

6. sklop: Primer hierarhicnega modela z Gibbsovim vzorcevalnikom

Nina Ruzic Gorenjec

Gradivo tega sklopa temelji na knjigi P. D. Hoff, *A First Course in Bayesian Statistical Methods*, 2009, in sicer na 8. poglavju *Group comparisons and hierarchical modeling*, str. 125.

1 Podatki

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

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

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

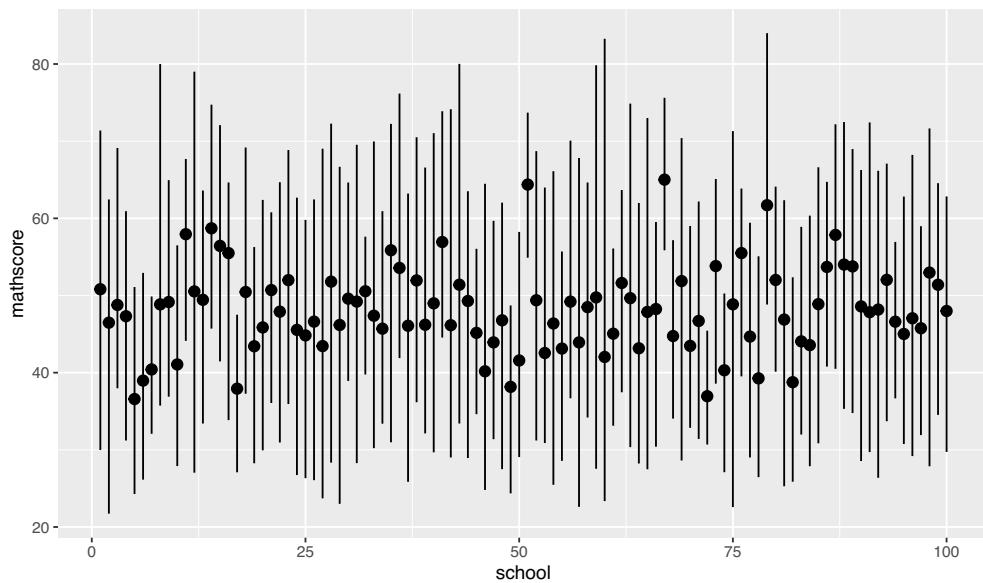
Oglejmo si podatke.

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

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

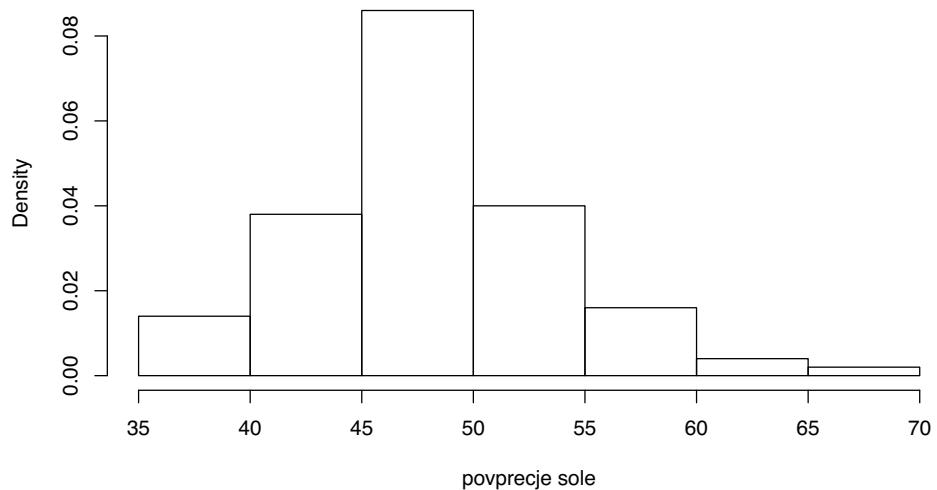
library(ggplot2)
ggplot(pod, aes(x = school, y = mathscore, group = school)) +
  stat_summary(fun.ymin = min, fun.ymax = max, fun.y = mean) +
  labs(title = "Povprecja (pika) in razpon rezultatov po solah")
```

Povprecja (pika) in razpon rezultatov po solah

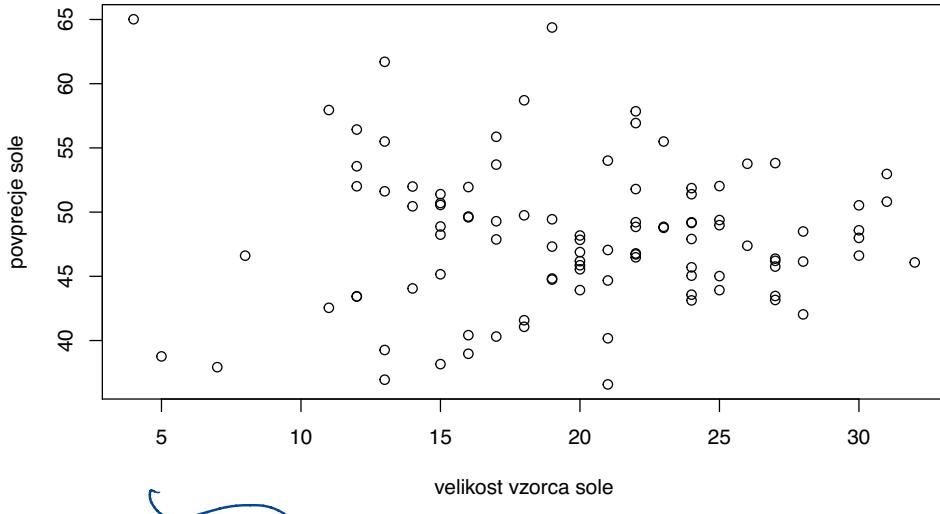


```
library(dplyr)
pod.sole = pod %>%
  group_by(school) %>%
  summarise(povprecje = mean(mathscore), n=length(mathscore), varianca = var(mathscore))

hist(pod.sole$povprecje, prob=T,
      xlab = "povprecje sole", main = "")
```



```
plot(pod.sole$n, pod.sole$povprecje,
      xlab = "velikost vzorca sole", ylab = "povprecje sole")
```



2 Model

2.1 Splosna ideja hierarhicnega modela

Podatki X_{ij} , $i \in \{1, 2, \dots, n_j\}$, $j \in \{1, 2, \dots, m\}$ (m skupin, velikost posamezne skupine n_j).

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

$$(X_{1,j}, \dots, X_{n_j,j}) \mid \theta_j \sim \text{n.e.p. } f(x \mid \theta_j)$$

Variabilnost med skupinami (*between-group sampling variability*) – model in apriorne porazdelitve θ_j :

$$(\theta_1, \dots, \theta_m) \mid \phi \sim \text{n.e.p. } f(\theta \mid \phi)$$

Bistvo hierarhicnega modela: podatki so odvisni od parametrov le preko θ_j in ne tudi ϕ (tj. $f(x \mid \theta_j, \phi) = f(x \mid \theta_j)$), θ_j pa so realizacija vzorca (tj. n.e.p.) iz neke skupne porazdelitve s hiperparametrom. Okrajsava: n.e.p. = neodvisne enako porazdeljene (iid).

Apriorna, oziroma natancneje *hiperapriorna* porazdelitev za ϕ :

$$\phi \sim \pi(\phi)$$

V modelu imamo *parametre* $\theta_1, \dots, \theta_m$ (vsak je lahko vektor) in *hiperparametre* ϕ (lahko vektor).

2.2 Hierarhici normalni model

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

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

Variabilnost med skupinami (*between-group sampling variability*) – model in apriorne porazdelitve μ_j :

$$(\mu_1, \dots, \mu_m) \mid \mu, \eta^2 \sim \text{n.e.p. } N(\mu, \eta^2)$$

Privzeli smo, da so vse variance znotraj skupin enake (σ^2).

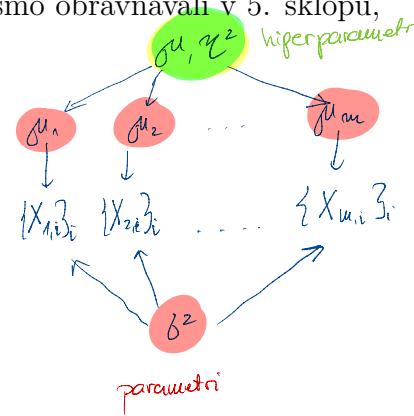
Potrebujemo torej se (hiper)apriorno porazdelitev za (μ, η^2) in apriorno proazdelitev σ^2 .

Izberemo si:

$$\sigma^2 \sim \text{Inv-Gama}(\nu_0/2, \sigma_0^2 \nu_0/2).$$

Za hiperapriorno porazdelitev vzamemo “polkonjugirano” (te nismo obravnavali v 5. sklopu, glejte 6. poglavje v knjigi Hoff):

$$\begin{aligned} \pi(\mu, \eta^2) &= \pi(\mu) \pi(\eta^2), \\ \mu &\sim \mathcal{N}(\mu_0, \tau_0^2), \\ \eta^2 &\sim \text{Inv-Gama}(\kappa_0/2, \eta_0^2 \kappa_0/2). \end{aligned}$$



Zapisali smo matematični model.

Za boljso predstavo si narisemo graficni model (vaje).

2.3 Apriorna porazdelitev – natancneje

Imamo torej naslednjo vecrazsezno apriorno porazdelitev (upostevamo zgornje izbore gostot in kateri parametri so neodvisni):

$$\begin{aligned} \pi(\mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2) &= \pi(\mu_1, \dots, \mu_m \mid \mu, \eta^2) \cdot \pi(\sigma^2) \cdot \pi(\mu, \eta^2) \\ &= \left(\prod_{j=1}^m \pi(\mu_j \mid \mu, \eta^2) \right) \cdot \pi(\sigma^2) \cdot \pi(\mu) \cdot \pi(\eta^2) \\ &= \left(\prod_{j=1}^m f_{\text{Normal}}(\mu_j \mid \mu, \eta^2) \right) \cdot f_{\text{Inv-Gama}}(\sigma^2 \mid \sigma_0^2, \nu_0) \cdot f_{\text{Normal}}(\mu \mid \mu_0, \tau_0^2) \\ &\quad \cdot f_{\text{Inv-Gama}}(\eta^2 \mid \eta_0^2, \kappa_0), \end{aligned}$$

kjer so $\mu_0, \tau_0^2, \sigma_0^2, \nu_0, \eta_0^2, \kappa_0$ konstante.

2.4 Aposteriorna porazdelitev

Zelimo vzorciti iz aposteriorne porazdelitve

$$\pi(\mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2 \mid [x_{ij}]_{ij}) = \pi(\mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2 \mid \mathbf{x}_1, \dots, \mathbf{x}_m).$$

Ne bomo izpeljali/podali cele vecrazsezne gostote (kot v sklopih 1-3 in 5), uporabili bomo Gibbsov vzorcevalnik.

2.5 Gibbsov vzorcevalnik – splosno

Vzorciti zelimo iz aposteriorne porazdelitve

$$\pi(\theta, \phi | x).$$

Za Gibbsov algoritem moramo poznati obe pogojni aposteriorni porazdelitvi:

$$\begin{aligned} \pi(\theta | \phi, x), \\ \pi(\phi | \theta, x). \end{aligned}$$

↓ potrebuje eno
 ↓ izpeljati

Na koraku s Gibbsovega vzorcevalnika smo dobili $(\theta^{(s)}, \phi^{(s)})$. Naslednji korak:

1. Vzorcimo $\theta^{(s+1)} \sim \pi(\theta | \phi^{(s)}, x)$.
2. Vzorcimo $\phi^{(s+1)} \sim \pi(\phi | \theta^{(s+1)}, x)$.

2.6 Aposteriorna porazdelitev - nadaljevanje

Potrebujemo torej vse pogojne aposteriorne porazdelitve.

Najprej razpisimo aposteriorno porazdelitev, podobno kakor smo apriorno:

$$\begin{aligned} & \pi(\mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2 | \mathbf{x}_1, \dots, \mathbf{x}_m) \\ \xrightarrow{\substack{\text{Bayesova} \\ \text{formula}}} & \propto \pi(\mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2) \cdot f(\mathbf{x}_1, \dots, \mathbf{x}_m | \mu_1, \dots, \mu_m, \mu, \eta^2, \sigma^2) \\ \xrightarrow{\substack{\text{fizikalna} \\ \text{vez}} & \text{formula}} & = \pi(\mu, \eta^2, \sigma^2) \cdot \pi(\mu_1, \dots, \mu_m | \mu, \eta^2, \sigma^2) \cdot f(\mathbf{x}_1, \dots, \mathbf{x}_m | \mu_1, \dots, \mu_m, \sigma^2) \\ & = \pi(\mu) \cdot \pi(\eta^2) \cdot \pi(\sigma^2) \cdot \left[\prod_{j=1}^m \pi(\mu_j | \mu, \eta^2) \right] \cdot \left[\prod_{j=1}^m \left(\prod_{i=1}^{n_j} f(x_{i,j} | \mu_j, \sigma^2) \right) \right]. \end{aligned}$$

Velja torej:

$$\begin{aligned} \pi(\mu | \mu_1, \dots, \mu_m, \eta^2, \sigma^2, \mathbf{x}_1, \dots, \mathbf{x}_m) & \propto \pi(\mu) \prod_{j=1}^m \pi(\mu_j | \mu, \eta^2) \\ & = f_{\text{Normal}}(\mu | \mu_0, \tau_0^2) \cdot \left(\prod_{j=1}^m f_{\text{Normal}}(\mu_j | \mu, \eta^2) \right). \end{aligned}$$

Iz tega se ne znamo vzorciti, izracunati moramo gostoto. Opazimo lahko, da je to aposteriorna porazdelitev za parameter μ iz normalnega modela z znano varianco η^2 , kjer ima (μ_1, \dots, μ_m) vlogo vzorca velikosti m , in ima μ konjugirano apriorno porazdelitev $N(\mu_0, \tau_0^2)$. Po formulah iz 3. sklopa (razdelek 4.3, str. 5) je torej

$$\mu | \text{vse ostalo} \sim N \left(\frac{\bar{\mu}m/\eta^2 + \mu_0/\tau_0^2}{m/\eta^2 + 1/\tau_0^2}, \left[m/\eta^2 + 1/\tau_0^2 \right]^{-1} \right).$$

↓ μ_0 ↓ m ↓ η^2 ↓ parametra aposteriorne
 ↓ 3 sklop

Podobno lahko izpeljemo se preostale pogojne aposteriorne porazdelitve, kjer uporabimo že znano iz (1) normalnega modela z znano varianco in konjugirano apriorno porazdelitvijo za povprecje; in (2) dvoparametricnega normalnega modela s polkonjugirano apriorno porazdelitvijo (v 5. sklopu smo imeli konjugirano, vec o polkonjugirani pa v 6. poglavju knjige Hoff). Dobimo:

$$\begin{aligned}\eta^2 \mid \text{vse ostalo} &\sim \text{Inv-Gama} \left(\frac{\kappa_0 + m}{2}, \frac{\kappa_0 \eta_0^2 + \sum_{j=1}^m (\mu_j - \mu)^2}{2} \right) \\ \mu_j \mid \text{vse ostalo} &\sim N \left(\frac{\bar{x}_{\cdot j} n_j / \sigma^2 + \mu / \eta^2}{n_j / \sigma^2 + 1 / \eta^2}, [n_j / \sigma^2 + 1 / \eta^2]^{-1} \right) \\ \sigma^2 \mid \text{vse ostalo} &\sim \text{Inv-Gama} \left(\frac{\nu_0 + \sum_{j=1}^m n_j}{2}, \frac{\nu_0 \sigma_0^2 + \sum_{j=1}^m \sum_{i=1}^{n_j} (x_{i,j} - \mu_j)^2}{2} \right)\end{aligned}$$

3 Uporaba modela na podatkih

Dolociti moramo se parametre (hiper)apriornih porazdelitev $\mu_0, \tau_0^2, \sigma_0^2, \nu_0, \eta_0^2, \kappa_0$:

$$\begin{aligned}\sigma^2 &\sim \text{Inv-Gama}(\nu_0/2, \sigma_0^2 \nu_0/2), \\ \mu &\sim \mathcal{N}(\mu_0, \tau_0^2), \\ \eta^2 &\sim \text{Inv-Gama}(\kappa_0/2, \eta_0^2 \kappa_0/2).\end{aligned}$$

Spomnimo se, da je matematični test konstruiran tako, da je pricakovana vrednost enaka 50 in standardni odklon 10.

Izberemo si:

$\sigma_0^2 = 100$ (ker naj bi bil standardni odklon 10), $\nu_0 = 1$ (sibko informativna),

~~$\mu_0 = 50$ (ker naj bi bil standardni odklon 10),~~ *bila porazvana vrednost 50*

$\tau_0^2 = 25$ (ker zelimo sibko informativno in s tem dopuscamo s 95% μ med 40 in 60),

$\eta_0^2 = 100$, $\kappa_0 = 1$.

3.1 Gibbsov vzorcevalnik za nas primer

```
### Parametri (hiper)apriornih porazdelitev
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$povprecje
    (mu1, ..., mun)
### Dolocimo si zacetne vrednosti
muGroups = x.povpr
sigma2 = mean(pod.sole$varianca)
mu = mean(muGroups)
eta2 = var(muGroups)
    nest parametri
### Pripravimo si prostor za shranjevanje
n.iter = 5000

muGroups.all = matrix(nrow = n.iter, ncol = m)
sigma2.all = rep(NA, n.iter)
mu.all = rep(NA, n.iter)
eta2.all = rep(NA, n.iter)
```

```

### Na prvo mesto si shranimo zacetne vrednosti (nepotrebno)
muGroups.all[1, ] = muGroups
sigma2.all[1] = sigma2
mu.all[1] = mu
eta2.all[1] = eta2

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

  ### Vzorcimo sigma2
  ss = nu0 * sigma20
  for (j in 1:m) {
    ss = ss + sum((x[, 1] == j, 2] - muGroups[j])^2)
  }
  sigma2 = 1 / rgamma(1, (nu0 + sum(n)) / 2, ss / 2)

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

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

  ### Shranimo nove parametre
  muGroups.all[s, ] = muGroups
  sigma2.all[s] = sigma2
  mu.all[s] = mu
  eta2.all[s] = eta2
}

```

→ Že uporabljen na novo izracunam (glej Gibbsov algortitem)

3.2 Preucevanje konvergencije

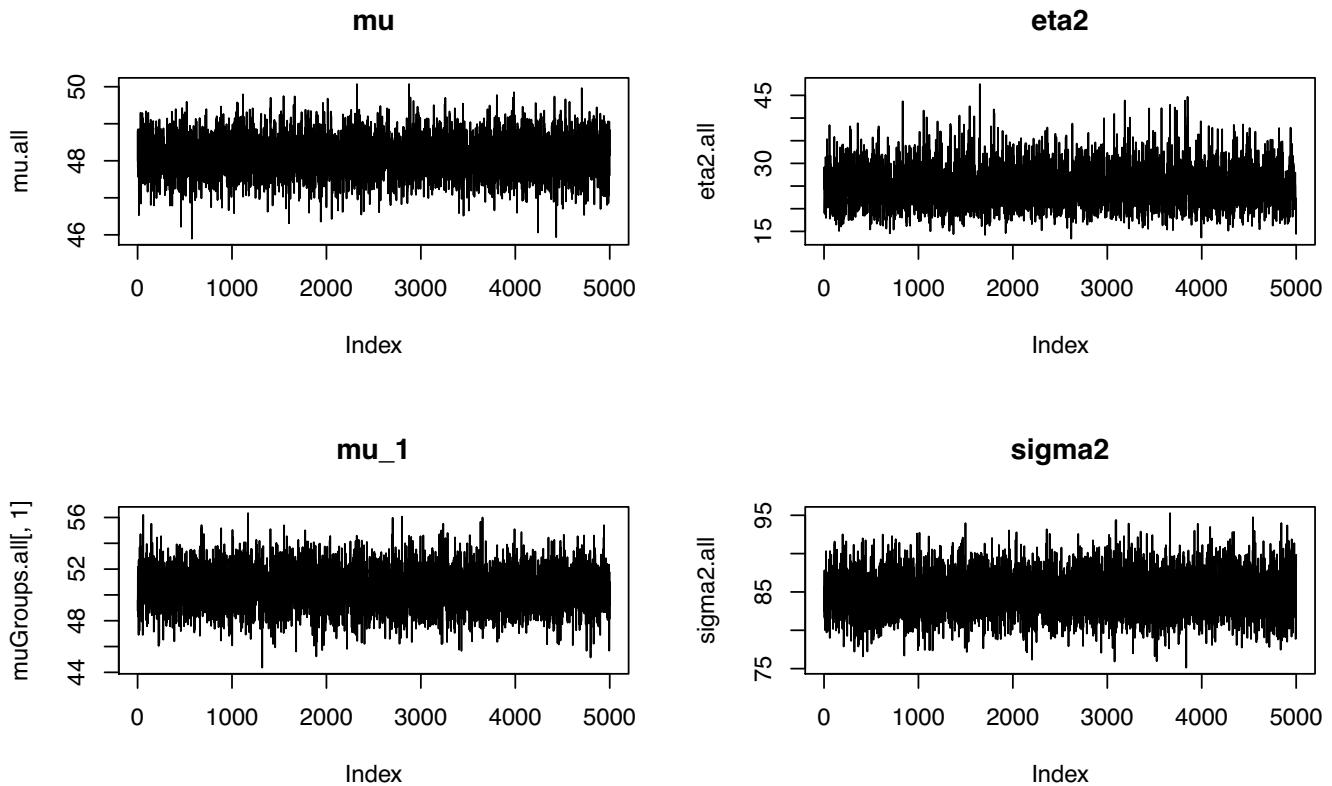
V podrazdelkih so navedeni načini za preucevanje konvergencije.

3.2.1 Trace plots

Na podlagi spodnjih slik verig dolocimo potreben *burn-in* in graficno presodimo, ali je konvergenca dosezena (tako kot v 4. sklopu).

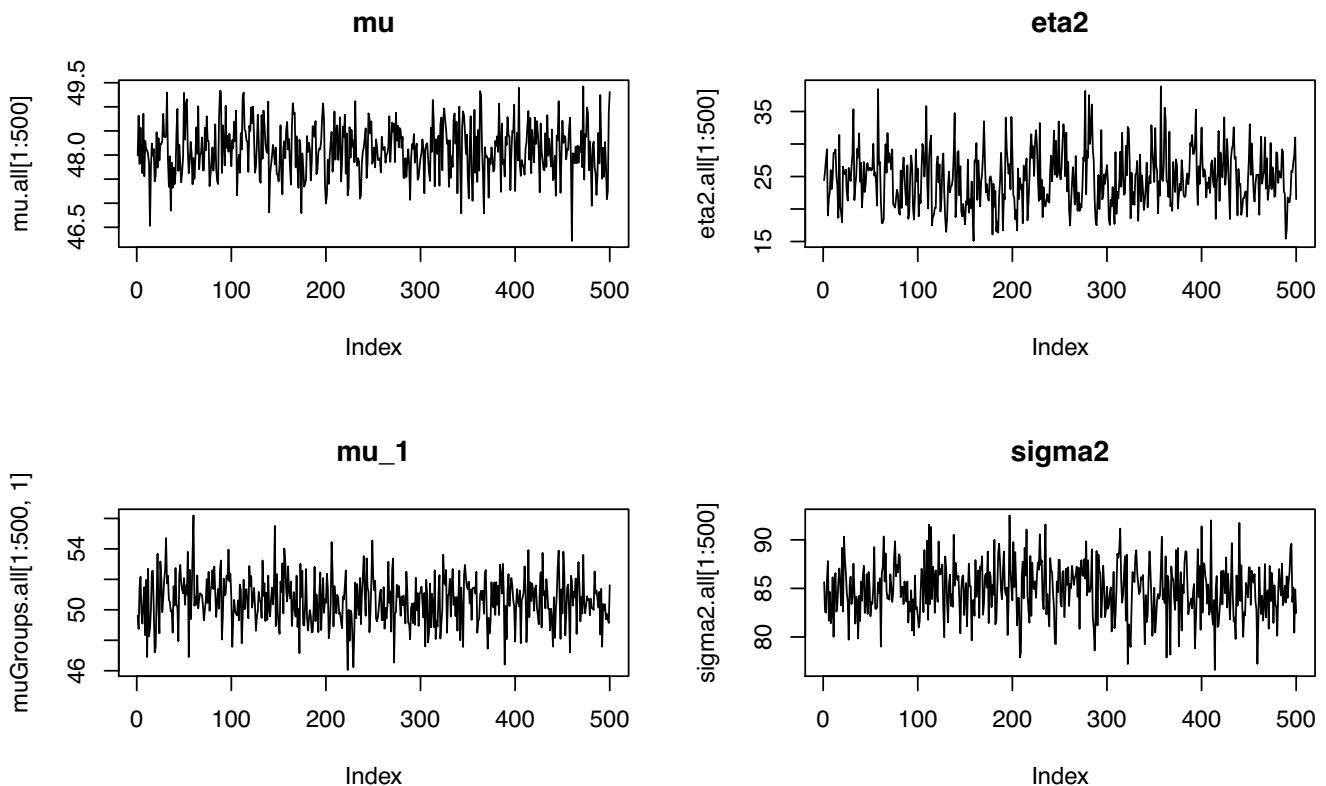
Vse iteracije (izgleda v redu):

```
par(mfrow=c(2,2))
plot(mu.all, type="l", main="mu")
plot(eta2.all, type="l", main="eta2")
plot(muGroups.all[,1], type="l", main="mu_1")
plot(sigma2.all, type="l", main="sigma2")
```



Prvih 500 iteracij (izgleda v redu):

```
par(mfrow=c(2,2))
plot(mu.all[1:500], type="l", main="mu")
plot(eta2.all[1:500], type="l", main="eta2")
plot(muGroups.all[1:500,1], type="l", main="mu_1")
plot(sigma2.all[1:500], type="l", main="sigma2")
```



3.2.2 Vec verig in *mixing*

Preizkusimo razlicne zacetne vrednosti (kljucno je, da vkljucimo tudi manj smiselne) in pogledamo, ali dobimo po nekem zacetnem *burn-in* podobno verigo. Verigi lahko narisemo na eno sliko in pogledamo, ali je dober *mixing*. V koncni vzorec potem zdruzimo vse verige.

Tukaj analizo vecih verig izpustimo (to smo podrobnejše preucevali ze pri algoritmu Metropolis-Hastings).

3.2.3 Porazdelitve podvzorcev

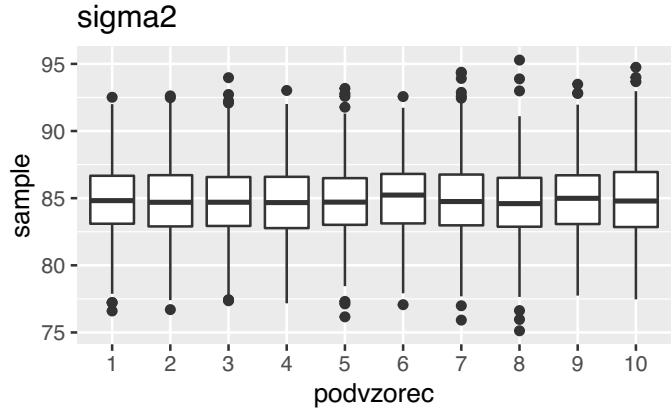
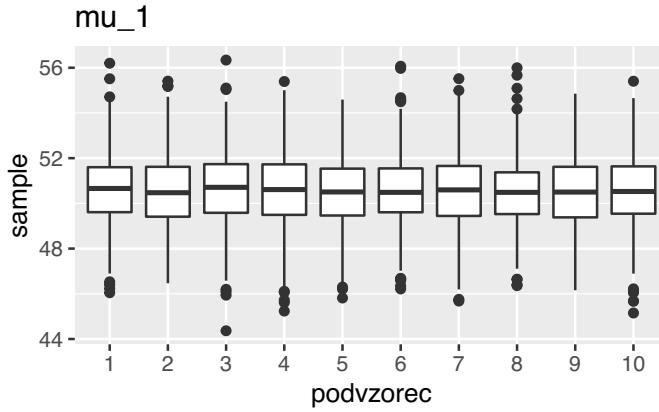
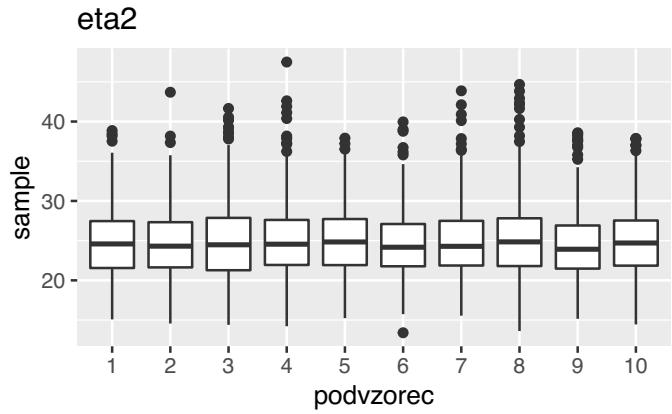
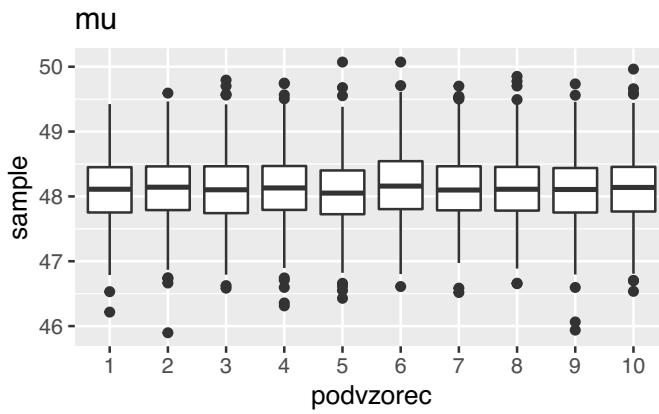
Pogledamo, ali se priblizno ujemajo porazdelitve po zaporednih odsekih, tj. ali je veriga "stabilna" (izgleda v redu).

```
library(gridExtra)
mu.all2 = data.frame(sample = mu.all, podvzorec = factor(sort(rep(1:10,500))))
p1 = ggplot(mu.all2, aes(x = podvzorec, y = sample)) +
  geom_boxplot() + labs(title = "mu")

eta2.all2 = data.frame(sample = eta2.all, podvzorec = factor(sort(rep(1:10,500))))
p2 = ggplot(eta2.all2, aes(x = podvzorec, y = sample)) +
  geom_boxplot() + labs(title = "eta2")

mu1 = data.frame(sample = muGroups.all[,1], podvzorec = factor(sort(rep(1:10,500))))
p3 = ggplot(mu1, aes(x = podvzorec, y = sample)) +
  geom_boxplot() + labs(title = "mu_1")

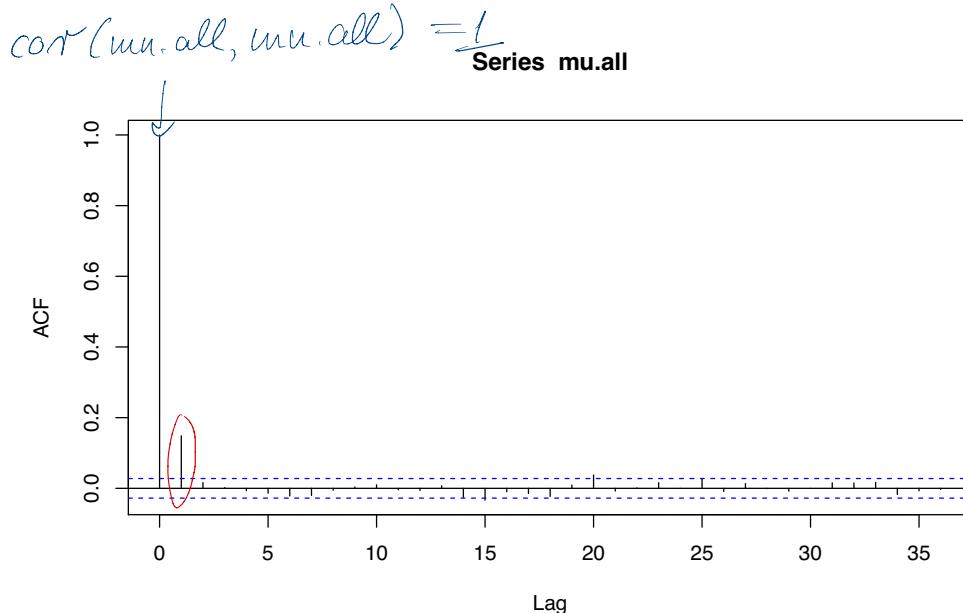
sigma2.all2 = data.frame(sample = sigma2.all, podvzorec = factor(sort(rep(1:10,500))))
p4 = ggplot(sigma2.all2, aes(x = podvzorec, y = sample)) +
  geom_boxplot() + labs(title = "sigma2")
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



3.2.4 Avtokorelacijske in effective sample size VS thinning

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*) zmanjšuje (odvisnost neke vrednosti od vrednosti iz enega koraka nazaj je najvecja, iz dveh korakov nazaj malo manjsa, itd.).

```
acf(mu.all) #dobimo graf, cel vzorec
```



```
ac.mu = acf(mu.all, plot = FALSE)
ac.mu #dobimo izpis
```

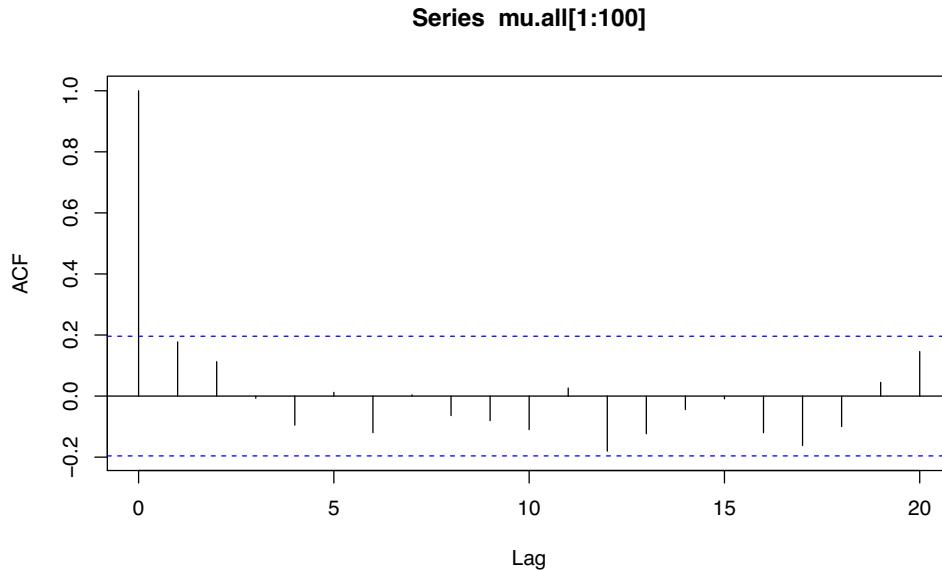
```
##
## Autocorrelations of series 'mu.all', by lag
##
##      0      1      2      3      4      5      6      7      8      9      10 
##  1.000  0.149  0.016  0.002 -0.007 -0.014 -0.022 -0.020 -0.005  0.004  0.010 
##     11     12     13     14     15     16     17     18     19     20     21 
## -0.004 -0.006  0.007 -0.025 -0.033 -0.010 -0.015 -0.023  0.009  0.038 -0.008 
##     22     23     24     25     26     27     28     29     30     31     32 
##  0.001  0.016  0.000  0.026 -0.009  0.014 -0.001 -0.008 -0.001  0.016  0.015 
##     33     34     35     36 
##  0.017 -0.017 -0.007 -0.003

head(ac.mu$acf) #dobimo natancnejsi izpis
```

```
## [1] 1.000000000 0.148547488 0.016374954 0.001949737 -0.007071617
## [6] -0.013949257
```

Kako lahko priblizno izracunamo avtokorelacijo z zamikom 1?

$\left[\mu_{all_1}, \dots, \mu_{all_{4999}} \right]$ $\left[\mu_{all_2}, \dots, \mu_{all_{5000}} \right]$
 cor(mu.all[-length(mu.all)], mu.all[-1])
 ## [1] 0.1485619
 ac.mu\$acf[2]
 ## [1] 0.1485475
 acf(mu.all[1:100]) #za prvih 100 iteracij



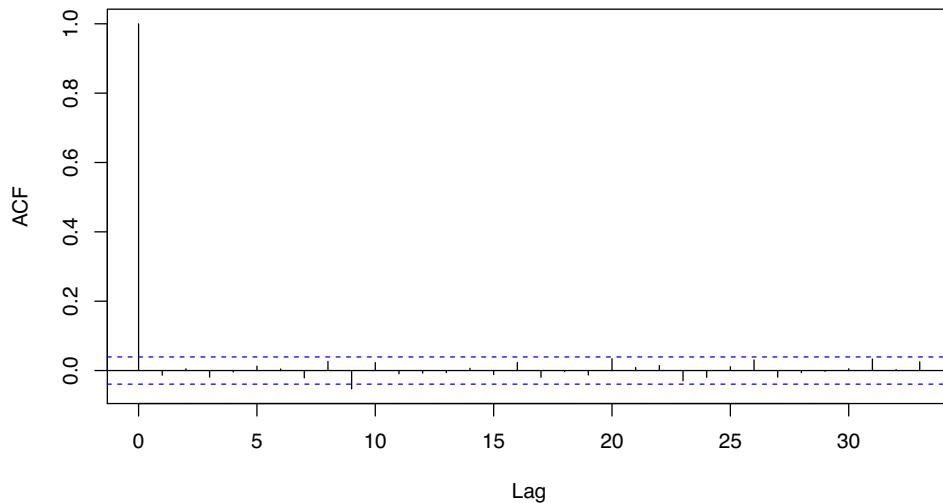
Kaj predstavlja crti?

$\pm 1.96/\sqrt{N}$, N stevilo iteracij – avtokorelacijske izven obmocja so statistično značilno različne 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*, glejte 4. sklop)?

```
acf(mu.all[seq(1, length(mu.all), by=2)])
```

Series mu.all[seq(1, length(mu.all), by = 2)]



Pricakovano se zmanjsajo, vendar smo pri tem zmanjsali tudi velikost vzorca, za katerega smo ze porabili cas za izracun.

Effective sample size lahko interpretiramo kakor stevilo slučajnih vzorcev (n.e.p.), ki nam dajo enako natancno informacijo kakor nas dobljeni MCMC vzorec.

```
library(coda)
effectiveSize(mu.all)

##      var1
## 3705.908
```

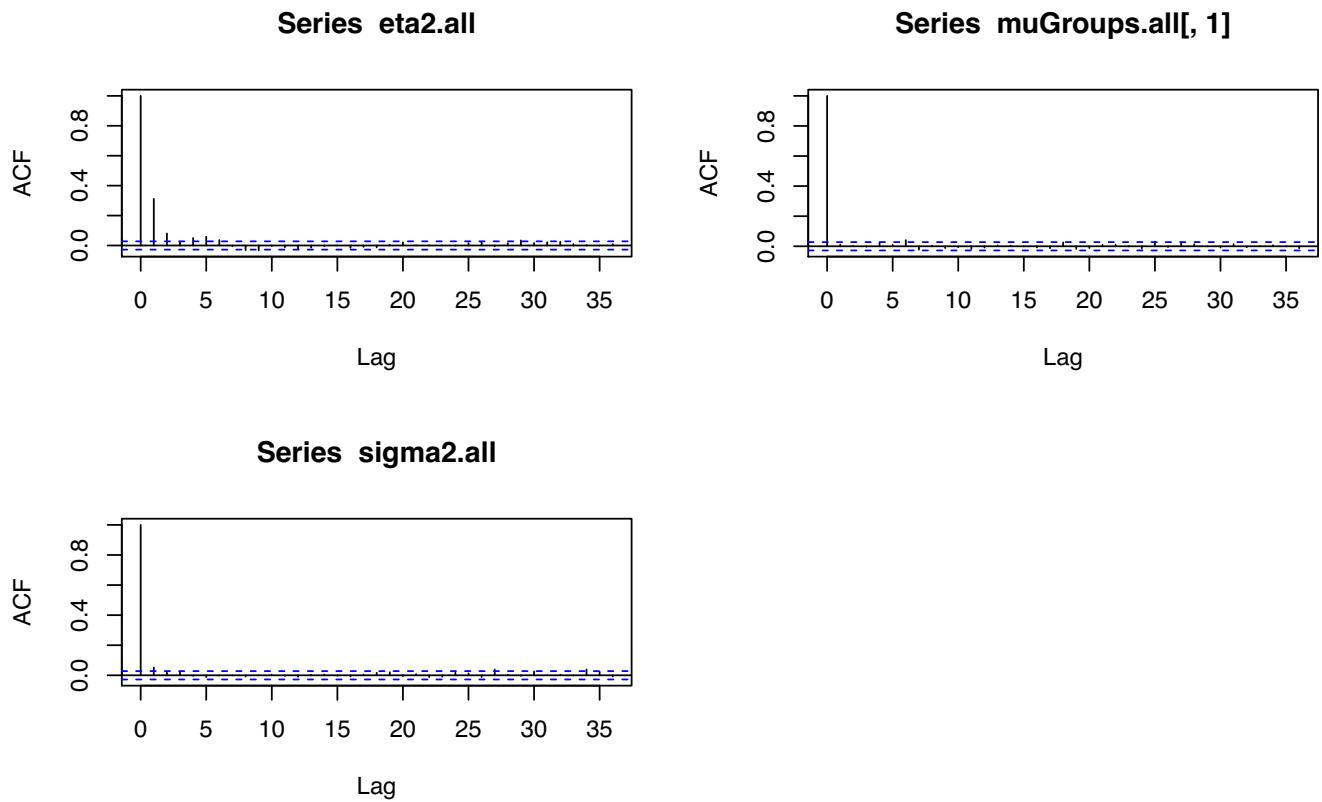
Torej:

- Celoten vzorec za μ dolzine 5000 je avtokoreliran – ne smemo ga imeti za slučajni vzorec (n.e.p.) iz aposteriorne porazdelitve.
- Ce v verigo vzamemo vsakega drugega (*thinning* = 2, kar je najmanjse možno), potem vzorec ni avtokoreliran, njegova velikost pa je 2500.
- *Effective sample size*, ki pove za koliko n.e.p. realizacij je nas vzorec “vreden”, je enak 3705, kar je mnogo vec od 2500. Uporabimo zato raje celoten vzorec in upostevamo *effective sample size* kot njegovo “pravo velikost”.

→ oz. lahko uporabimo še karzen drug sampler
 oz. model sprememimo
 nimde stan
 (zadnje vaje)
 računata v C++ → rabiti Rtools

Avtokorelacijske preostalih (nekaj) parametrov in njihovi *effective sample size*:

```
par(mfrow=c(2,2))
acf(eta2.all)
acf(muGroups.all[,1])
acf(sigma2.all)
```



```
effectiveSize(eta2.all)
```

```
##      var1
## 2503.468
```

```
effectiveSize(muGroups.all[,1])
```

```
##      var1
## 4421.197
```

```
effectiveSize(sigma2.all)
```

```
##      var1
## 4498.545
```



$$SE_{\hat{\theta}} = \text{sd}(\text{apost. za } \theta) / \sqrt{N}$$

eff.
st. meracij → sample
size

Preko *effective sample size* izracunamo "standardne napake" nasih ocen, kjer je ocena povprecje marginalne aposteriorne porazdelitve:

```
sd(mu.all) / sqrt( effectiveSize(mu.all) )

##           var1
## 0.008795844

sd(eta2.all) / sqrt( effectiveSize(eta2.all) )

##           var1
## 0.08842911

sd(muGroups.all[,1]) / sqrt( effectiveSize(muGroups.all[,1]) )

##           var1
## 0.02359033

sd(sigma2.all) / sqrt( effectiveSize(sigma2.all) )

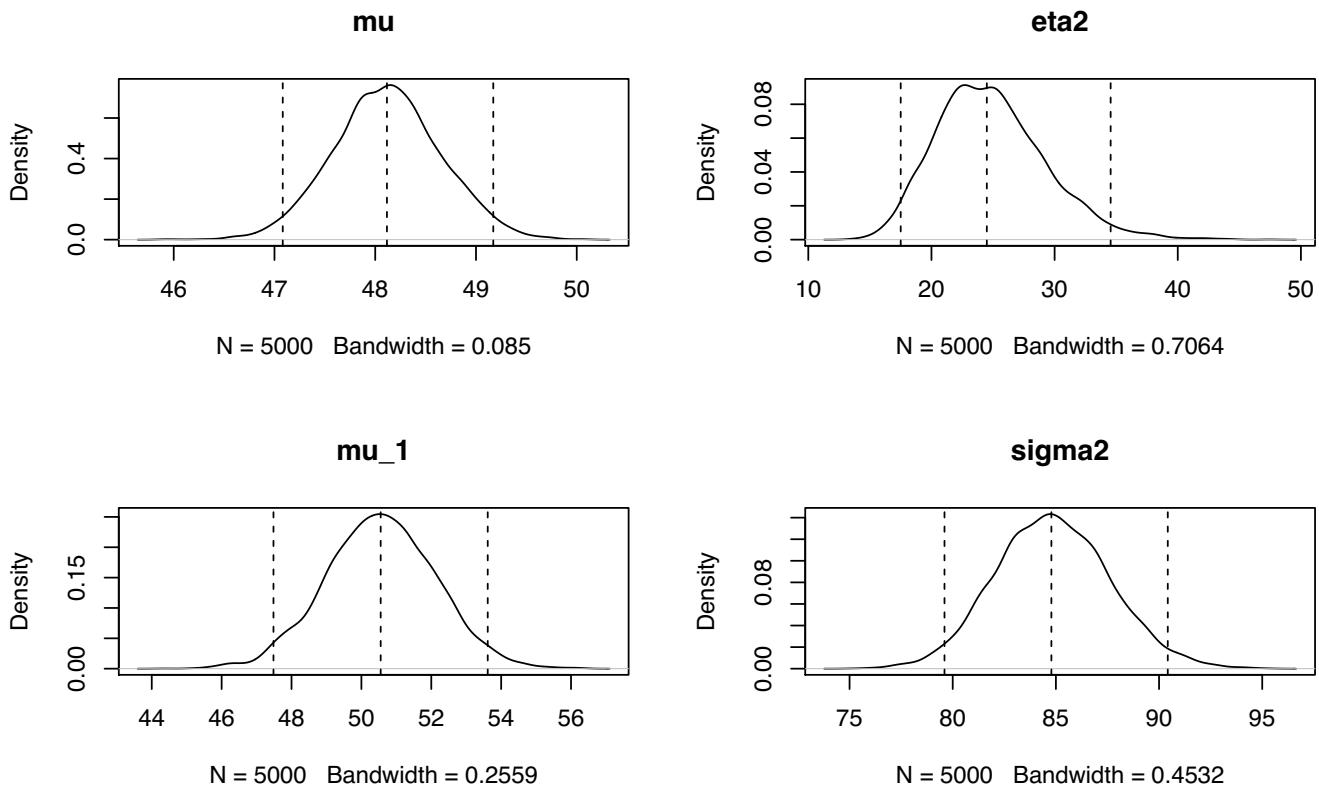
##           var1
## 0.04123463
```

Ali so velike ali majhne? Odvisno glede na skalo parametrov oz. enote spremenljivk.

Rezultati

3.3 Robne aposteriorne porazdelitve

```
par(mfrow=c(2,2))
plot(density(mu.all), type="l", main="mu")
abline(v = quantile(mu.all, prob=c(0.025, 0.5, 0.975)), lty = 2)
plot(density(eta2.all), type="l", main="eta2")
abline(v = quantile(eta2.all, prob=c(0.025, 0.5, 0.975)), lty = 2)
plot(density(muGroups.all[,1]), type="l", main="mu_1")
abline(v = quantile(muGroups.all[,1], prob=c(0.025, 0.5, 0.975)), lty = 2)
plot(density(sigma2.all), type="l", main="sigma2")
abline(v = quantile(sigma2.all, prob=c(0.025, 0.5, 0.975)), lty = 2)
```



7 Interpretirati rezultate (srednja vrednost in razprsenost porazdelitve sta pomembni) tako za hiperparametre, kakor tudi za posamezne sole (katera ima najboljse, katera najslabse rezultate, ali se parametri "statisticno znacilno" razlikujejo, ipd.).

cepstevanje + seminarski material
potem + seminarski material

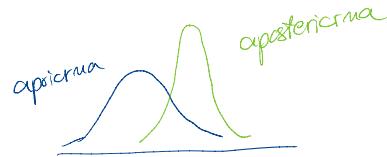
preverimo, da vrednosti rezultator glede na izvir apriorne



Preveriti odnos med aposterornimi in apriorimi porazdelitvami (hiper)parametrov

Mozne vrednosti aposteriorne morajo biti vsebovane v tistih od apriorne, ce:

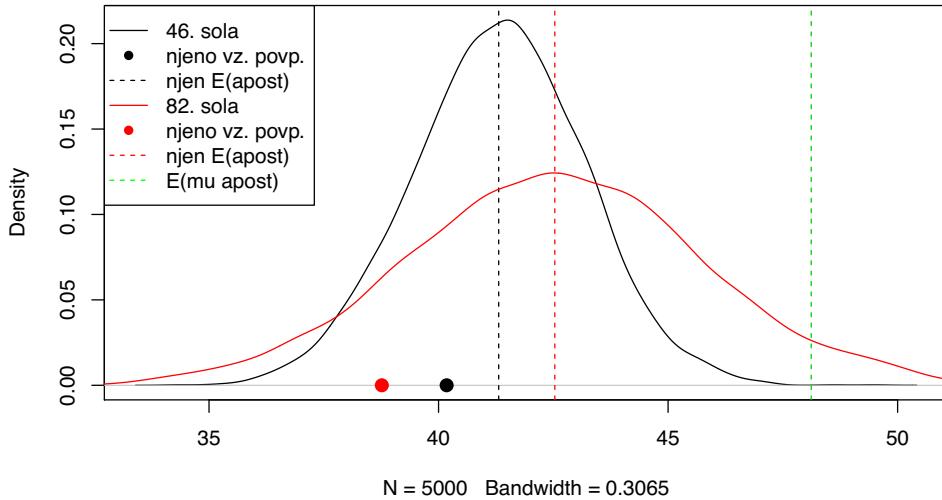
- Zelimo imeti objektiven Bayesov pristop (tj. sibke apriorne, ki ne smejo informirati aposteriornih).
- Si slednje lahko privoscimo glede na stevilo enot oz. imamo dovolj podatkov, ki so zmožni informirati nase parametre (ce si ne moremo, je verjetno modeliranje parametra kljub odstopanjem aposteriorne od apriorne se zmeraj boljse kakor fiksiranje na neko vrednost, kar nekako ustreza apriorni z gostoto 1 v tej vrednosti).
- Ni namen modela drugacen, npr. skrciti koeficiente v regresiji proti nic.



3.3.1 Shrinkage

Oglejmo si aposteriorni porazdelitvi dveh sol:

```
plot(density(muGroups.all[,46]), type="l", main="")
points(pod.sole[46,]$povprecje, 0, pch=16, cex=1.5)
abline(v = mean(muGroups.all[,46]), lty=2)
lines(density(muGroups.all[,82]), type="l", col="red")
points(pod.sole[82,]$povprecje, 0, pch=16, cex=1.5, col="red")
abline(v = mean(muGroups.all[,82]), lty=2, col="red")
abline(v = mean(mu.all), lty=2, col="green3")
legend("topleft", c("46. sola", "njeno vz. povp.", "njen E(apost)",
                     "82. sola", "njeno vz. povp.", "njen E(apost)",
                     "E(mu apost)"),
       col=c("black", "black", "black", "red", "red", "green"), lty=c(1,NA,2,1,NA,2,2),
       pch=c(NA,16,NA,NA,16,NA))
```



Navidez paradoksalno, saj je vzorcno povprecje 82. sole manjše od 46., medtem ko je pricakovana vrednosti aposteriorne porazdelitve velja ravno obratno.

Iz pogojne porazdelitve μ_j glede na vse ostalo vemo, da je:

$$E(\mu_j \mid \text{vse ostalo}) = \frac{n_j/\sigma^2}{n_j/\sigma^2 + 1/\eta^2} \bar{x}_{.j} + \frac{1/\eta^2}{n_j/\sigma^2 + 1/\eta^2} \mu$$

Pricakovana vrednost μ_j se torej pomakne proti skupnemu populacijskemu povprecju μ .

Ta efekt je pri velikem n_j majhen.

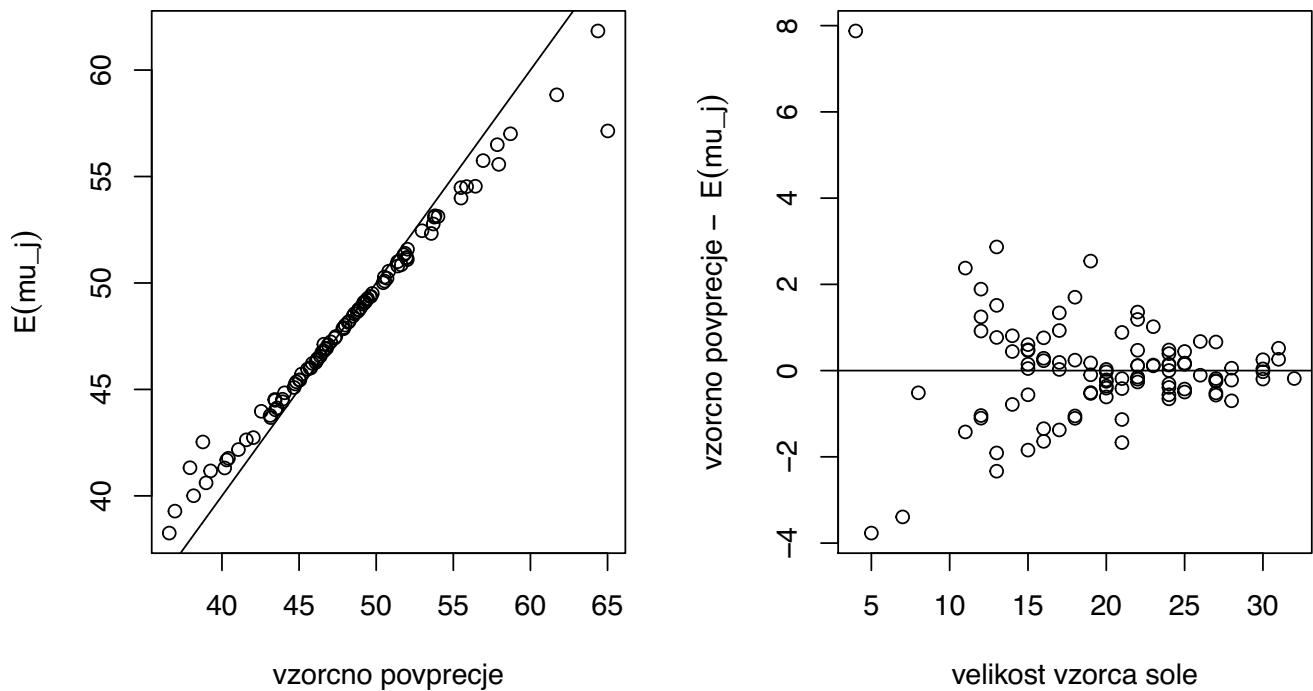
Poanta (zaradi katere zgornje ni paradoksalno): Pri soli z majhnim vzorcem smo lahko bolj zgresili povprecje te sole in zato bolj verjamemo skupnemu povprecju μ kot njenemu vzorcemu povprecju oz. se k μ premaknemo bolj kot pri veliki soli.

Ocene povprecij sol (pricakovane vrednosti aposterironih porazdelitev μ_j) se torej skrcijo (*shrinkage*) proti skupnemu povprecju μ , v primerjavi z vzorcimi povprecji sol $\bar{x}_{\cdot j}$, kjer je ta premik vecji pri manjsi velikosti vzorca sole n_j . Slednje je jasno prikazano spodaj.

```
pod.sole$EmuGroups = colMeans(muGroups.all)

par(mfrow=c(1,2))
plot(pod.sole$povprecje, pod.sole$EmuGroups,
      xlab = "vzorcno povprecje", ylab = expression(E(mu_j)))
abline(a = 0, b = 1)

plot(pod.sole$n, pod.sole$povprecje - pod.sole$EmuGroups,
      xlab = "velikost vzorca sole",
      ylab = expression(paste("vzorcno povprecje - ", " ", E(mu_j), sep="")))
abline(h = 0)
```



4 O ideji hierarhicnih modelov na splošno...

... vendar ob razmisljanju na nasem primeru.

Zamenljivost (exchangeability)

Znotraj modela smo predpostavili, da so tako meritve znotraj sol $X_{1,j}, \dots, X_{n_j,j}$ zamenljive (tj. slučajni vzorec, torej standardna predpostavka), kakor tudi populacijska povprecja sol μ_1, \dots, μ_m . Ali je to smiselno?

*Da, ker nimamo drugih informacij.
Sicer bi lahko uporabili mnogo drugih sprememb (izbran staršev, inace,...).*

Nismo pa predpostavili, da so vse meritve $X_{1,1}, \dots, X_{n_1,1}, \dots, X_{m,1}, \dots, X_{n_m,m}$ med seboj zamenljive. Zakaj? *Ker nemamo, da pripadajo različnim skupinam (imajo različne učitelje),
zar lahko uporabim rezultat preverjanja znanija*

Torej: manj potrebuje, vendar sicer je predpostava o zamenljivosti ↗

Kaj so prednosti/slabosti hierarhicnega modela v primerjavi z modelom z le enim skupnim povprecjem?

Primerjamo torej z modelom:

$$(X_{1,1}, \dots, X_{n_1,1}, \dots, X_{m,1}, \dots, X_{n_m,m}) \mid \mu, \sigma^2 \sim \text{n.e.p. } N(\mu, \sigma^2),$$

tj. dvorazsežni normalni model, kjer podatke vseh sol zdruzimo v en vzorec.

- ⊖ ni dovolj fleksibilen
- ⊖ nismo upoštevali informacije, ki nam je bila dana

Kaj so prednosti/slabosti hierarhicnega modela v primerjavi z modelom z vsemi razlicnimi nevezanimi povprecji?

Primerjamo torej z modelom:

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

kjer nadalje velja

$$\begin{aligned} \pi(\mu_1, \dots, \mu_m, \sigma^2) &= \pi(\mu_1, \dots, \mu_m) \cdot \pi(\sigma^2) \\ &= \pi(\mu_1) \cdot \dots \cdot \pi(\mu_m) \cdot \pi(\sigma^2). \end{aligned}$$

- ⊖ overfitting
- ⊖ veliko nevezanih parametrov → nismo reže skupine
- ⊖ nismo paramestra ju, ki predstavlja upr. poravnje na ravni države
(če mislimo zde na nenočne rezultate, kateri delite neki z močno domino zelo mimočutnem, da to seznamemo)

Pri močnem ščetku si v hierarhičnem modelu „izpostavimo“ informacije od vecjih skupin, ki so informirale hiperparametre.

5 Tretja domaca naloga

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

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

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

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

$$(\mu_1, \dots, \mu_m) \mid \mu, \eta^2 \sim \text{n.e.p. } N(\mu, \eta^2), \quad \text{tj. enako kot prej;} \\ (\sigma_1^2, \dots, \sigma_m^2) \mid \sigma_0^2, \nu_0 \sim \text{n.e.p. Inv-Gama}(\nu_0/2, \sigma_0^2 \nu_0/2).$$

Označ: *↑ naštev řešenj firma, anaz sta hiperparametri*

Hiperapriorne porazdelitve $\mu, \eta^2, \sigma_0^2, \nu_0$ naj bodo neodvisne. Hiperapriorna porazdelitev μ in η^2 naj bo enaka kot prej. Hiperapriorna porazdelitev za σ_0^2 naj bo:

$$\sigma_0^2 \sim \text{Gama}(a, b), \quad \text{kjer vzamemo } a = 1, b = 1/100;$$

- Hiperapriorna porazdelitev za ν_0 naj bo diskretna z vrednostmi $k \in \{1, 2, \dots\}$, za katero velja $P(\nu_0 = k) \propto e^{-\alpha k}$, kjer vzamemo $\alpha = 1$.

Za uporabo Gibbsovega vzorcevalnika potrebujemo pogojne porazdelitve (hiper)parametrov. Hiperparametra μ in η^2 imata enako pogojno porazdelitev kot prej. Pri pogojno porazdelitvi za μ_j moramo v prejsnji formuli nadomestiti σ^2 z σ_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 ν_0 lahko pokazemo, da je za vsak $k \in \{1, 2, \dots\}$

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

Iz te porazdelitve lahko vzorcimo, ce se omejimo na velik izbor $k \in \{1, 2, \dots, k_{\max}\}$, za te k izracunamo zgornji izraz in nato vzorcimo ν_0 iz mnozice $\{1, 2, \dots, k_{\max}\}$ z utezmi (1). To je bolje narediti na log skali in na koncu utezi “preskalirati”. V pomoc pri vzorcenju iz te porazdelitve vam je lahko naslednja koda v R:

```
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)))
```

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

V domaci nalogi sledite spodnjim korakom.

1. V porocilo vkljucite celotno kodo Gibbsovega vzorcevalnika, kjer ob kodi kratko komentirajte (ob kodi komentar zapisite za #), kje in kako ste obstojeco kodo spremenili.
2. Uporabite zadostno stevilo iteracij in ustrezen *burn-in*. Za dobljeni vzorec preucite konvergenco, tako da sledite korakom v teh navodilih (*trace plots*, porazdelitve podvzorcev, avtokorelacije, *effective sample size*). Slednje naredite za vse hiperparametre in nekaj “ne-hiper-parametrov”. Pri tem kratko komentirajte konvergenco.
3. Za vse hiperparametre in nekaj “ne-hiper-parametrov” narisite marginalne aposteriorne porazdelitve in izracunajte 95% Bayesovske intervale zaupanja. Kratko interpretirajte rezultate.
4. Graficno ovrednotite *shrinkage* efekt tako za povprecja kot tudi za variance, tako da narisete podobne slike kakor v razdelku 3.3.1. Rezultate kratko komentirajte.
5. Primerjajte rezultate obeh hierarhicnih modelov. Ali so rezultati za (hiper)parametre, ki opisujejo povprecja podobni? Razmislite, ali je uporaba splosnejsega hierarhicnega modela z razlicnimi variancami skupin potrebna pri teh podatkih? Oziroma, kdaj bi bila bolj potrebna in kdaj manj?

Namig: Ali ste algoritem pravilno popravili, lahko priblizno graficno preverite tako, da vase rezultate pri tockah 3 in 4 primerjate z rezultati iz knjige Hoff, str. 145-146.