# 3. DN - Bayesova Statistika: Primer hierarhicnega modela z Gibbsovim vzorcevalnikom

Jan Črne

Hierarhicni model iz 6. sklopa posplosimo tako, da dovolimo razlicne variance skupin $\sigma_j^2$, ki so neodvisno enako porazdeljene z inverzno gama porazdelitvijo s parametri, za katere dolocimo neko hiperapriorno porazdelitev. Natancneje:

Variabilnost znotraj skupine (*within-group sampling variability*):

$$(X_{1,j}, \ldots, X_{n_j,j}) \mid \mu_j, \sigma_j^2 \sim \text{n.e.p. } N(\mu_j, \sigma_j^2)$$

Variabilnost med skupinami (*between-group sampling variability*):

$$(\mu_1, \ldots, \mu_m) \mid \mu, \eta^2 \sim \text{n.e.p. } N(\mu, \eta^2), \quad \text{tj. enako kot prej;}$$

$$(\sigma_1^2, \ldots, \sigma_m^2) \mid \sigma_0^2, \nu_0 \sim \text{n.e.p. Inv-Gama}(\nu_0/2, \sigma_0^2\nu_0/2).$$

Hiperapriorne porazdelitve $\mu, \eta^2, \sigma_0^2, \nu_0$ naj bodo neodvisne. Hiperapriorna porazdelitev $\mu$ in $\eta^2$ naj bo enaka kot prej. Hiperapriorna porazdelitev za $\sigma_0^2$ naj bo:

$$\sigma_0^2 \sim \text{Gama}(a, b), \quad \text{kjer vzamemo } a = 2, b = 1/10;$$

Hiperapriorna porazdelitev za $\nu_0$ naj bo diskretna z vrednostmi $k \in \{1, 2, \ldots\}$, za katero velja $P(\nu_0 = k) \propto e^{-\alpha k}$, kjer vzamemo $\alpha = 2$.

Za uporabo Gibbsovega vzorcevalnika potrebujemo pogojne porazdelitve (hiper)parametrov. Hiperparmetra $\mu$ in $\eta^2$ imata enako pogojno porazdelitev kot prej. Pri pogojni porazdelitvi za $\mu_j$ moramo v prejsnji formuli nadomestiti $\sigma^2$ z $\sigma_j^2$, tj.

$$\mu_j \mid \text{vse ostalo} \sim N\left(\frac{\bar{x}_{\cdot j} n_j/\sigma_j^2 + \mu/\eta^2}{n_j/\sigma_j^2 + 1/\eta^2}, \left[n_j/\sigma_j^2 + 1/\eta^2\right]^{-1}\right).$$

Z uporabo izpeljav pri normalnem modelu lahko izpeljemo, da za vsak $j$ velja

$$\sigma_j^2 \mid \text{vse ostalo} \sim \text{Inv-Gama}\left(\frac{\nu_0 + n_j}{2}, \frac{\nu_0\sigma_0^2 + \sum_{i=1}^{n_j}(x_{i,j} - \mu_j)^2}{2}\right).$$

Izpeljemo lahko

$$\sigma_0^2 \mid \text{vse ostalo} \sim \text{Gama}\left(a + \frac{m\nu_0}{2}, b + \frac{\nu_0}{2}\sum_{j=1}^{m}(1/\sigma_j^2)\right).$$

Za pogojno porazdelitev $\nu_0$ lahko pokazemo, da je za vsak $k \in \{1, 2, \ldots\}$

$$P(\nu_0 = k \mid \text{vse ostalo}) \propto \left(\frac{(k\sigma_0^2/2)^{k/2}}{\Gamma(k/2)}\right)^m \left(\prod_{j=1}^{m}(1/\sigma_j^2)\right)^{k/2-1} \exp\left(-k\left(\alpha + \frac{1}{2}\sigma_o^2\sum_{j=1}^{m}(1/\sigma_j^2)\right)\right). \quad (1)$$

Iz te porazdelitve lahko vzorcimo, ce se omejimo na velik izbor $k \in \{1, 2, \ldots k_{\max}\}$, za te $k$ izracunamo zgornji izraz in nato vzorcimo $\nu_0$ iz mnozice $\{1, 2, \ldots k_{\max}\}$ z utezmi (1). To je bolje narediti na log skali in na koncu uteži "preskalirati". V pomoc pri vzorcenju iz te porazdelitve vam je lahko naslednja koda v R:

**Vasa naloga je**, da za opisani hierarhicni model z razlicnimi variancami po skupinah vzorcite aposteriorno porazdelitev parametrov s pomocjo Gibbsosovega vzorcevalnika. To naredite tako, da ustrezno predrugacite prejsno kodo Gibbsosovega vzorcevalnika za hierarhicni model z enakimi variancami.

# 1 Podatki

V raziskavi *Educational Longitudinal Study (ELS)* iz leta 2002 so preucevali rezultate testov ucencev ameriskih srednjih sol. Podane imamo rezultate matematicnih testov ucencev 10. razreda iz 100 javnih srednjih sol (velike sole z vsaj 400 ucenci 10. razreda, urbano okolje).

Za vsakega ucenca imamo podano solo in rezultat matematicnega testa – nasi podatki so torej vecnivojski/hierarhicni.

Rezultati matematicnega testa so del nacionalnega preverjanja znanja, ki naj bi bil konstruiran tako, da je pricakovana vrednost enaka 50 in standardni odklon 10.

Oglejmo si podatke.

```
source("podatki_sole.R")
str(pod)
```

```
## 'data.frame':    1993 obs. of  2 variables:
##  $ school   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ mathscore: num  52.1 57.6 66.4 44.7 40.6 ...
```

```
library(dplyr)
# Preureditev podatkov:
pod.sole <- pod %>%
```

```
  group_by(school) %>%
  summarise(povprecje = mean(mathscore),
            n=length(mathscore),
            varianca = var(mathscore))
```

## 2 Preurejen Gibbsov vzorčevalnik

```
### Parametri (hiper)apriornih porazdelitev (enako kot prej)
sigma20 = 100
nu0 = 1
eta20 = 100
kappa0 = 1
mu0 = 50
tau20 = 25

### Pripravimo si kolicine, ki jih bomo potrebovali iz podatkov
x = pod
m = length(pod.sole$school)
n = pod.sole$n
x.povpr = pod.sole$povprecje
# NA NOVO
x.var   <- pod.sole$varianca  # variance posameznih šol, sedaj te podatke rabimo

# NA NOVO
# za variance posameznih šol
a <- 2 # parameter za  gama porazdeljeno sigma2
b <- 1/10 # parameter za  gama porazdeljeno sigma2
alpha <- 2 # parameter za diskretno porazdeljeni nu0
k.max <- 1000  # izberemo si neko dovolj veliko vrednost za verjetnost
               # iz katere se bomo vzorčili nu0

### Dolocimo si zacetne vrednosti
muGroups = x.povpr
sigma2Groups = x.var  #sedaj za vsako šolo rabimo svojo začetnon vrednost variance
mu = mean(muGroups)
eta2 = var(muGroups)

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

muGroups.all = matrix(nrow = n.iter, ncol = m)
sigma2Groups.all = matrix(nrow = n.iter, ncol = m) #za shranjevanje varianc
```

```
sigma20.all = rep(NA, n.iter)
mu.all = rep(NA, n.iter)
eta2.all = rep(NA, n.iter)
nu0.all = rep(NA, n.iter) # za shranjevanje nu0
```

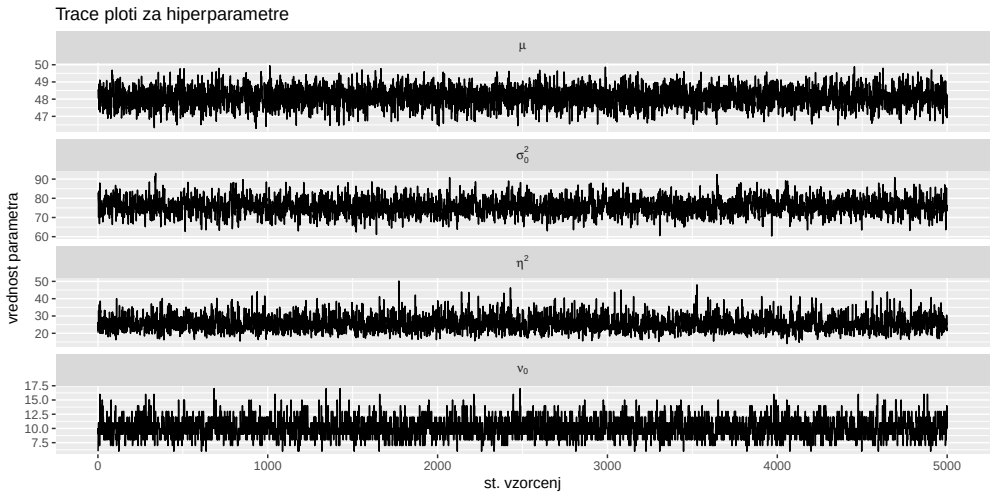
```
### Na prvo mesto si shranimo zacetne vrednosti (nepotrebno)
muGroups.all[1, ] = muGroups
sigma2Groups.all[1] = sigma2Groups
mu.all[1] = mu
eta2.all[1] = eta2

sigma20.all[1] = sigma20 #zac. vr za sigma0
sigma2Groups.all[1,] = sigma2Groups #zac. vrednosti za sigme
nu0.all[1] = nu0 #zac. vr. za nu0

### Pozenemo Gibbsov vzorcevalnik
set.seed(1)
for (s in 1:n.iter) {
  ### Vzorcimo muGroups (sigma2 smo zamenjali s sigma2Groups[j])
  for (j in 1:m) {
    muGroups[j] = rnorm(1,
                        mean = (x.povpr[j] * n[j] / sigma2Groups[j] + mu / eta2) /
                          (n[j] / sigma2Groups[j] + 1 / eta2),
                        sd = sqrt(1 / (n[j] / sigma2Groups[j] + 1 / eta2)))
  }


  # Vzorčimo sigma2_j (NOVO)
  for (j in 1:m) {
    sigma2Groups[j] <- 1 / rgamma(1, (nu0 + n[j]) / 2,
                                  (nu0*sigma20 +
                                     sum((x[x[, 1] == j, 2] - muGroups[j])^2))/2)
  }

   # Vzorčimo sigma20 (hiperparameter) (gamma porazdeljen)
   sigma20 <- rgamma(1, a + m*nu0/2,b +nu0/2*sum(1/sigma2Groups))

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

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

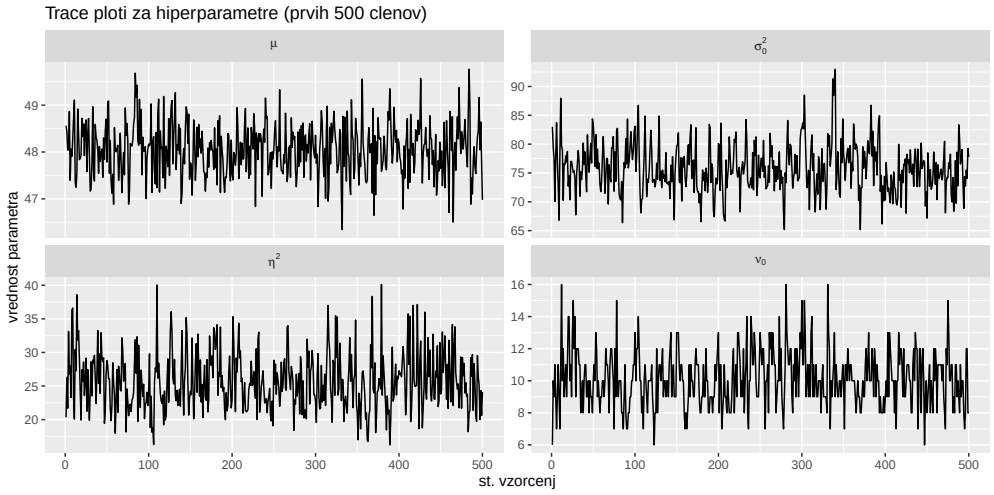
```r
  # Vzorčimo nu0 (NOVO)
  k <- 1:k.max
  logp.nu0 <- m * (0.5 * k * log(k*sigma20/2) - lgamma(k/2)) +
    (k/2-1) * sum(log(1/sigma2Groups)) +
    - k * (alpha + 0.5 * sigma20 * sum(1/sigma2Groups))
  nu0 <- sample(k, 1, prob = exp(logp.nu0 - max(logp.nu0)))

  ### Shranimo nove parametre
  muGroups.all[s, ] = muGroups
  sigma2Groups.all[s, ] = sigma2Groups # za shranjevanje sigma_j
  mu.all[s] = mu
  eta2.all[s] = eta2
  sigma20.all[s] = sigma20 # za shranjevanje sigma0 (hiperparameter)
  nu0.all[s] = nu0 # za shranjevanje nu0
}
```

# 3 Preucevanje konvergence

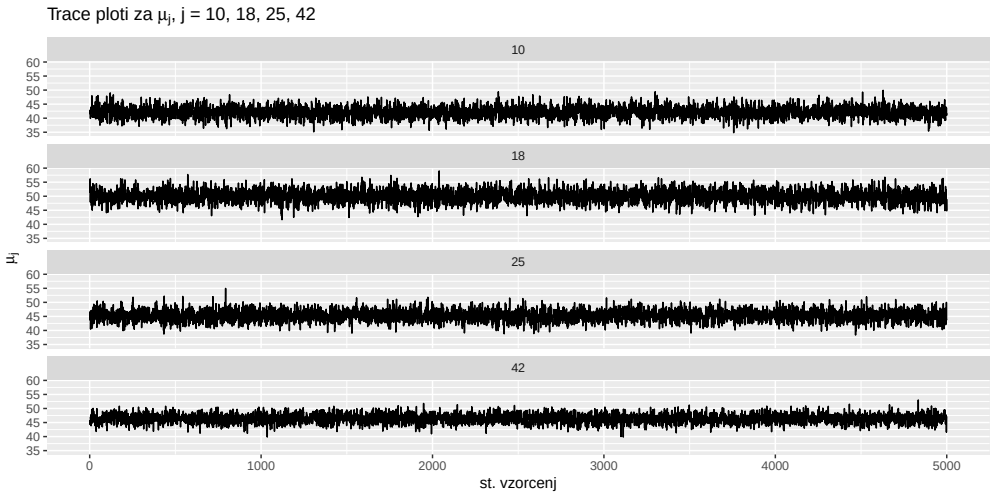
Uporabite zadostno stevilo iteracij in ustrezen *burn-in*. Za dobljeni vzorec preucite konvergenco, tako da sledite korakom v sklopu 6. (*trace plots*, porezdelitve podvzorcev, avtokorelacije, *effective sample size*.) Slednje naredite za vse hiperparametre in nekaj "ne-hiperparametrov". Pri tem tudi komentirajte konvergenco.

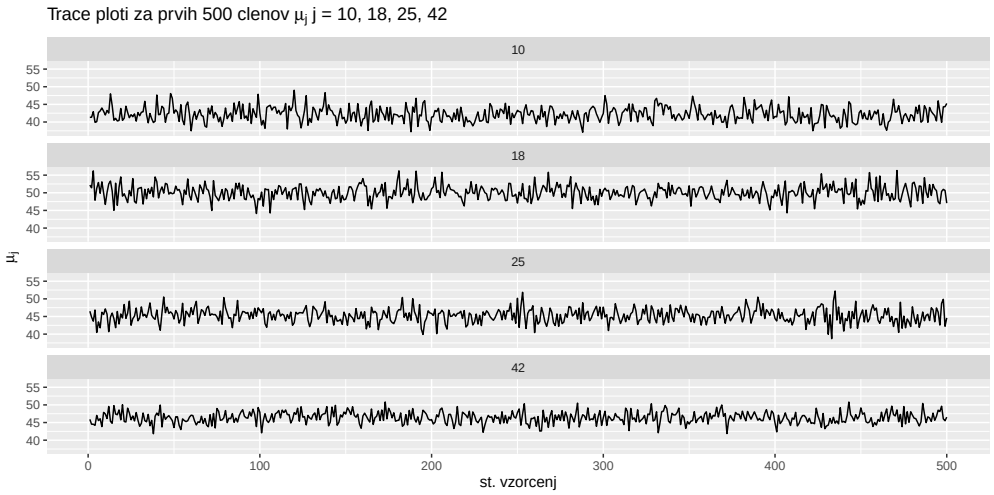
## 3.1 *Trace plots*

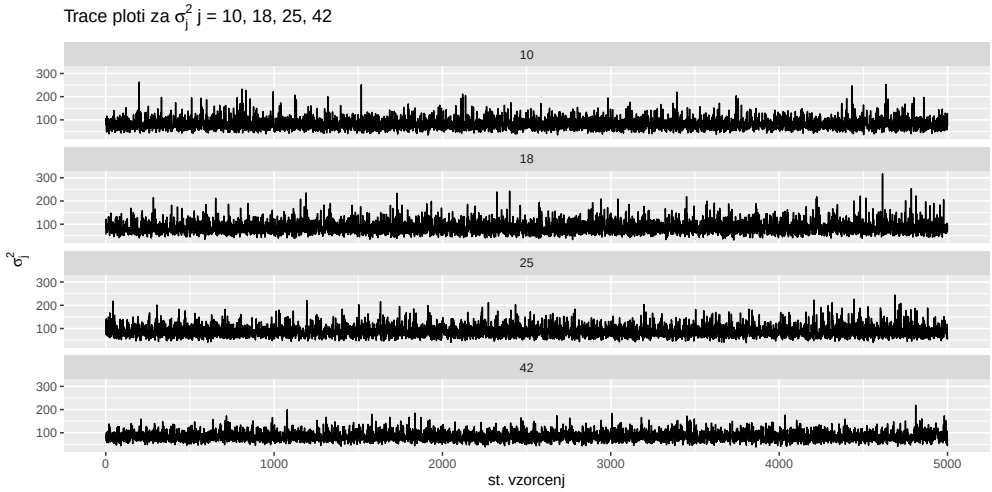


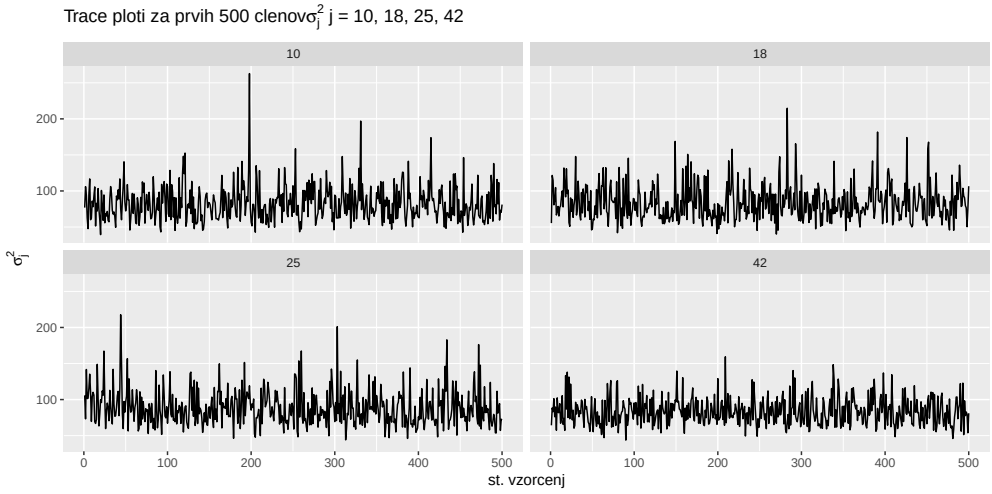



Konvergence izgeldajo dobro, burn-in del ni viden.



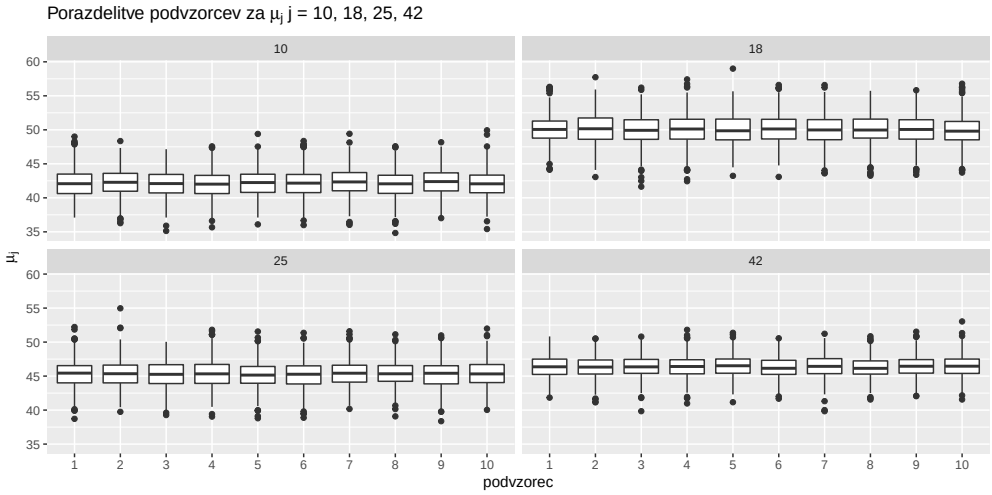



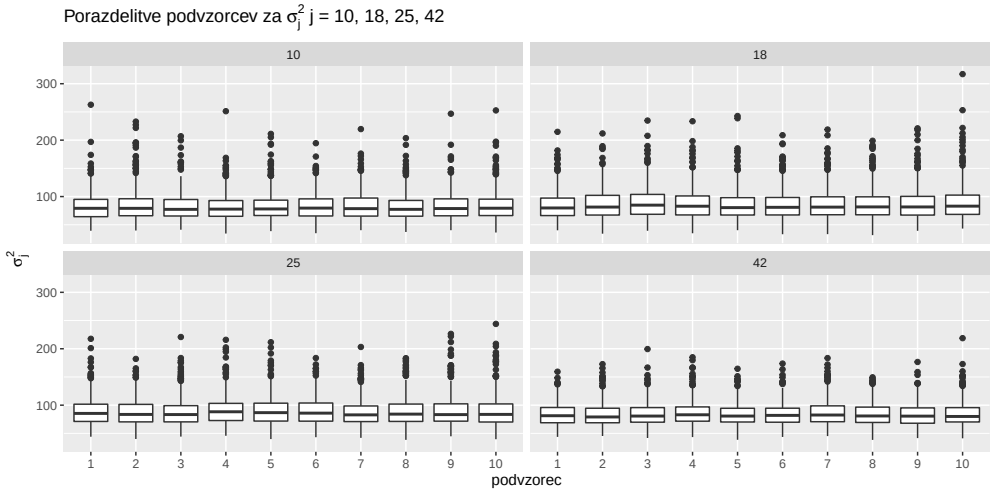


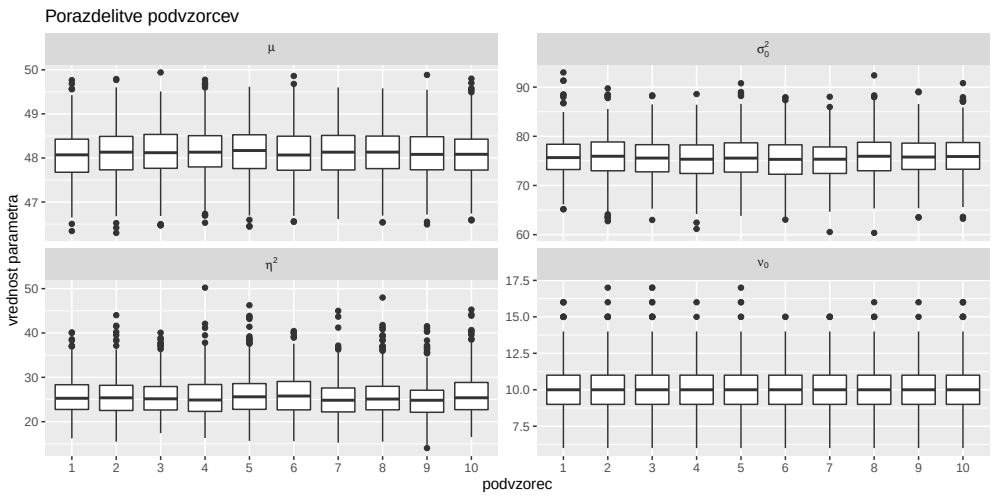




Konvergence izgledajo vredu, prav tako ni vidne potrebe po kakršnemkoli burn-in parametru. Med šolami se podatki pričakovano nekoliko razlikujejo tako v povprečjih kot v variancah.

## 3.2 Porazdelitev podvzorcev

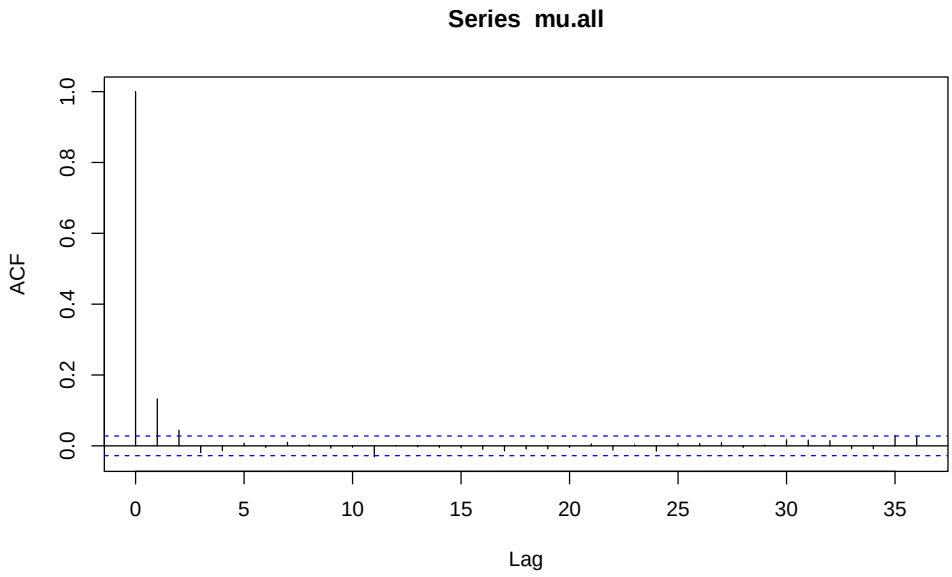


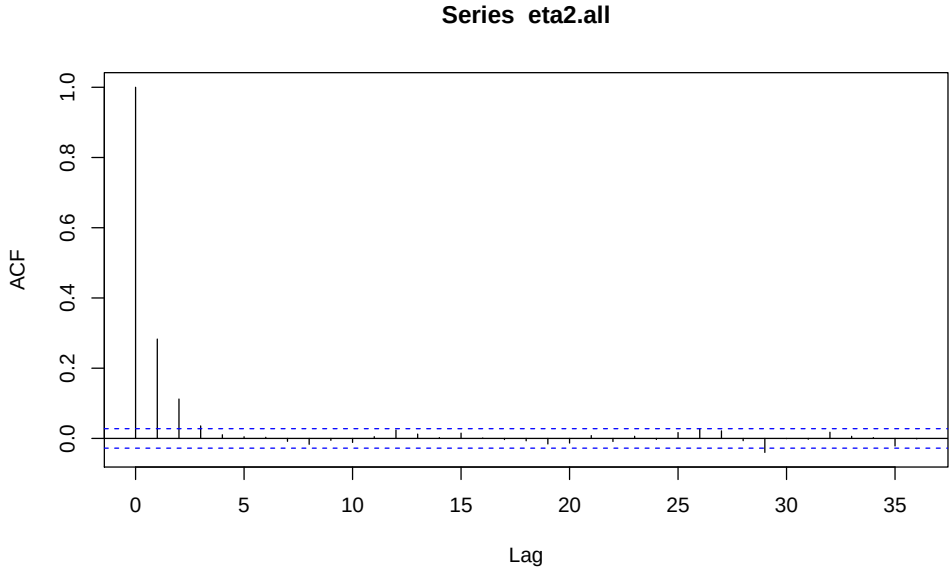



## 3.3 Avtokorelacije

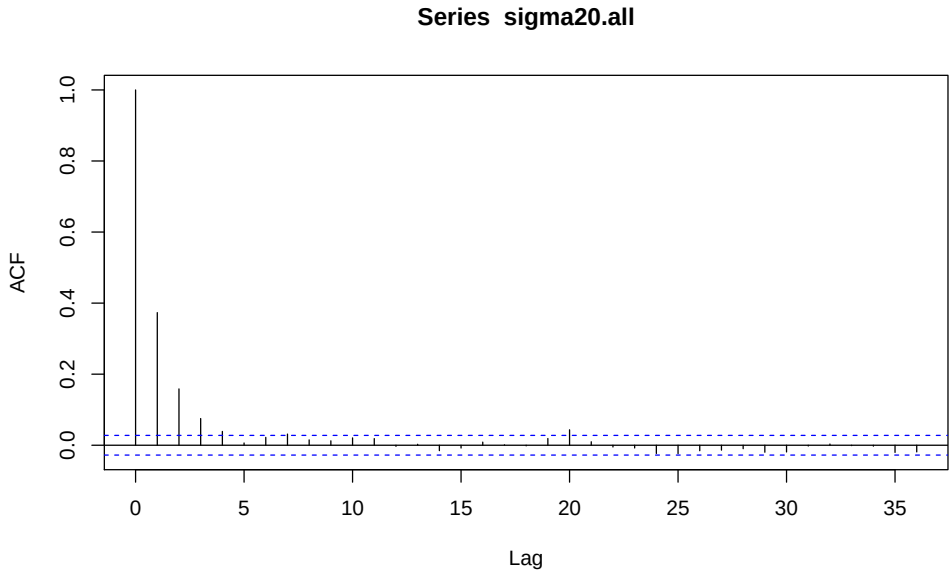
Po domače je avtokorelacija meritev povezave med trenutno vrednostjo slučajne spremenljivke in preteklimi vrednostmi. V naši nalogi si za njeno računanje pomagamo kar z vgrajeno funkcijo acf.



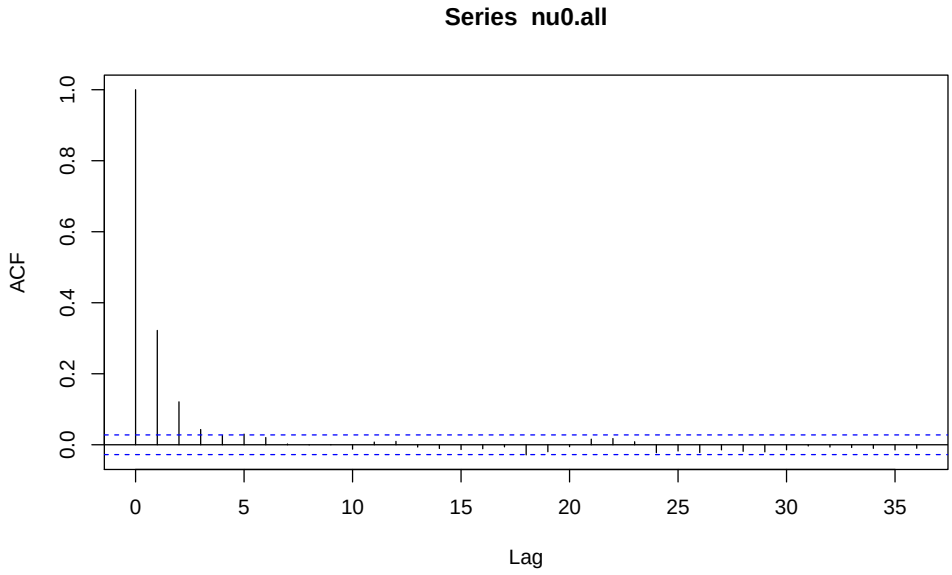

```
## [1]  1.000000000  0.132659758  0.044169347 -0.019126059 -0.013330591
## [6]  0.007427838
```



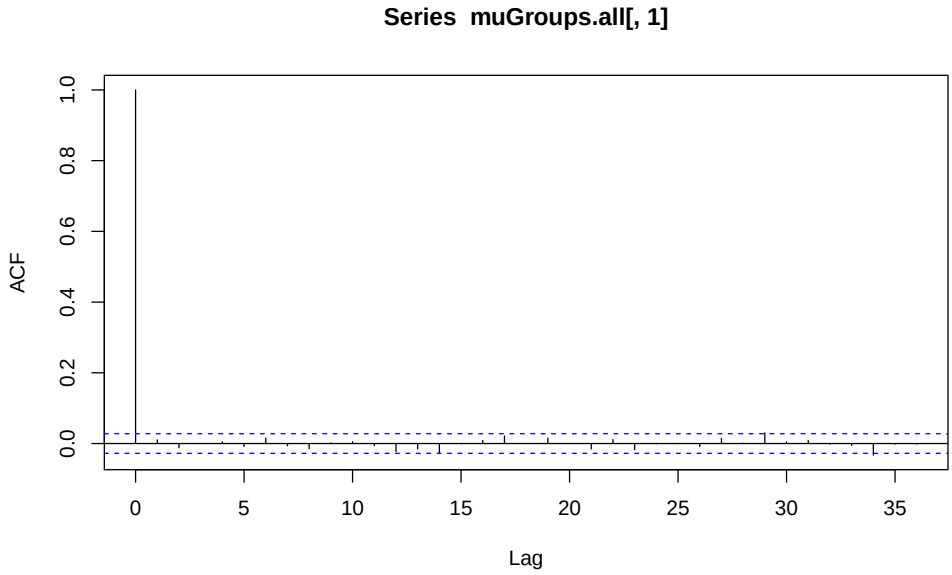

```
## [1] 1.000000000 0.282943265 0.112188187 0.035653575 0.010443963 0.004737707
```



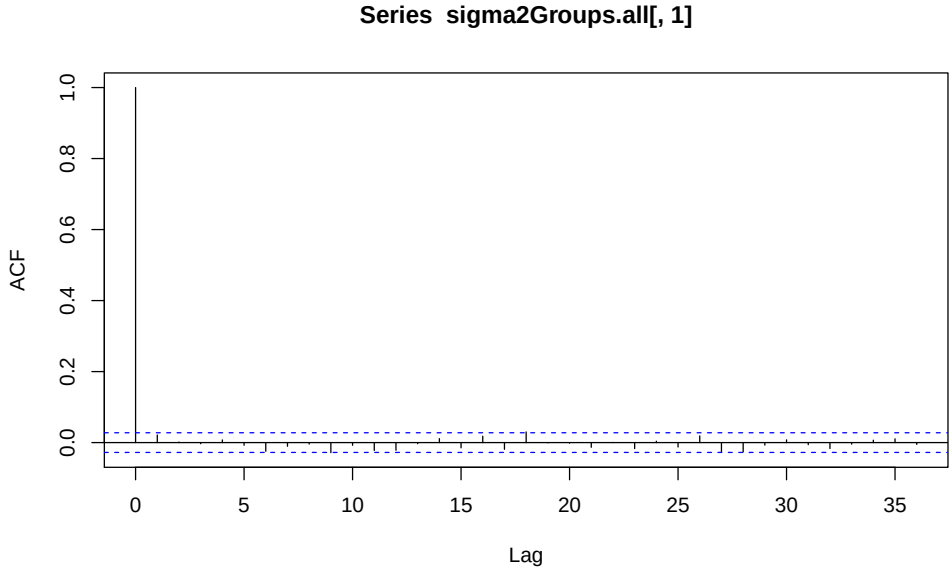

```
## [1] 1.000000000 0.373281669 0.158566687 0.075177602 0.038909983 0.006385229
```




```
## [1] 1.00000000 0.32203534 0.12089819 0.04291857 0.02625595 0.02995445
```




```
## [1]  1.0000000000  0.0104428696 -0.0120241511  0.0001737185  0.0048183359
## [6] -0.0084219244
```




```
## [1]  1.000000000  0.021465531  0.001609685 -0.003247028  0.007603028
## [6] -0.007534900
```

Pri zamiku (*lag*) 1 dobimo po definiciji avtokorelacijo 1, za kasnejse pa si zelimo, da so cim blizje 0 (kar pricakujemo pri n.e.p. vzorcu). Pri vzorcu dobljenim z MCMC metodami bo prisotna avtokorelacija, ki pa se z zamikom (*lag*) zmanjsuje (odvisnost neke vrednosti od vrednosti iz enega koraka nazaj je najvecja, iz dveh korakov nazaj malo manjsa, itd.).

Kako lahko približno izracunamo avtokorelacijo z zamikom 1? Ta izračun, analogen 6. nalogi, je ustrezen za parameter $\mu$. S spodnjim izračunom preverimo, da sta dejanska in približno izračunana vrednost (relativno) blizu.

```
## [1] 0.1326694
```

```
## [1] 0.1326598
```

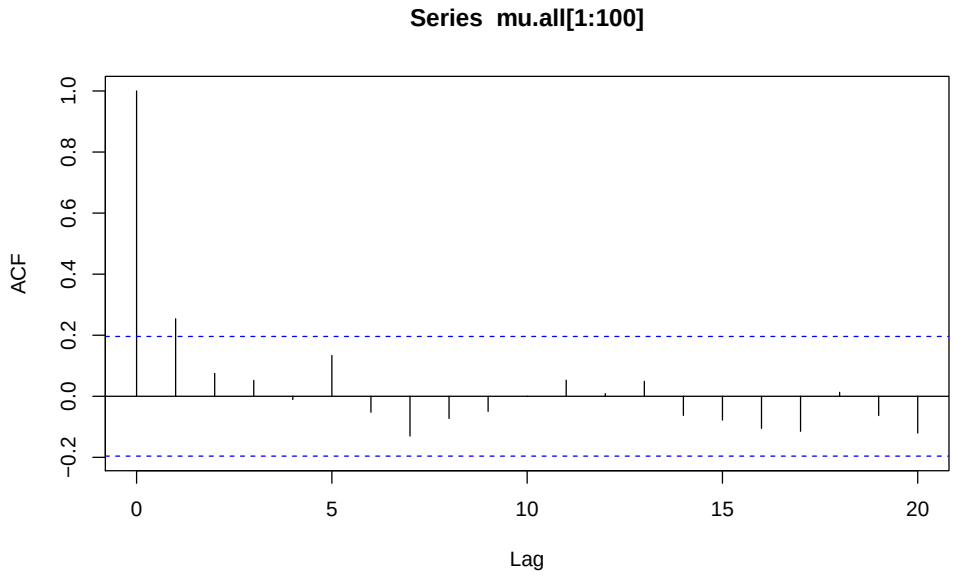
Podobne obravnave bo deležen le še hiperparameter $\nu_0$ zaradi podobnosti računanj avtokorelacij z različnimi zamiki. Oglejmo si še zamik za 3.

```
## [1] 0.04297548
```

```
## [1] 0.04291857
```

Velja omeniti še obravnavo prvih nekaj členov markovske verige, ki smo jo simulirali s splošnim Gibbsovim vzorčevalnikom. Zamakne se meja na grafih avtokorelacije glede na prejšnji primer.

```
acf(mu.all[1:100]) #za prvih 100 iteracij
```

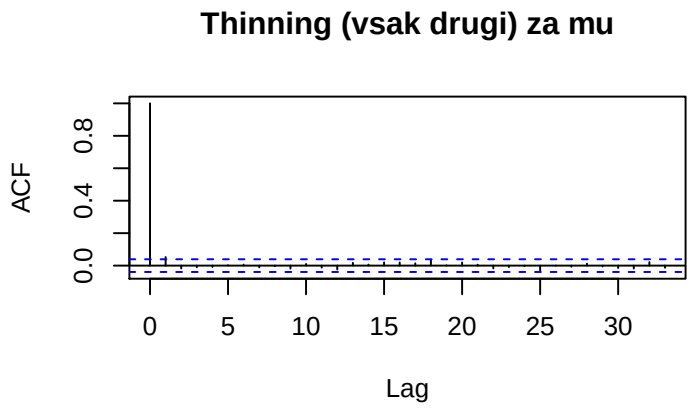


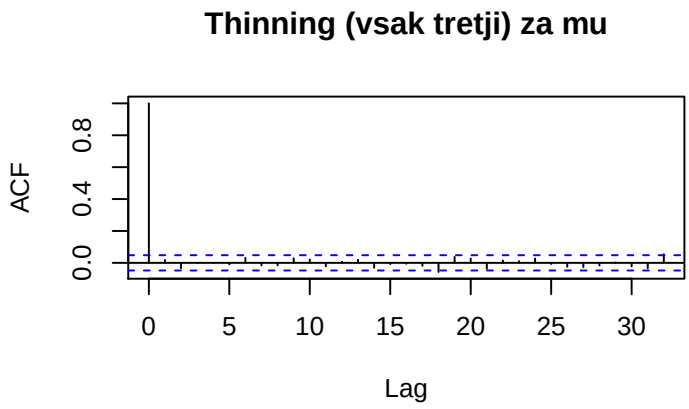

Kaj predstavljata crti?

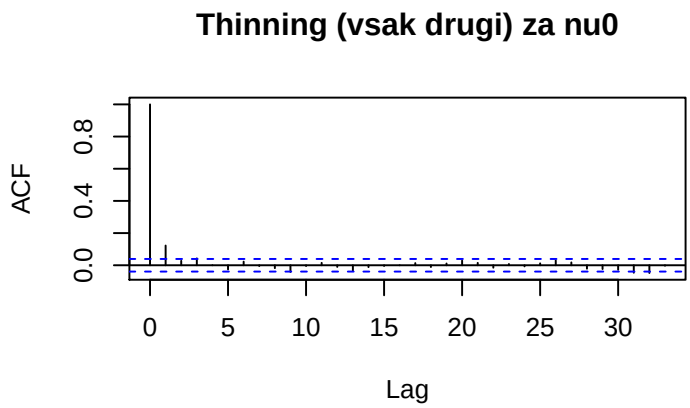
$\pm 1.96/\sqrt{N}$, $N$ stevilo iteracij – avtokorelacije izven obmocja so statisticno znacilno razlicne od 0 (oz. le 5% jih lahko pricakujemo *nekoliko* izven pri n.e.p. vzorcu).

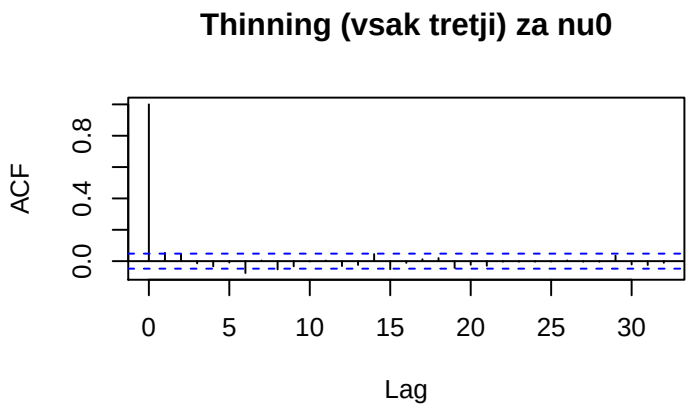
Kaj se v nasem primeru zgodi z avtokorelacijami, ce v zaporedju izbrisemo vsakega drugega (uporabimo *thinning*)?

Specificiramo dva primera za oba parametra, ki smo ju podrobneje obravnavali že zgoraj: $\nu_0$ in $\mu$.



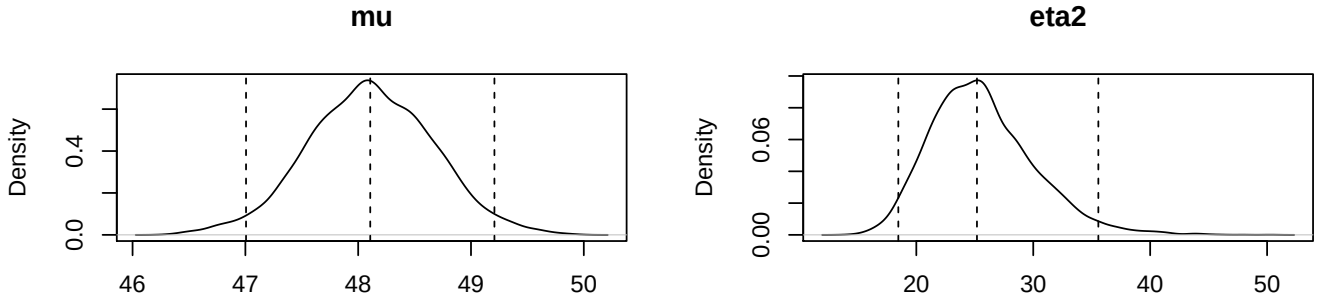



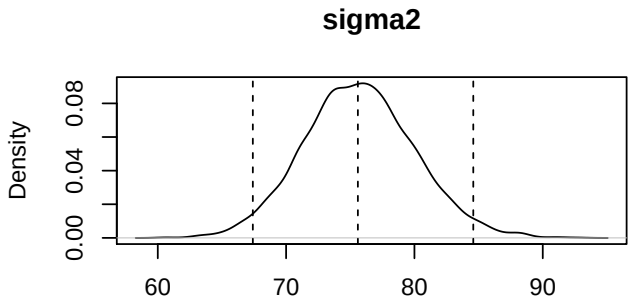

Pricakovano se zmanjsajo, vendar smo pri tem zmanjsali tudi velikost vzorca, za katerega smo ze porabili cas za izracun. Opazna je tudi razlika med tanjšanjem s faktorjem 2 in faktorjem 3.

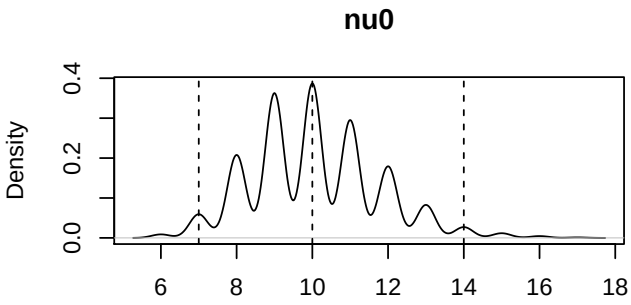
# 4 Robne aposteriorne porazdelitve

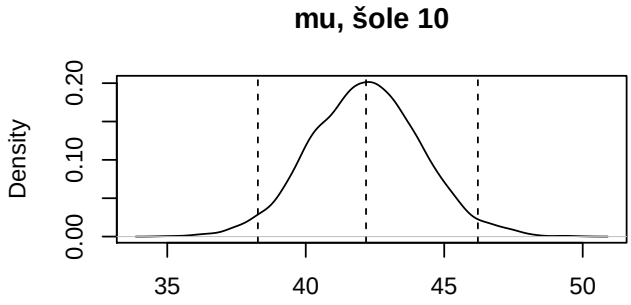


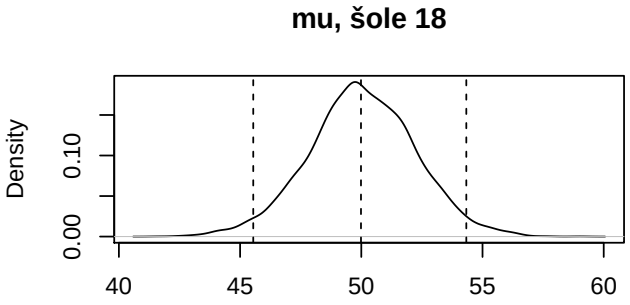


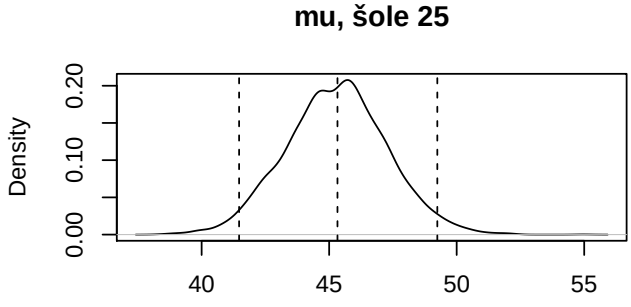


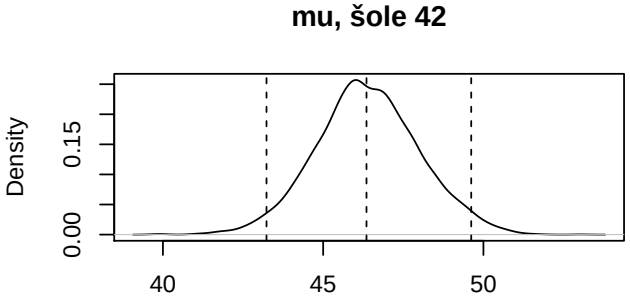


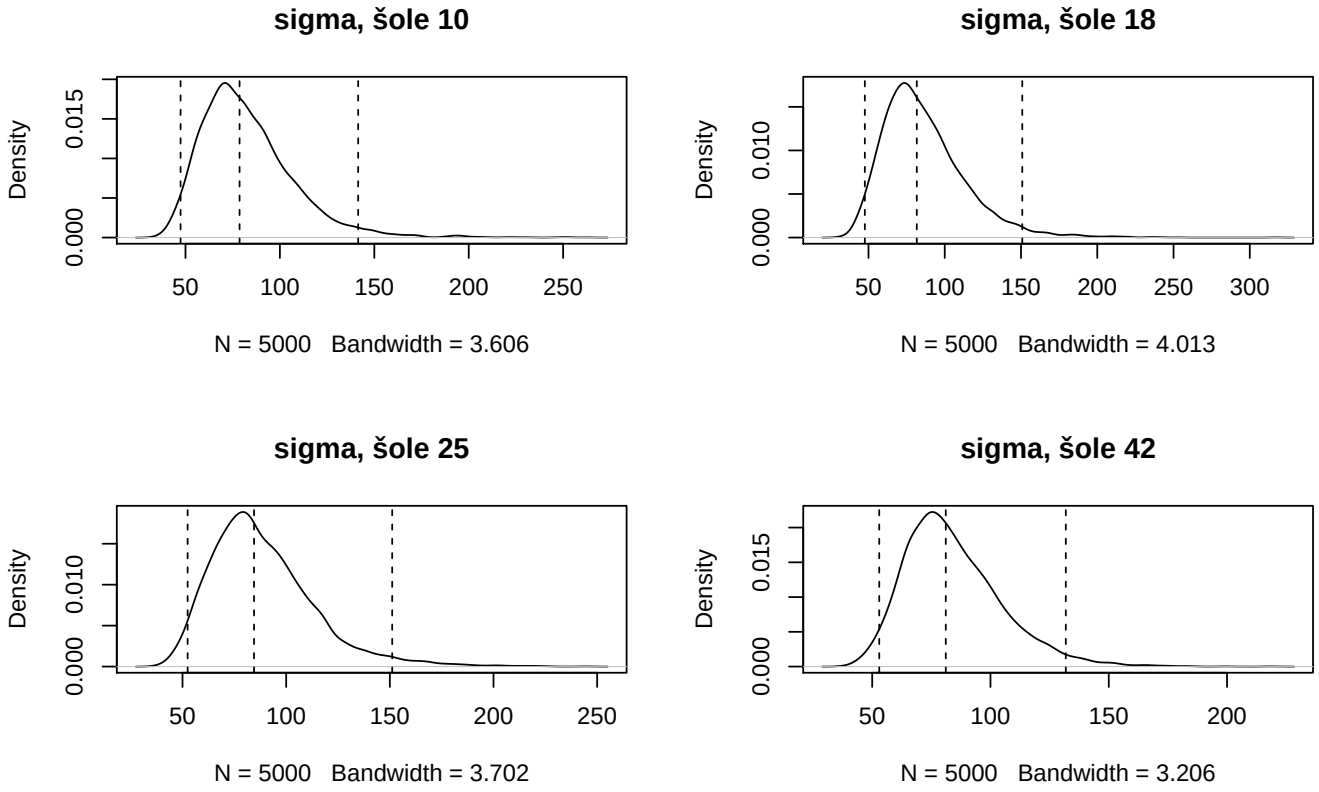



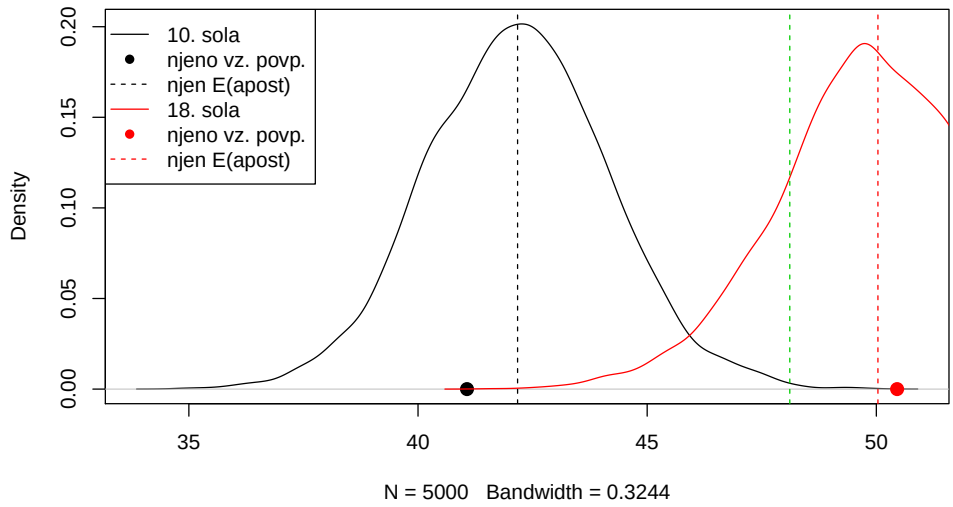



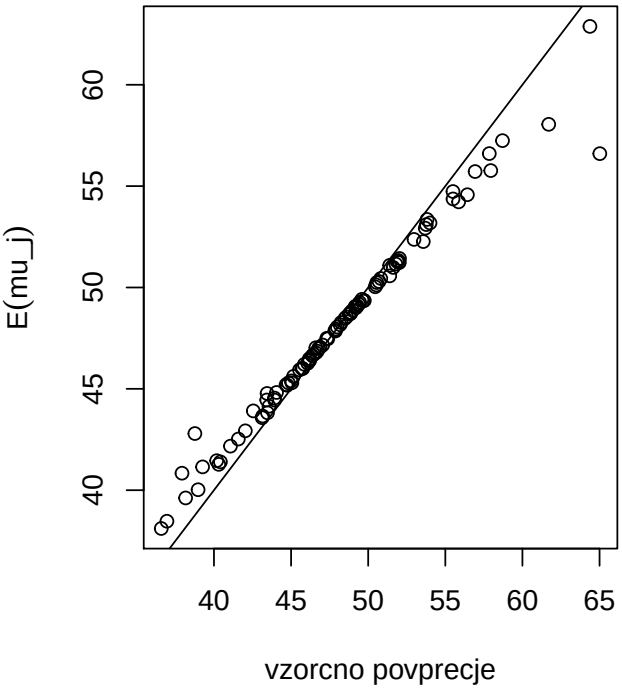


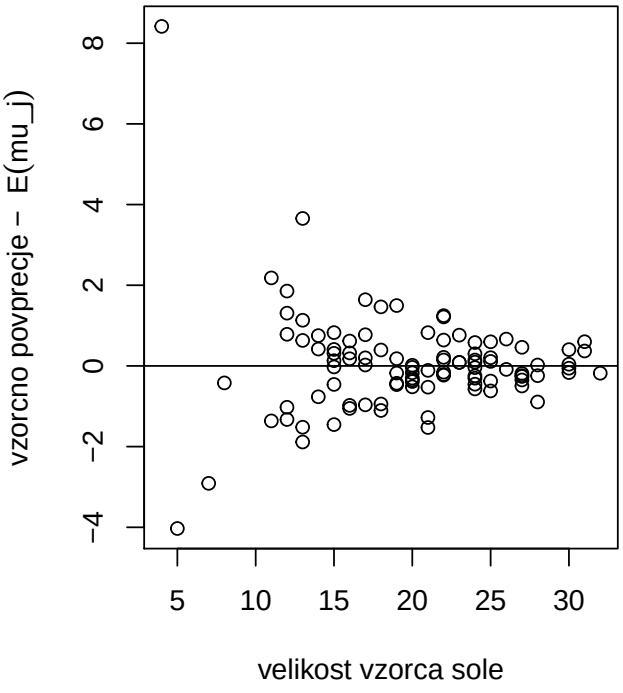

Oblike vseh gostot se ujemajo z oblikami porazdelitev iz katerih vzorčimo, prav tako vidimo da se porazdelitve upanj in varianc med različnimi šolami razlikujejo, kar nam pove nekaj o primerjavi šol.

# 5 *Shrinkage*

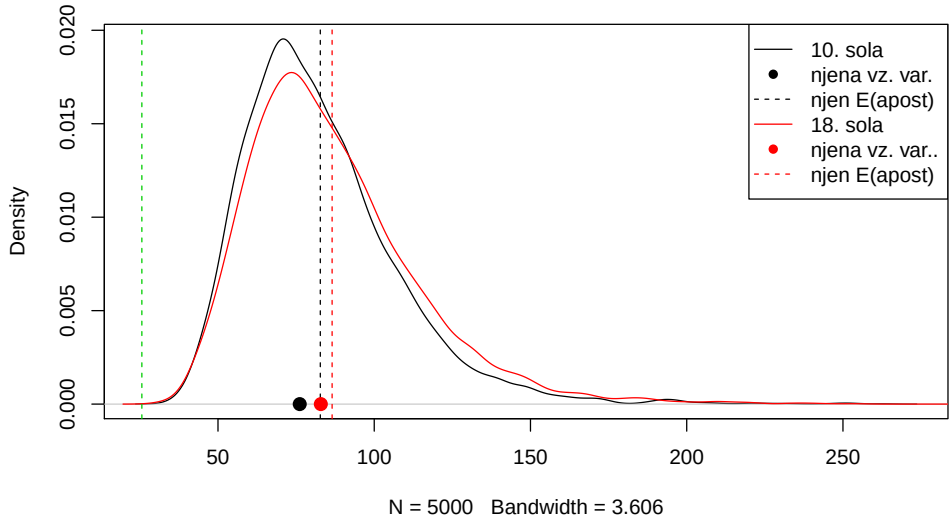
UPANJA:

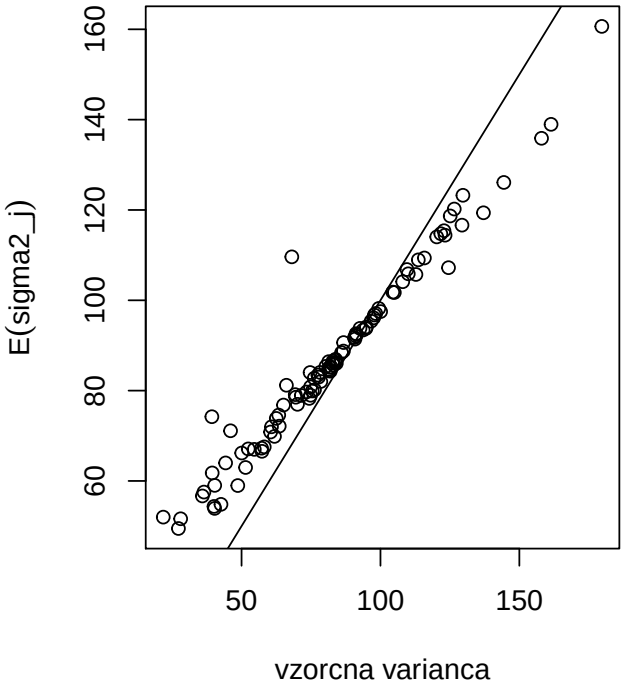

10. sola
njeno vz. povp.
njen E(apost)
18. sola
njeno vz. povp.
njen E(apost)
Density
0.00
0.05
0.10
0.15
0.20
35
40
45
50
N = 5000 Bandwidth = 0.3244

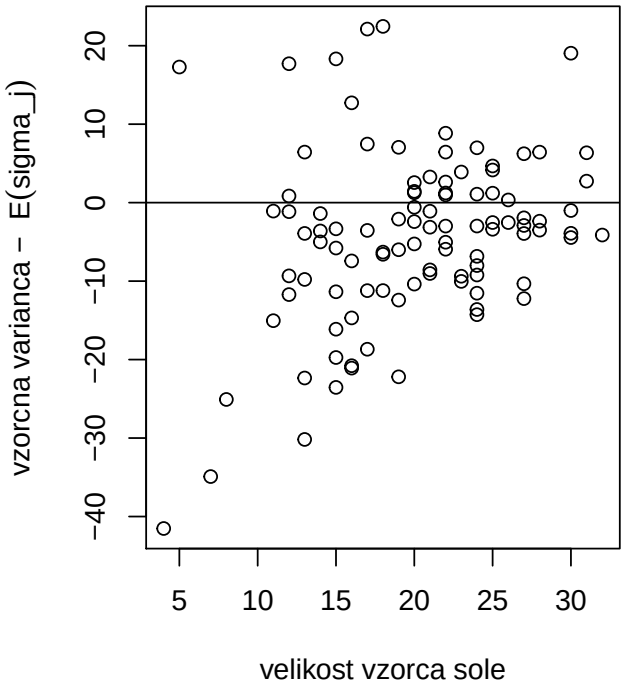

E(mu_j)
40
45
50
55
60
40
45
50
55
60
65
vzorcno povprecje


vzorcno povprecje − E(mu_j)
−4
−2
0
2
4
6
8
5
10
15
20
25
30
velikost vzorca sole

VARIANCE:


10. sola
njena vz. var.
njen E(apost)
18. sola
njena vz. var..
njen E(apost)
Density
0.000
0.005
0.010
0.015
0.020
50
100
150
200
250
N = 5000 Bandwidth = 3.606


E(sigma2_j)
60
80
100
120
140
160
50
100
150
vzorcna varianca


vzorcna varianca − E(sigma_j)
−40
−30
−20
−10
0
10
20
5
10
15
20
25
30
velikost vzorca sole

# 6 Primerjava

Rezultati so podobni rezultatom 6. sklopa, je pa vzorčenje z posplosenimi sigmami računsko veliko zahtevneje, zato se splača pretehtati, ce je posplosevanje smiselno.