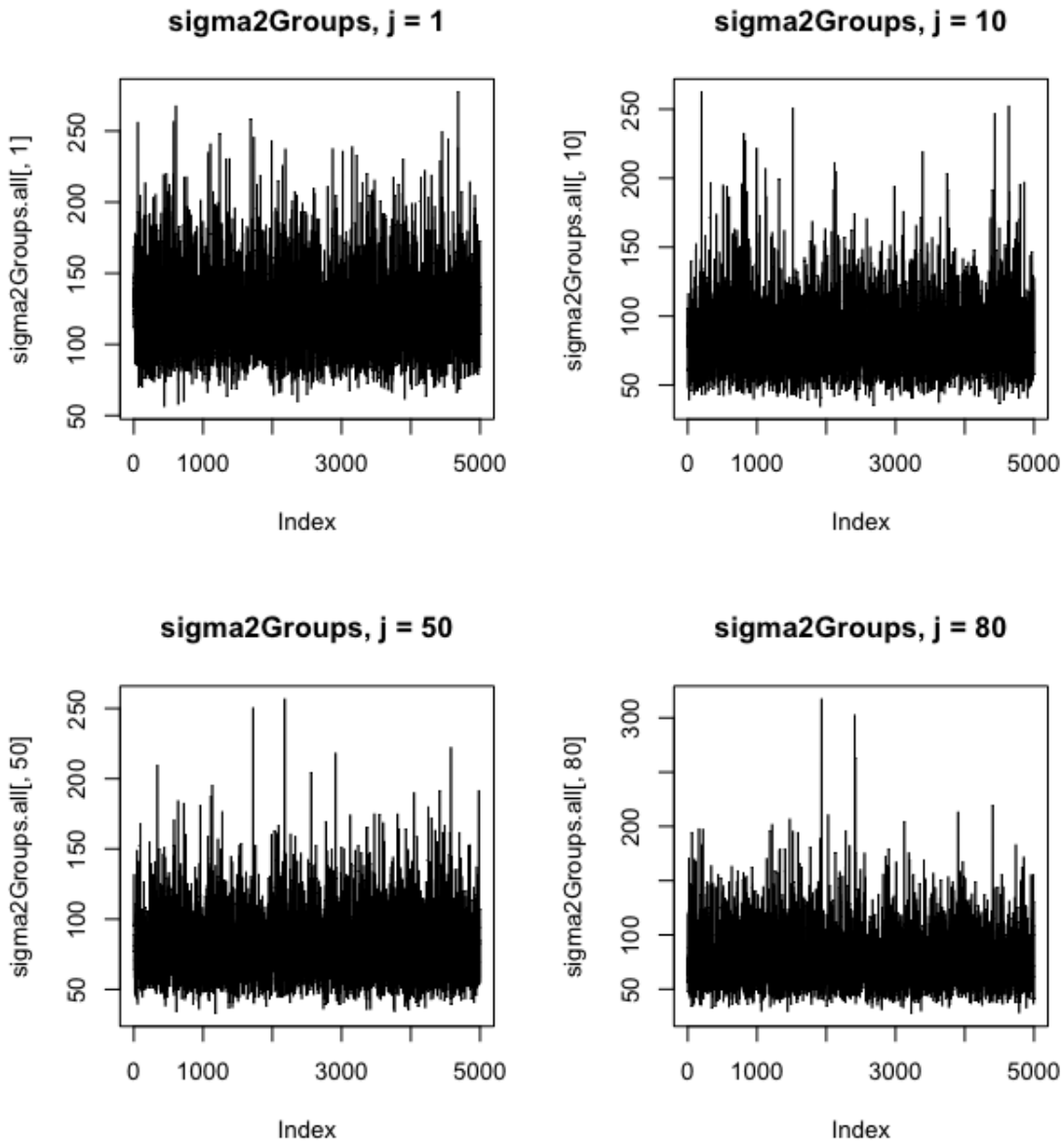

Figure 6: Trace plots za parameter `sigma2Groups`.

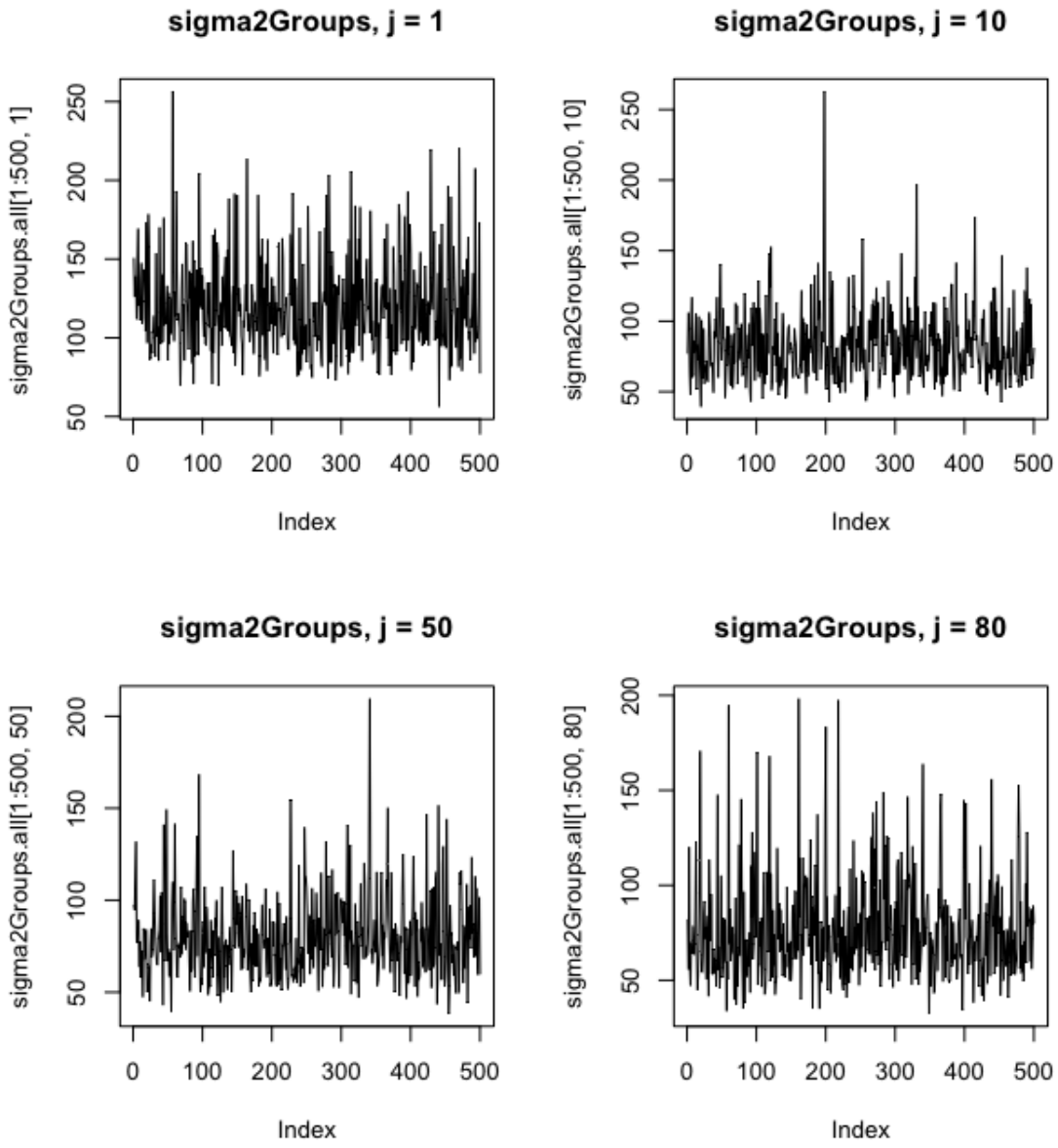

Figure 7: Trace plots za parameter `sigma2Groups` z izrisanimi prvimi 500 členi zaporedja.

Tudi pri teh parametrih, podobno kot pri hiper, ni videti *burn-in* dela, zato povsod ohranimo celotne verige.

Porazdelitev vzorcev.
Za opazovanje porazdelitve vzorcev verigo razdelimo na desetine, torej na 10 enakih delov. V nadaljevanju potem za vsak podvzorec posebej opazimo povprečja (oz. variance) z intervali zaupanja, ki se morajo, za stabilnost verig, znotraj posamezne skupine relativno ujemati – za stabilnost se torej znotraj posameznih podvzorcev povprečja *lahko* razlikujejo, skozi vse podvzorce posamezne skupine pa se morajo "gibati v istem rangu". Izrišimo sedaj grafe, da bo jasno, kaj smo s tem mislili.

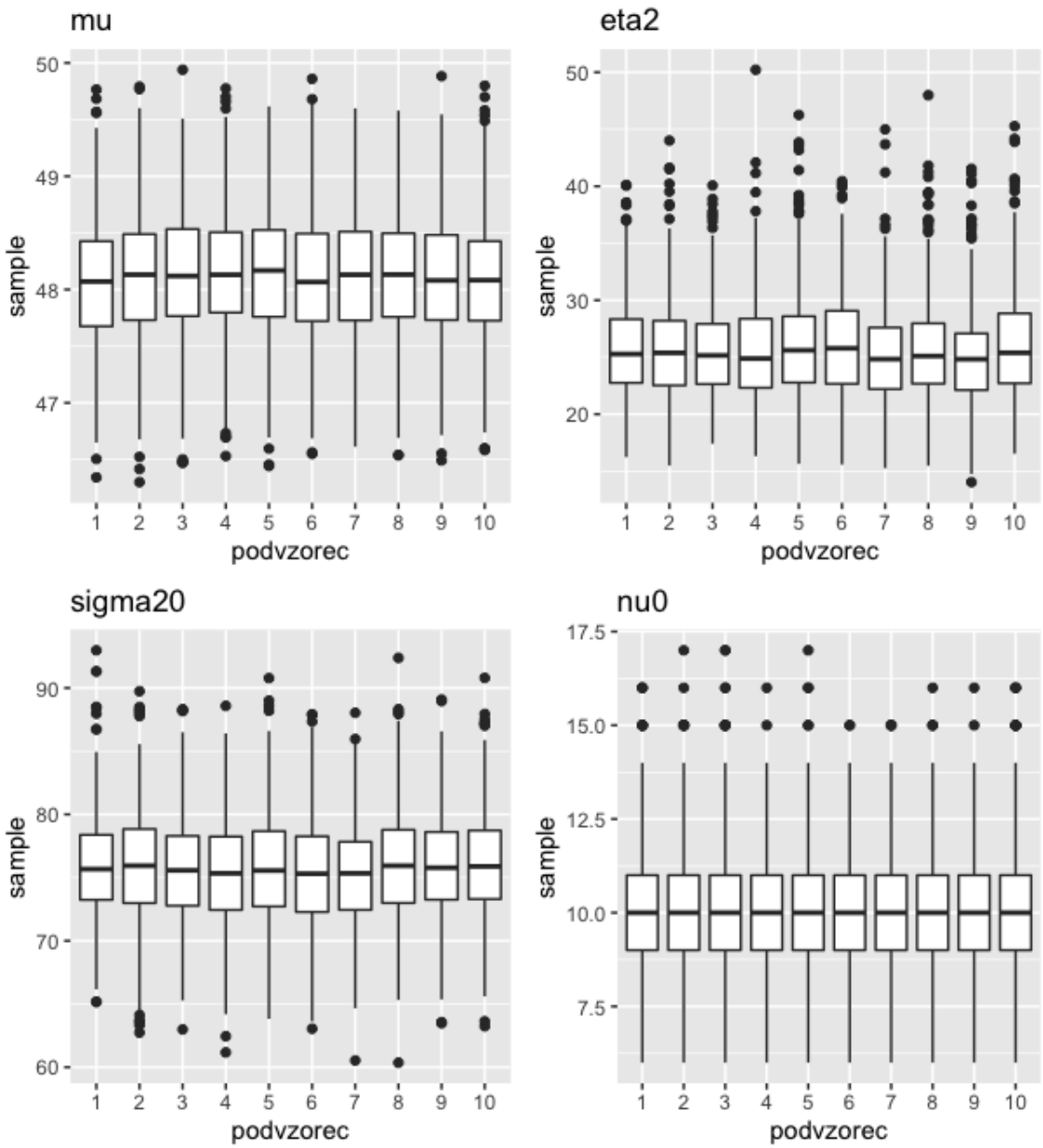

Figure 8: Porazdelitev podvzorcev za hiperparametre.

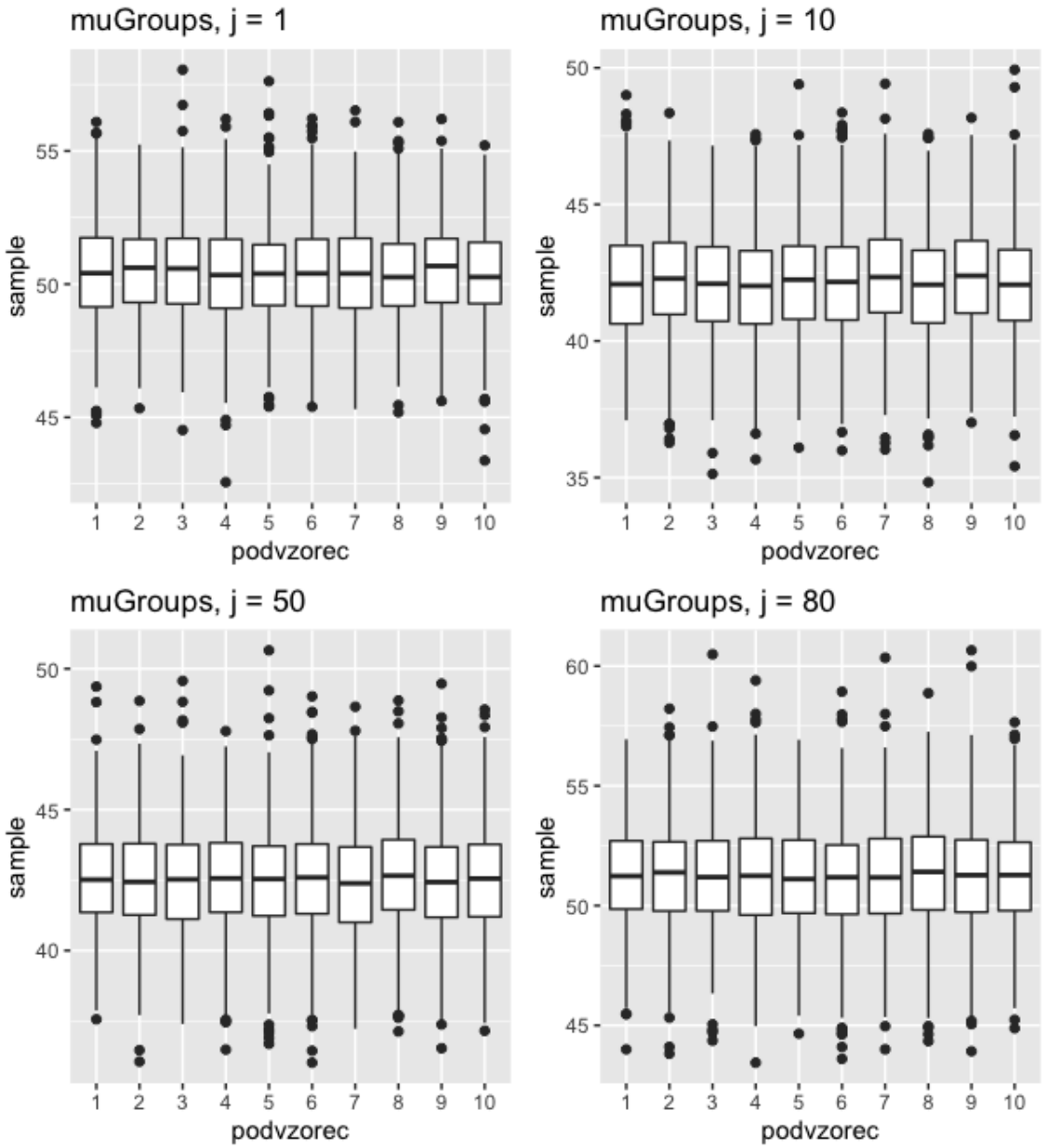

Figure 9: Porazdelitev podvzorcev za parameter `muGroups`.

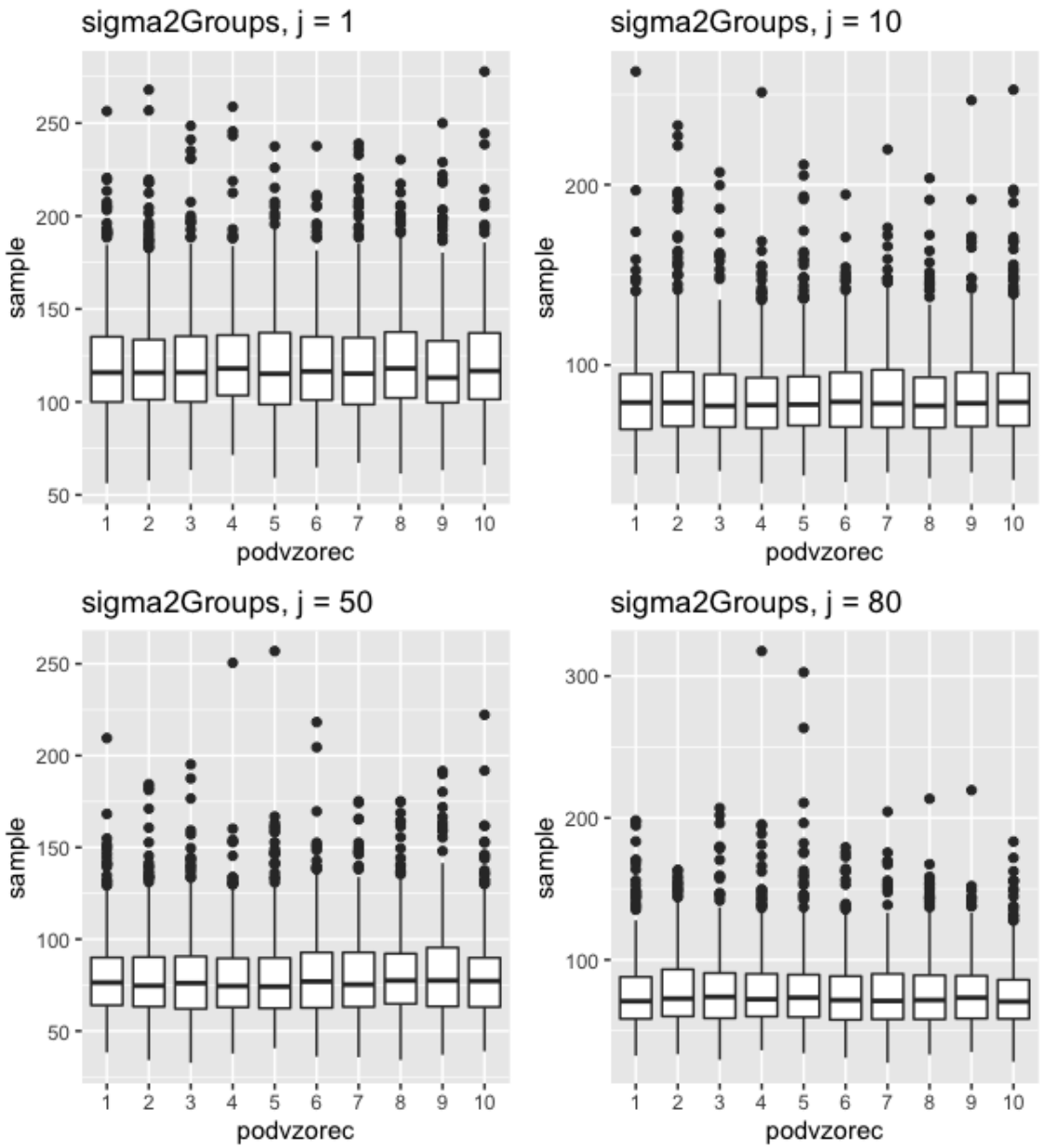

Figure 10: Porazdelitev podvzorcev za parameter `sigma2Groups`.

Iz grafov razberemo, da so verige stabilne. Generalno gledano so med posameznimi skupinami, tj. šolami, opazne manjše razlike, znotraj skupine pa so podvzorci stabilni. Imamo torej situacijo, kot smo jo opisali že zgoraj.

Avtokorelacije.

Poglejmo si še avtokorelacije. Najprej za hiperparametre in potem še za nekaj ostalih.

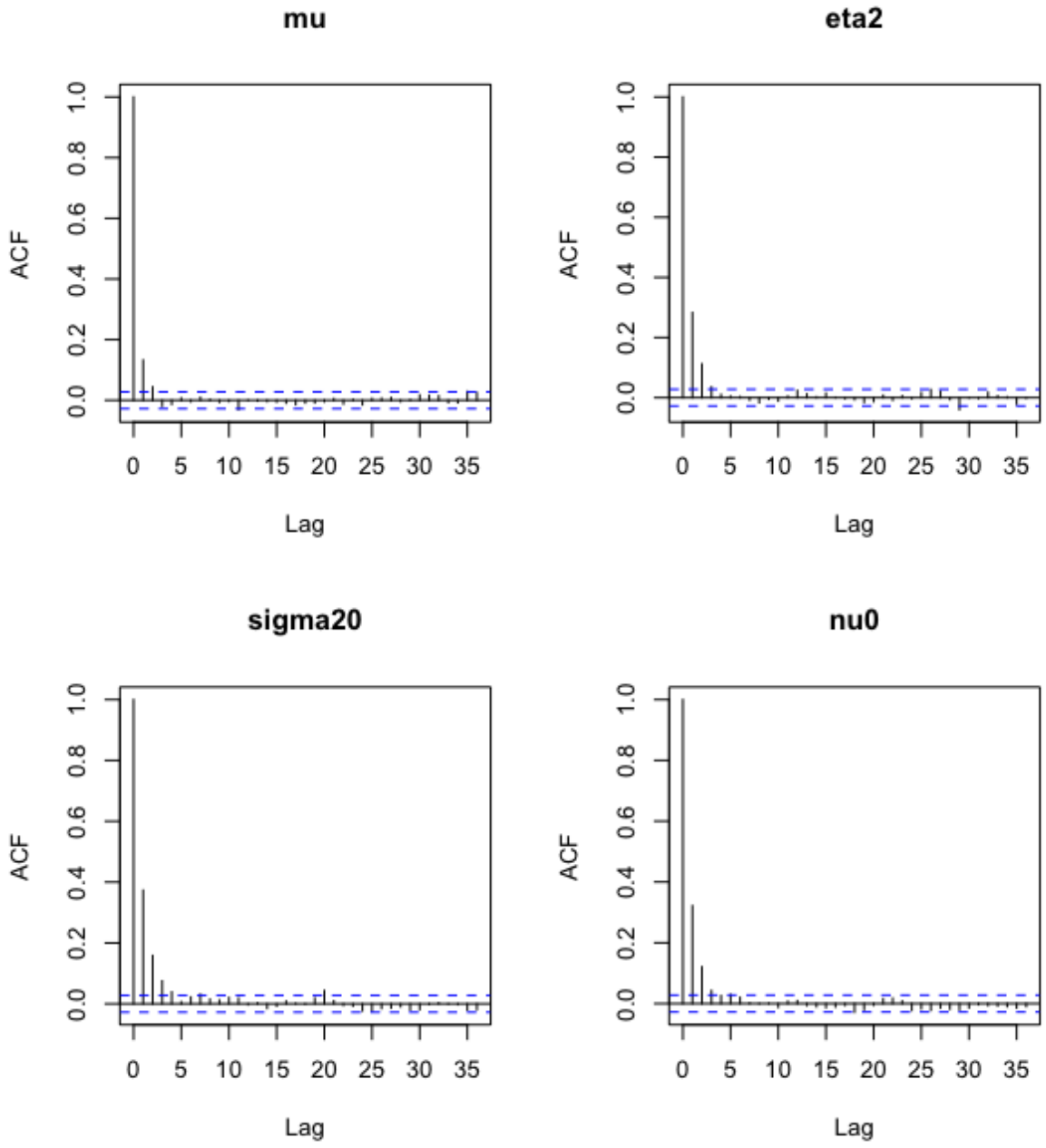

Figure 11: Avtokorelacije za hiperparametre

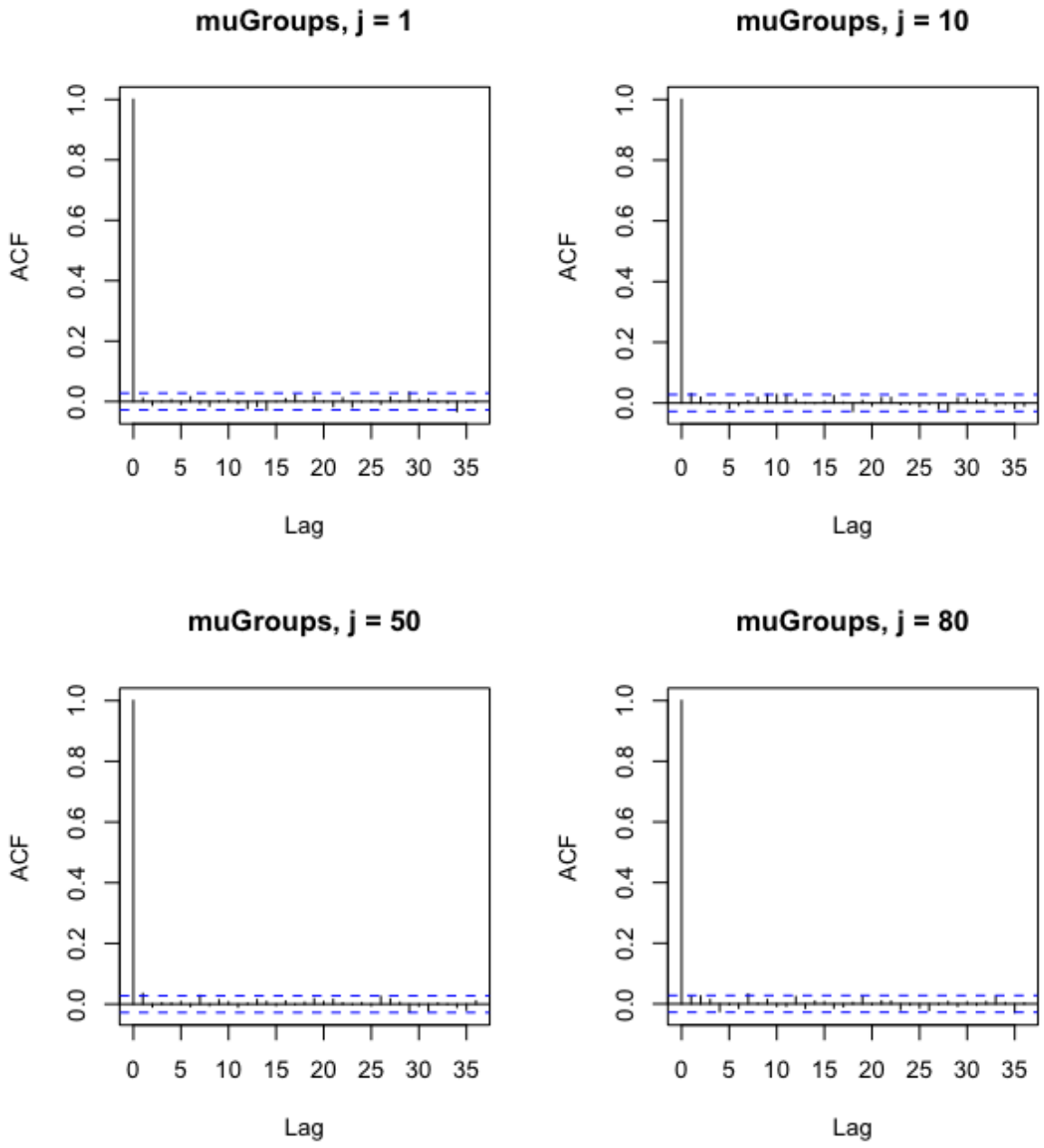

Figure 12: Avtokorelacije za parameter `muGroups`.

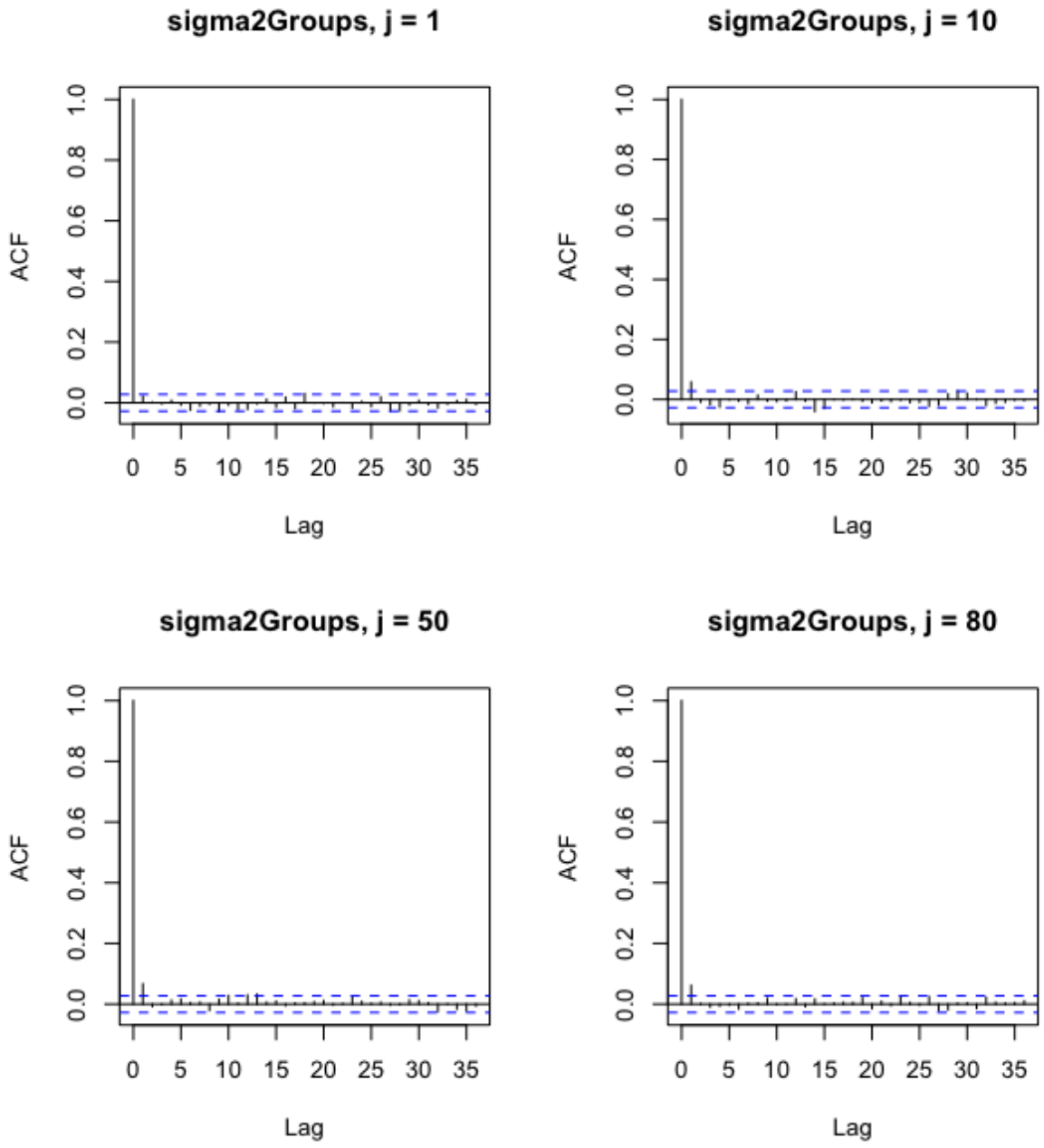

Figure 13: Avtokorelacije za parameter `sigma2Groups`.

Opazimo, da v splošnem ni težav z avtokorelacijo. Omenimo le, da je pri hiperharametrih opazno, da se avtokorelacija malce kasneje približa 0 (pri zamiku 1 je avtokorelacije po definiciji 1, zato je na začetku povsod tam).
Poglejmo sedaj še naslednje izračune, ki so povezani z avtokoreliranostjo.

Effective sample size.
Effective sample size nam pove, za koliko n.e.p. je naš vzorec "vreden".

- Hiperparametri:
  - `mu:  3841.048`
  - `eta2:  2604.908`
  - `sigma20:  2181.228`
  - `nu0:  2563.581`
- muGroups:
  - `j = 1:  5000`
  - `j = 10:  4698.121`
  - `j = 50:  4662.577`
  - `j = 80:  4484.269`
- sigma2Groups:
  - `j = 1:  4788.897`
  - `j = 10:  4452.038`
  - `j = 50:  4374.3`
  - `j = 80:  4419.596`

Takoj opazimo, da je *Effective sample size* pri hiperparametrih, kjer je bila opažena tudi malenkost slabša korelacija, nižji, kot pri ostalih. *Effective sample size* je torej višji pri parametih z višjo avtokorelacijo in obratno.

3. naloga: Marginalne aposteriorne porazdelitve

Poglejmo si marginalne aposteriorne porazdelitve za hiperparametre in nekaj ostalih parametrov (podobno kot prej za $j = 1, 10, 50, 80$).

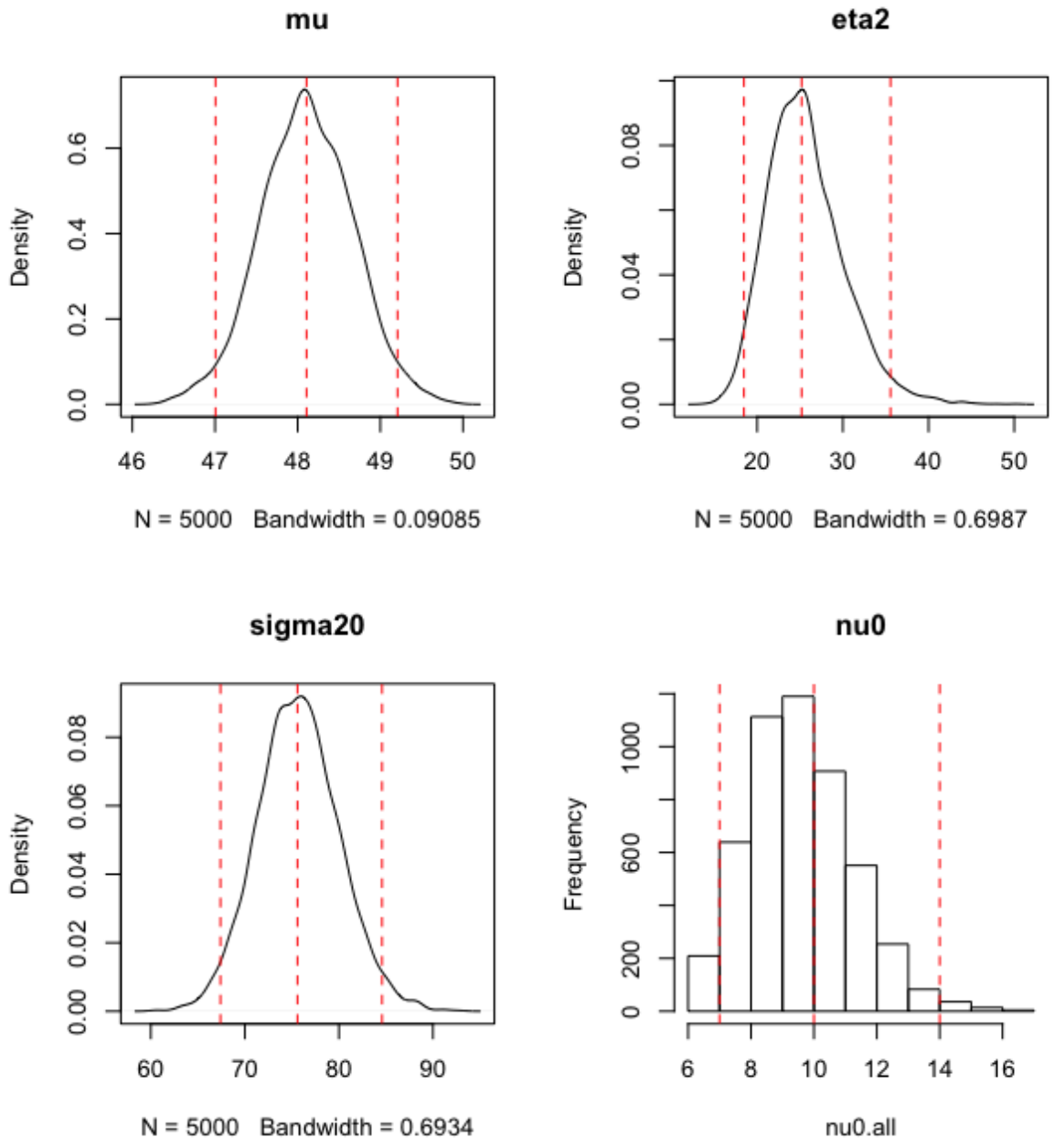

Figure 14: Marginalne aposteriorne porazdelitve za hiperparametre.

Opazimo, da je graf za parameter `mu` simetričen in normalno porazdeljen.
Podobno bi lahko trdili tudi za parameter `sigma20`, pa čeprav vemo, da je vzorčen iz gama porazdelitve. Ker pa gama porazdelitev pri dovolj velikih parametrih konvergira k normalni (oz. jo lahko aproksimiramo z normalno porazdelitvijo), je rezultat in grafika tukaj smiselna.
Parameter `eta2` je nekoliko asimetričen, ker izhaja iz inverzne gamma porazdelitve.

Ker ima `nu0` diskretne vrednosti, je njegova porazdelitev prikazan s histogramom. Opazimo, da je nekoliko asimetričen oz. nagnjen v levo.
Zapišimo še intervale zaupanja (označeno z rdečo) v zaporedju kot jih izpiše `R (2.5% 50% 97.5%)`:

- `mu: 47.00592 48.10700 49.20768`
- `eta2: 8.46046 25.18481 35.56685`
- `sigma20: 67.40757 75.59166 84.58613`
- `nu0: 7 10 14`

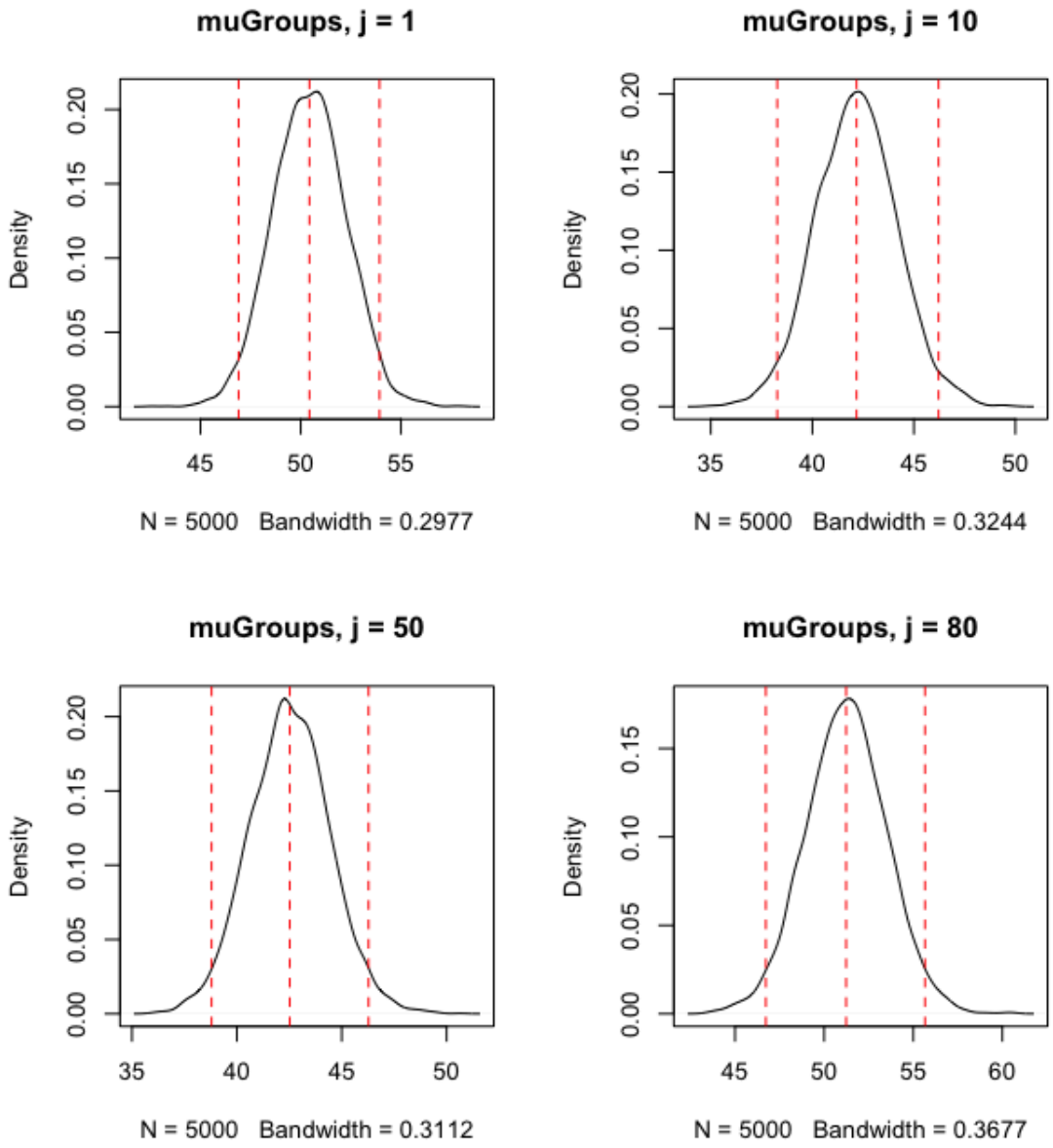

Figure 15: Marginalne aposteriorne porazdelitve za parameter `muGroups`.

Grafi porazdelitev izgledajo po pričakovanjih, saj smo jih vzorčili iz normalne porazdelitve – zato izgledajo normalno porazdeljeni, simetrični.
Zapišimo še intervale zaupanja (označeno z rdečo) v zaporedju kot jih izpiše `R (2.5% 50% 97.5%)`:

- `muGroups, j = 1:  46.90124 50.43449 53.93451`
- `muGroups, j = 10:  38.27189 42.17707 46.21525`
- `muGroups, j = 50:  38.79428 42.51917 46.27882`
- `muGroups, j = 80:  46.72947 51.23621 55.65984`

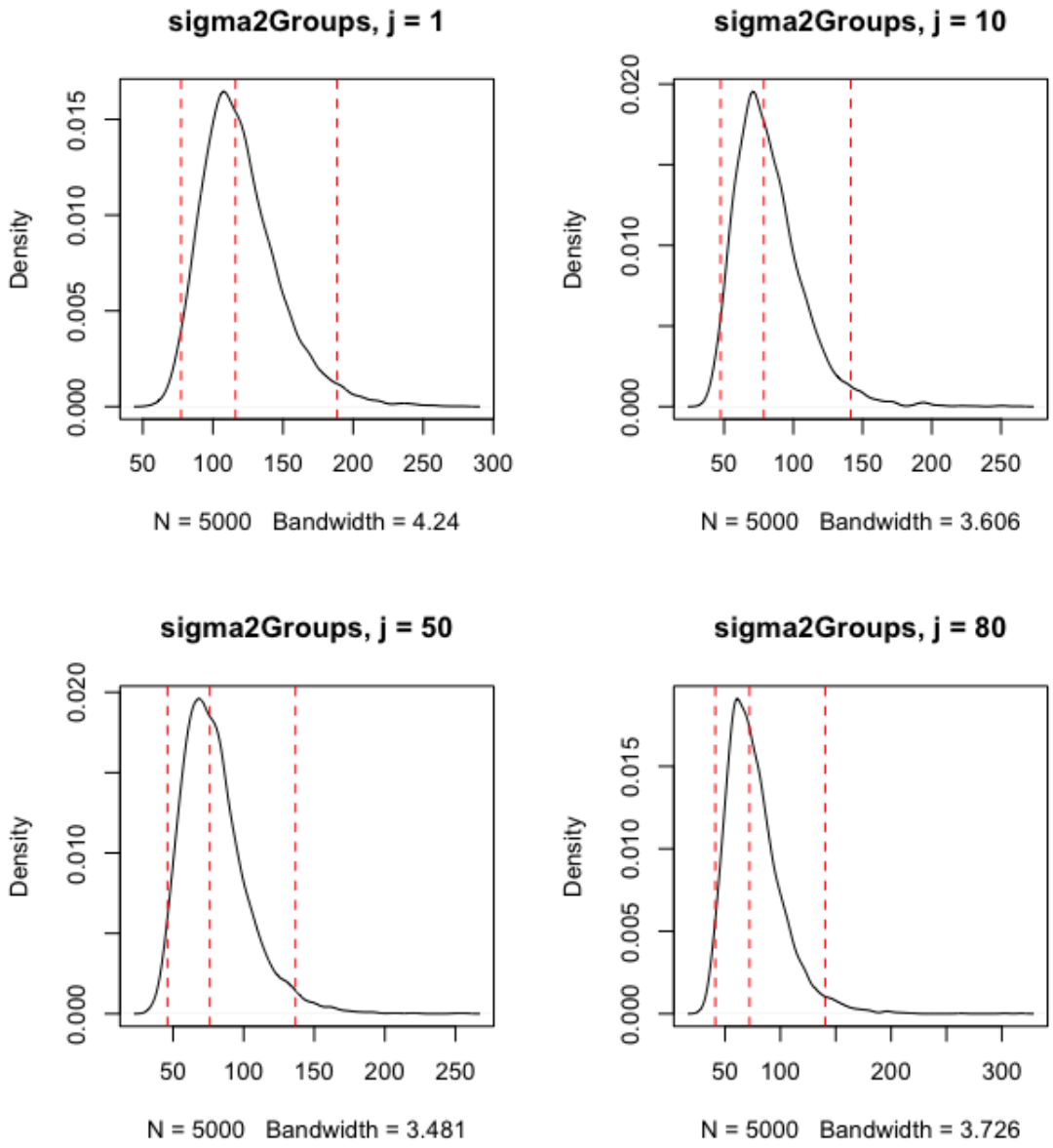

Figure 16: Marginalne aposteriorne porazdelitve za parameter `sigma2Groups`.

Tudi tukaj so rezultati pričakovani. Opazimo asimetričnost porazdelitev, kar je pričakovano, saj vzorčimo iz inverzne gamma porazdelitve.
Zapišimo še intervale zaupanja (označeno z rdečo) v zaporedju kot jih izpiše `R (2.5% 50% 97.5%)`:

- `sigma2Groups, j = 1:  77.31629 116.05267 188.50244`
- `sigma2Groups, j = 10:  47.39666 78.63839 141.44978`
- `sigma2Groups, j = 50:  46.05211 76.02615 136.74937`
- `sigma2Groups, j = 80:  41.14931 72.06370 140.34279`

Še splošen komentar:
Iz zadjih dveh sklopov grafov (za `muGroups` in `sigma2Groups`) razberemo povprečje oz. varianco ocen matematike za posamezne šole, zato si jih lahko interpretiramo kot prikaz uspešnosti posamezne šole na področju matematike.

4. naloga: Shrinkage Poglejmo si najprej *shrinkage* za povprečja.

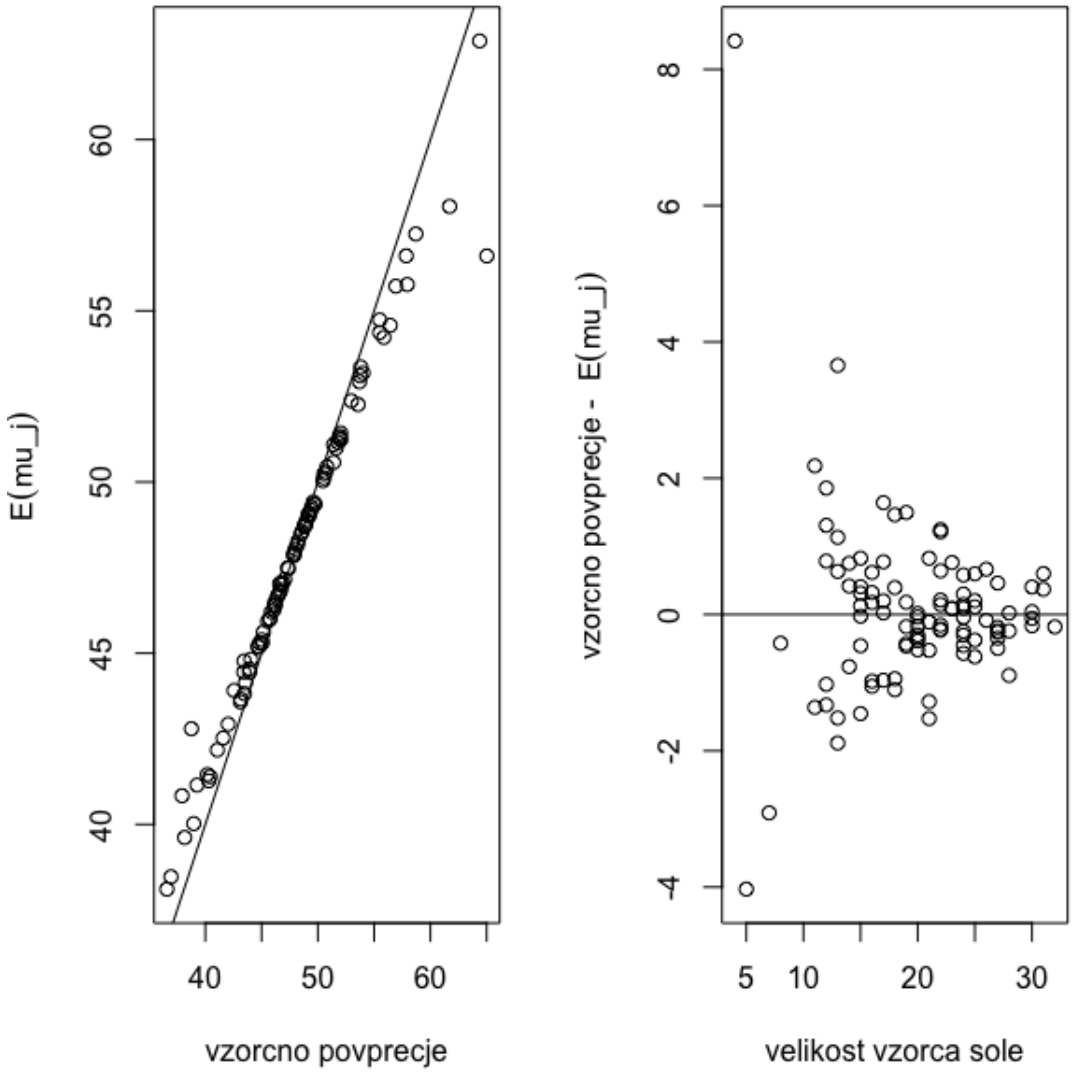

Figure 17: *Shrinkage* za povprečja.

Lepo se vidi *shrinkage* efekt. Opazimo, da so skupine, ki bolj odstopajo od povprečja, veliko bolj skrčene k skupnemu povprečju. Na desni sliki opazimo, da je za skupine z manjšim vzorcem (velikost vzorca šol) premik večji, za večje skupine pa manj (bolj verjamemo večjim skupinam).

To je zato, ker pri šoli z manjšim vzorvem lahko bolj zgrešimo povprečje šole in zato bolj verjamemo skupnemu povprečju kor pa njenemu vzorčnemu povprečju – zato se k skupnemu povprečju bolj približamo/premaknemo kot pa pri večjih šolah.
Podobno poglejmo še za variance.

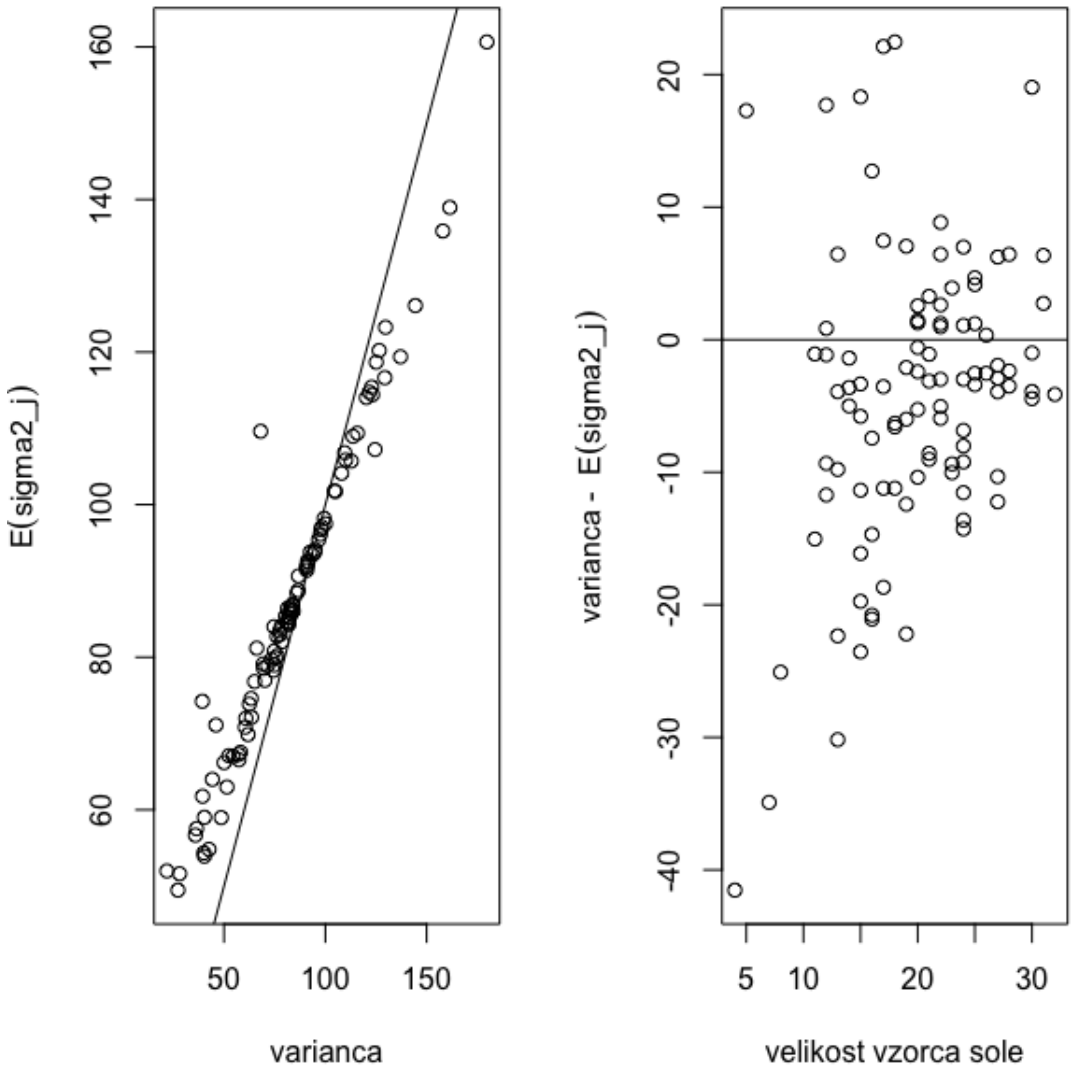

Figure 18: *Shrinkage* za variance.

Tukaj lahko rečemo podobno, sploh za prvo sliko. Če si pa ogledamo drugo sliko, bi lahko rekli, da velikost skupine ne vpliva toliko na skrčenje variance kot pri prejšnjem primeru s povprečji, je pa to še vedno moč opaziti.

5. naloga: Primerjava z modelom iz vaj

Če naš model primerjamo z modelom iz vaj, pri rezultatih ne opazimo večjih razlik.
Z dodajanjem variance za posamezne skupine nismo konkretno prispevali k porazdelitvam. Kljub temu, da z dodajanjem variance prinesemo dodatno informacijo o šolah (teh informacij prej nismo imeli, saj smo predpostavljali enake variance), pa vpeljava različnih varianc za naše podatke k temu modelu ni bila potrebna.

Kater model se odločimo uporabiti je seveda odvisno od veliko dejavnikov. Če bi imeli npr. veliko podatkov, informacija o posameznih variancah pa ne bi prispevala dodatnega znanja (ali pa da bi nas zanimala samo povprečja), potem bi uporabila model iz vaj. Po drugi strani pa ta model iz domače naloge ni veliko bolj časovno zahteven, zato bi, če bi nam informacija o variancah koristila in prispevala koristna znanja, uporabili ta model.
Zaključimo torej, da za naše podatke iz te domače naloge ne bi (nujno) potrebovali modela z različnimi variancami – po drugi strani nas pa ne stane veliko več časa, saj podatkov ni veliko. Tako da bi se lahko odločili za oboje, rezultati pa bi bili podobno interpretativni v obeh primerih.