P8 - MCMC metodoak (Metropolis-Hastings algoritmoa)

Probabilitatearekin lan egitean, askotan problema errealetan banaketa konplexuekin lan egin behar dugu. Are gehiago, kasu batzuetan banaketa horiek ez daude guztiz definituak eta, hortaz, ezin dira kalkulu analitikoak egin. Kasu horietan alternatiba interesgarri bat laginketen bidez hurbilketak egitea da. Adibidez, banaketa baten itxaropena estimatu nahi badugu, banaketa lagindu dezakegu eta balioen batazbestekoa hartu dezakegu itxaropenaren estimazio moduan.

Arazoa da, ideia hori praktikan jartzeko, interesatzen zaigun banaketa lagintzeko gai izan behar garela. Hau orohar ez da problema tribiala. Hainbat alterntatiba badaude ere, Markov Chain Monte Carlo (MCMC) metodoek prozedura orokor bat eskeintzen digute edozein banaketa lagindu ahal izateko. Horietatik, metodorik ezagunenetako bat **Metropolis-Hastings algoritmoa** da.

Teorian ikusi dugun moduan, Metropolis-Hastings algoritmoak Markov kate bat lagintzen du; are gehiago, kate honen banaketa egonkorra guk lagindu nahi dugun banaketa izango da. Hau hala izanik, i. pausuko katearen balioa zehazteko[1] θ_i, katearen aurreko balioa erabiliko dugu soilik, θ_{i-1}. Algoritmoak honako pausuak jarraitzen ditu:

- **1. pausoa**: Aukeratu hasierako balio bat ausaz, θ_0.
- **2. pausoa**: Errepikatu hurrengo pausoak behin eta berriz ($i \geq 1$):
  - **2.1. pausoa**: Proposatu θ^*, katearen hurrengo behaketa, $g(\theta^*|\theta_{i-1})$ banaketa erabiliz.
  - **2.2. pausoa**: Kalkulatu honako ratioa:

$$\rho = \frac{g(\theta_{i-1}|\theta^*) \cdot f(\theta^*)}{g(\theta^*|\theta_{i-1}) \cdot f(\theta_{i-1})}$$

  - **2.3. pausoa**: θ^* balioa onartu, hau da, $\theta_i = \theta^*$, $p = min\{\rho, 1\}$ probabilitatearekin.
  - **2.4. pausoa**: Balioa ez bada onartua izan, $\theta_i = \theta_{i-1}$.

Nahiz eta Markov katea ez dugun esplizituki definitu, prozesu hau errepikatuz, guk lagindu nahi dugun $f(\theta)$ banaketa egonkorra duen Markov Kate bat lagintzen ari gara. Beraz, laginketako lehen balioak kentzen baditugu, gure banaketaren lagin bat lortuko dugu.[2]

Emaitza analitikoak vs. MCMC

MCMC-ren ohiko erabilera bat, estatistika Bayesiarraren testuinguruan, a posteriori probabilitate-banaketak lagintzea da. Hau ikusteko, har dezagun a posteriori banaketa-mota ezagun bat duen kasu praktiko bat. Bizitza errealean ez genuke inoiz MCMC erabiliko horrelako kasu batean, zuzenean azter genezakeelako a posteriori banaketa. Baina, hain zuzren ere, horregatik erabiliko dugu: kalkulu analitikoak Metropolis-Hasting algoritmoarekin lortuko ditugun emaitzekin alderatzeko.

Emaitza analitikoak

Abiapuntu gisa, har dezagun Bernoulli banaketa bat jarraitzen duen zorizko aldagai bat (txanpon bat jaurtitzean lortutako emaitza adierazten duena, adibidez). Beraz, aldagiak `TRUE` edo `FALSE` balioak hartuko ditu, `TRUE` izateko probabilitatea θ izango delarik. Analisiari begira, Bernoulliren parametroa (θ) estimatzea izango da gure helburua.

[1] Gardenkietan kateko aldagaien izenak X_i ziren, baina tutorial honetan kate horren aldagaiek banaketa baten parametroari egiten diotenez erreferentzia, θ_i erabiliko dugu.

[2] Gogora ezazu banaketa egonkorrera konbergitu arte, lortutako balioak ezin direla hartu banaketa horren lagin bat bezala.

Hala ere, praktika honen helburu nagusia Metropolis-Hastings algoritmoaren portaera aztertzea denez, parametroaren benetako balioa (egoera erreal batean inoiz ezagutuko ez genukeena) guk finkatuko dugu. Zehazki, asumitu dezagun benetako probabilitatea $\theta = 0.7$ dela. Jarraian, finkatutako banaketa laginduko dugu $n = 50$ bider, lagin bat lortzeko:

```
probabilitate.erreala <- 0.7
bernouilli.laginketa.tamaina <- 50
bernouilli.lagina <- runif(bernouilli.laginketa.tamaina) < probabilitate.erreala
bernouilli.lagina
```

```
##  [1] FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## [13]  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [25] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [37]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## [49]  TRUE FALSE
```

Kontutan izan Bernoulliko esperimentu hori n aldiz errepikatzean lortutako `TRUE` kopurua $Binomial(n, \theta)$ banaketa bat jarraituko duela. Kalkula dezagun gure laginean konkretuki lortutako `TRUE` balioen kopurua, `FALSE` balioen kopuruarekin batera:

```
binomial.true  <- sum(bernouilli.lagina)
binomial.false <- sum(!bernouilli.lagina)
binomial.true
```

```
## [1] 33
```

```
binomial.false
```

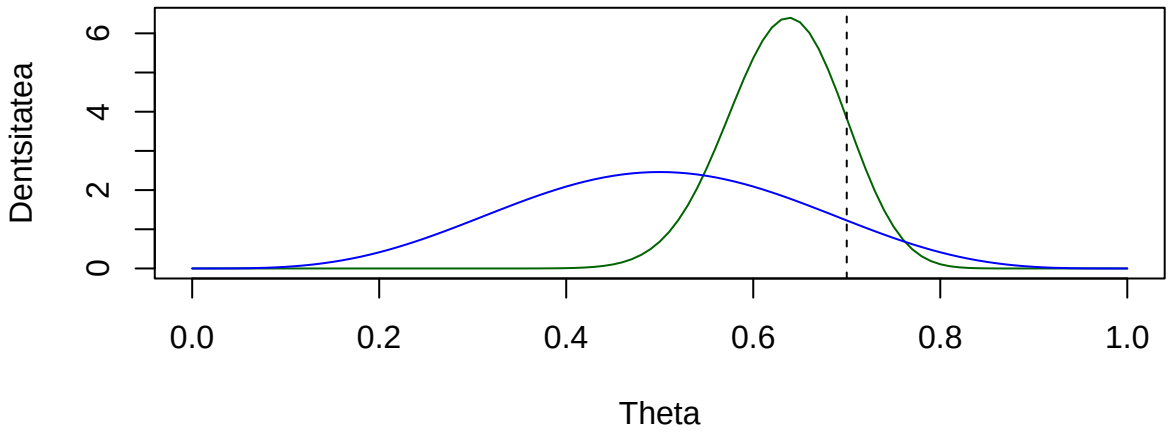
```
## [1] 17
```

Egoera erreal batean, lortutako lagina hartuko genuke (`bernouilli.lagina` aldagaian gordeta duguna), eta gure eginkizuna θ parametroa aztertzea izango litzateke (gure adibide honetan ezaguna dena). Estimazio frekuentista bat egingo bagenu, `TRUE` balio kopurua lagin tamainarekin zatituko genuke, baina praktika honetan analisi Bayesiar bat egingo dugu.

Horrelako kasuetan, arrunta da θ parametroaren *a priori* banaketa Beta banaketa bat izatea, α eta β *hiperparametroduna*: $f_0(\theta) \sim Beta(\alpha, \beta)$. Teorian ikusi dugun bezala, horrela egiten badugu, a posterioria ere Beta distribuzio bat izango da, $P(\theta|n_t, n_f) \sim Beta(\alpha + n_t, \beta + n_f)$, non n_t `binomial.true` izango den eta n_f `binomial.false`.

Gure kasuan, $(\alpha = 5, \beta = 5)$ hiperparametroak aukeratuko ditugu a priorirako eta, ezer baino lehen, a prioria eta a posterioria bistaratuko ditugu. Parametroaren balio erreala ere $(\theta = 0.7)$ irudikatuko dugu.

```
alfa.apriori <- 5
beta.apriori <- 5

alfa.aposteriori <- alfa.apriori + binomial.true
beta.aposteriori <- beta.apriori + binomial.false

x <- seq(0,1,0.01)
dentsitatea.apriori     <- dbeta(x, alfa.apriori,     beta.apriori)
dentsitatea.aposteriori <- dbeta(x, alfa.aposteriori, beta.aposteriori)

plot(x, dentsitatea.aposteriori, type="l", col="darkgreen", xlab="Theta", ylab="Dentsitatea")
lines(x, dentsitatea.apriori, col="blue")
abline(v=probabilitate.erreala, lty=2)
```



Ariketa: Probatu aldatzen esperimentuaren parametro guztiak: probabilitate erreala, Bernoulli banaketaren laginketa, lagin tamaina, a prioria, etab.

MCMC hurbilketa

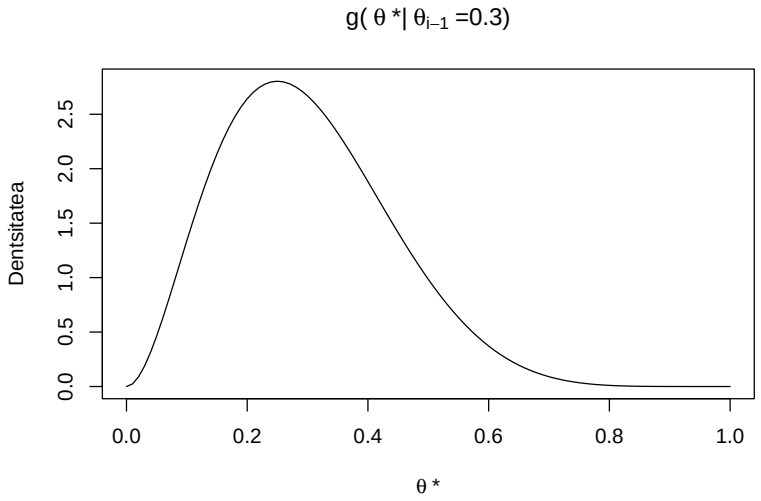
MCMC metodoekin a posteriori banaketa lagintzeko bi elementu behar ditugu:

- Hurrengo balioa proposatzeko banaketa bat $g(\theta^*|\theta_{i-1})$.
- Lagindu nahi dugun dentsitate funtzioa (edo horrekiko proportzionala den beste funtzio bat).

Balio berrien proposamenen banaketarako, $g(\theta^*|\theta_{i-1})$, edozein banaketa aukeratu dezakegu. Hala ere, gomendagarria da aurreko baliotik (θ_{i-1}tik) hurbil dauden balio berriak proposatzen dituen banaketa bat izatea. Gure kasuan, gainera, $[0, 1]$ tartean dauden balioak proposatzen dituen banaketa bat behar dugu, aztertzen ari garen θ parametroa probabilitate bat baita.

Bi alderdi horiek betetzeko, θ_{i-1} balioan zentratuta dagoen Beta banaketa bat erabil dezakegu. Zehazki, $Beta\big(10 \cdot \theta_{i-1},\ 10 \cdot (1 - \theta_{i-1})\big)$ banaketarekin lor dezakegu hori. Horrela, demagun $\theta_{i-1} = 0.3$ daukagula. Hurrengo baliorako (θ_i) proposamen bat lortzeko hurrengo kodea erabiliko genuke.

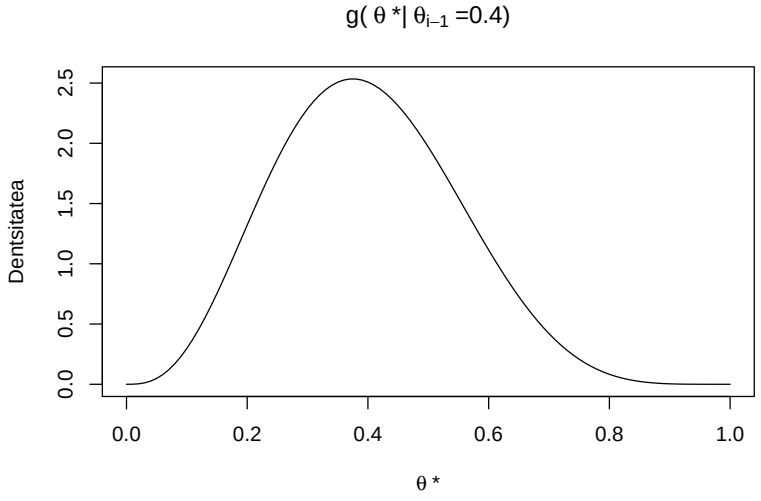
```
theta <- 0.3
x <- seq(0,1, by=0.01)
plot(x,dbeta(x, 10*theta, 10*(1-theta)), type="l",
     main=expression("g("~theta~"*|"~theta[i-1]~"=0.3)"),
     xlab=expression(theta~"*"), ylab="Dentsitatea")
```



Aurreko irudian ikusi dezakegun moduan, pausu honetako proposamen-banaketa $\theta_{i-1} = 0.3$ balioan zentratuta egongo litzateke, eta, hortaz, balio horren inguruan proposatuko dizkigu balioak. Hala ere, banaketa ez da simetrikoa, beraz: $g(\theta_i|\theta_{i-1}) \neq g(\theta_{i-1}|\theta_i)$. Ondorioz, ρ ratioaren kalkuluan espresio osoa hartu beharko dugu, ezin baitugu sinplifikatu (ikusi klaseko gardenkiak).

Ariketa: Demagun $\theta_{i-1} = 0.4$ dela. Kalkulatu proposamen banaketaren arabera ea zeintzuk izango diren hurrengo balioak proposatzeko probabilitate-dentsitatea: $\theta^* = 0.2 = (\theta_{i-1} - 0.2)$ eta $\theta^* = 0.6 = (\theta_{i-1} + 0.2)$

```
theta <- 0.4
x <- seq(0,1, by=0.01)
plot(x,dbeta(x, 10*theta, 10*(1-theta)), type="l",
     main=expression("g("~theta~"*|"~theta[i-1]~"=0.4)"),
     xlab=expression(theta~"*"), ylab="Dentsitatea")
```



Metropolis-Hastings algoritmoa aplikatu ahal izateko behar dugun beste elementua dentsitate-funtzio bat da, $f(\theta)$, gutxienez lagindu nahi dugun banaketarekiko proportzionala dena (edo berau). Gure kasuan a posteriori banaketa bat lagindu nahi dugunez ($f_1(\theta|D)$ notazioarekin adieraziko duguna), aski dugu Bayesen teorema kontuan

hartzearekin:

$$f_1(\theta|D) \propto P(D|\theta) f_0(\theta)$$

Hau da, parametroaren a posteriori banaketa proportzionala da $f_0(\theta)$ a prioriaren eta $P(D|\theta)$ egiantz-funtzioaren biderkadurarekiko (hau da, laginaren ***eta*** parametroaren baterako probabilitatea). Izan ere, falta den beste elementua, laginaren probabilitate marjinala ($P(D)$), ez dago θren menpe eta, beraz, konstante bat da.

Gure kasuan, zuzenean horrela adierazi dezakegu aurreko espresioa:

$$f_1(\theta|n_t, n_f) \propto P(n_t, n_f|\theta) f_0(\theta)$$

Izan ere, Metropolis-Hastings-en algoritmoa probabilitate-ratioetan oinarritzen denez, termino konstanteak baliogabetu egiten dira, eta horrek asko errazten digu lana, ez baitugu datuen probabilitate marjinal hori kalkulatzeko beharrik.

Behar ditugun bi elementuak argi ditugunean, algoritmoa oso erraz aplika dezakegu. Lehenik eta behin, hasierako balio bat finkatu behar dugu (θ_0). Lehendabizikoa denez, ezin dugu proposamen-banaketa erabili, baina a priori banaketa erabil dezakegu. Lehen balioa daukagunean, algoritmoaren iterazio bakoitzak balio berriaren (θ^*) proposamen bat lortu beharko dugu, aurreko balioa kontuan hartuta (θ_{i-1}) eta, bi balioak emanda, hurrengo ratioa kalkulatu beharko dugu:

$$\rho = \frac{Binomial\big(n_t;\ n, \theta^*\big) \cdot Beta\big(\theta^*;\ \alpha, \beta\big) \cdot Beta\big(\theta_{i-1};\ 10\theta^*, 10(1-\theta^*)\big)}{Binomial\big(n_t;\ n, \theta_{i-1}\big) \cdot Beta\big(\theta_{i-1};\ \alpha, \beta\big) \cdot Beta\big(\theta^*;\ 10\theta_{i-1}, 10(1-\theta_{i-1})\big)}$$

Ariketa: Identifika ezazu goiko ekuazioan dauden banaketen papera. Zein da proposamenak egiteko banaketa? Eta lagindu nahi dugun funtzioa?

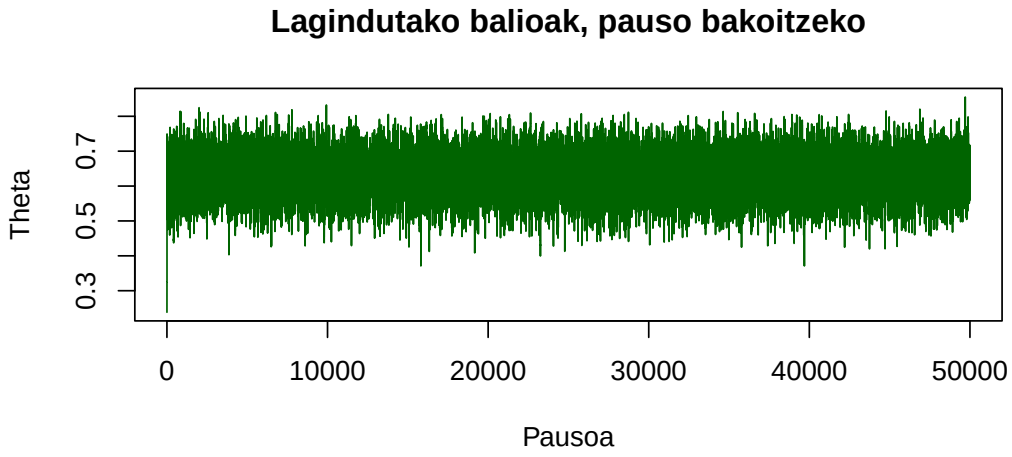
Ratio hori 1 edo handiagoa bada, $\theta_{i+1} = \theta^*$ izango da ziur. Txikiagoa bada, ρ probabilitatearekin $\theta_{i+1} = \theta^*$ izango da eta, bestela, $\theta_{i+1} = \theta_i$ izango da. Programatu dezagun beraz.

```
mcmc.lagin.kopurua <- 50000

#Lehenengo balioa a priori banaketatik ausaz
aposteriori.lagina <- rbeta(1, alfa.apriori, beta.apriori)

for (i in 1:mcmc.lagin.kopurua) {
  #theta^* balioa proposatu g banaketa erabiliz (ohartu aurreko balioaren baldintzapean dagoela!)
  balio.posible <- rbeta(1, 10*aposteriori.lagina[i], 10*(1-aposteriori.lagina[i]))

  #f(theta_{i-1})
  azkenaren.aposteriori <- (aposteriori.lagina[i]^binomial.true *
                            (1-aposteriori.lagina[i])^binomial.false)*
                           dbeta(aposteriori.lagina[i], alfa.apriori, beta.apriori)
  #f(theta^*)
  berriaren.aposteriori  <- (balio.posible^binomial.true *
                             (1-balio.posible)^binomial.false) *
                             dbeta(balio.posible, alfa.apriori, beta.apriori)
  #rho ratioa
  ratio <- (berriaren.aposteriori *
            dbeta(aposteriori.lagina[i], 10*balio.posible, 10*(1-balio.posible))
            )/(azkenaren.aposteriori *
```

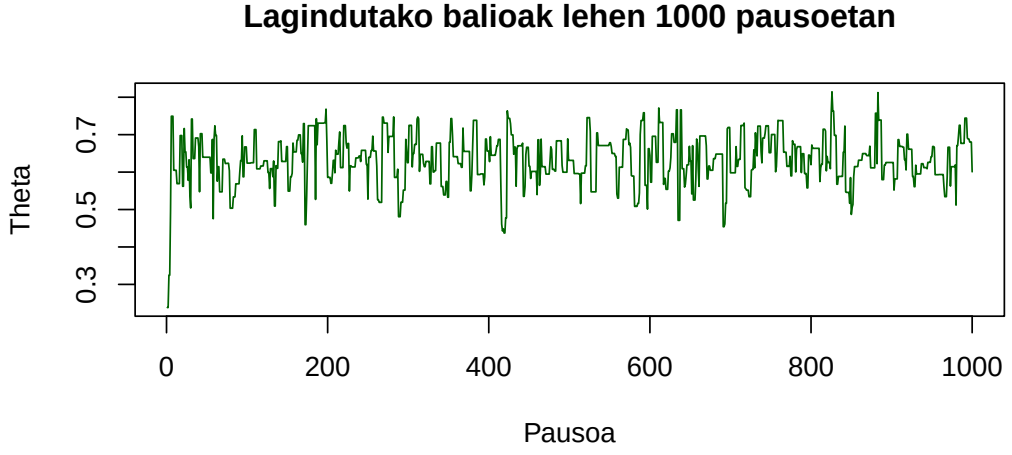
```
                dbeta(balio.posible, 10*aposteriori.lagina[i], 10*(1-aposteriori.lagina[i])))

  #Onartu theta^* edo ez probabilistikoki
  if (ratio > runif(1)) {
    balioa.gehitzeko <- balio.posible
  } else {
    balioa.gehitzeko <- aposteriori.lagina[i]
  }
  #Gehitu erabakitako balioa katera
  aposteriori.lagina <- c(aposteriori.lagina, balioa.gehitzeko)
}
```

Kateak jarraitu duen 'bidea' ikus dezakegu laginaren balioak irudikatuz. Asko direnez eta grafikoan zaila izango denez horiek era egokian bistaratzea, lehenengo eta azken 1000ak ere irudikatuko ditugu ondoren.

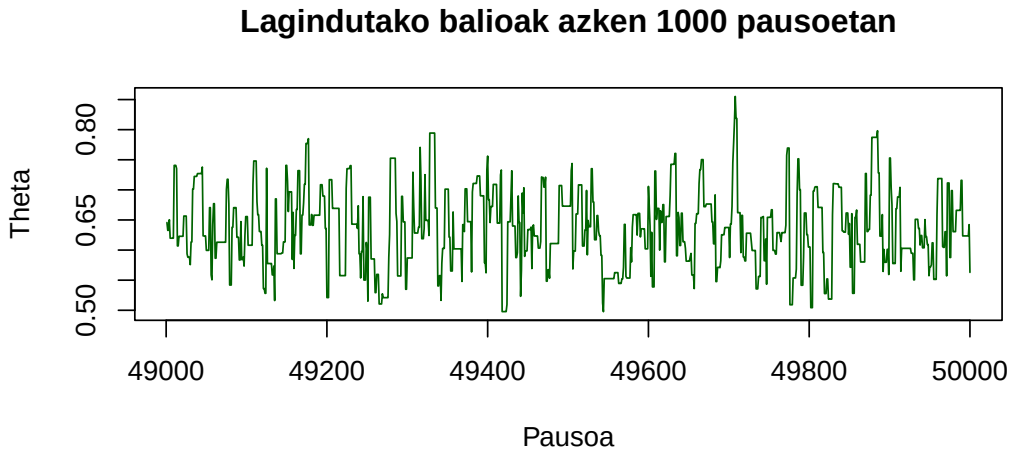
```
#Irudikatu kateak egindako ibilibidea
plot(aposteriori.lagina, type="l", col="darkgreen", xlab="Pausoa", ylab="Theta",
     main="Lagindutako balioak, pauso bakoitzeko")
```




```
plot(aposteriori.lagina[1:1000], type="l", col="darkgreen", xlab="Pausoa", ylab="Theta",
     main="Lagindutako balioak lehen 1000 pausoetan")
```

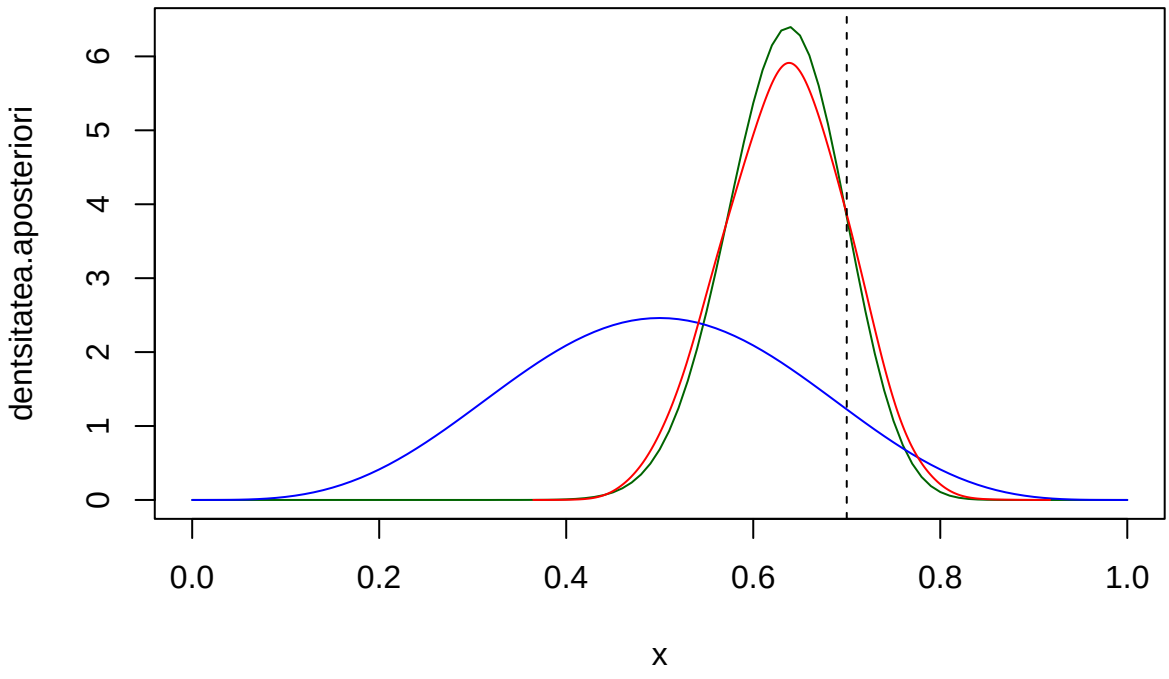


```r
plot(49001:50000, aposteriori.lagina[49001:50000], type="l", col="darkgreen",
     xlab="Pausoa", ylab="Theta", main="Lagindutako balioak azken 1000 pausoetan")
```



Teorian ikusi dugun moduan **burn-in** tartea kenduta (hau da, katearen lehen balioak kenduta) gainerako balioak banaketa egonkorraren laginak izango dira. Adibidez, lagindutako katearen azken 5000 balioak hartzen baditugu, lortutako lagina guri interesatzen zaigun banaketaren (a-posteriori banaketa kasu honetan) hurbilketa bat izango da:

```r
#Hartu lagindutako katetik azken 5000 balioak eta estimatu dentsitate funtzioa
dens <- density(tail(aposteriori.lagina, 5000), adjust=2)

#Irudikatu banaketa egonkorra (lagindu nahi dugun banaketa)
plot(x, dentsitatea.aposteriori, type="l", col="darkgreen")
lines(x, dentsitatea.apriori, col="blue")
lines(dens, col="red")
abline(v=probabilitate.erreala, lty=2)
```



Ikus dezakegunez, MCMC bidez lortutako hurbilketa (gorriz) soluzio analitikoaren oso antzekoa da (berdez).

Ariketa: Hasieran esan dugu gure helburua estimazio bat egitea zela. Kalkulatu bai kasu analitikorako eta bai MCMC bidez egindako laginketa erabiliz a posteriori itxaropena.

Ariketa: Kalkulatu, prozedura analitikoa eta MCMC erabiliz, 95%eko tarte Bayesiarra eta konparatu.

Ariketak

1. ariketa: Egin berriro aurreko adibidea, baina hurrengo balioaren proposamena egiteko honako banaketa uninforme hau erabiliz: $g(x^*|x_{i-1}) = Unif(x_{i-1} - w, x_{i-1} + w)$, non $w \in \mathbb{R}$. Probatu w parametroaren balio ezberdinetarako (adibidez $w \in \{0.0001, 0.1, 1, 100, 1000\}$) eta irudikatu kateak egiten duen ibilbidearen lehen 5000 balioak.

- Zer gertatzen da w-ren balio oso handietarako? Eta oso txikietarako? Zergatik lortzen dituzu emaitza horiek?
- Irudikatu baita ere, katearen azken 5000 balioen (banaketa egonkorra) histograma edota dentsitatea eta aztertu ea a posteriori banaketaren hurbilketa onak lortu dituzun kasu bakoitzean. Nolakoak dira hurbilketa hoiek w-ren balio desberdinetarako?

2. ariketa Izan bedi $(x_1, x_2, \ldots x_n)$ λ parametrodun Poisson banaketa batetik ateratako zorizko lagin bakuna:

```
lagina <- c(2, 3, 1, 3, 3,  3, 3, 4, 5, 7, 8, 9, 2, 1, 6, 4, 8, 6, 5, 5,
            8, 5, 6, 4, 10, 7, 3, 2, 7, 4, 3, 7, 8, 2, 9, 4, 5, 2, 4, 2,
            2, 7, 5, 5,  4, 2, 3, 5, 5, 6)
```

- Kalkulatu bai analitikoki (eskuz) eta bai MCMC erabiliz λ-ren a posteriori banaketa λ-rentzat a priori banaketa moduan $Gamma(\alpha, \beta)$ banaketa bat hartuz eta α eta β balio ezberdinekin probatuz. Gogoratu Rn α parametroa `shape` deitzen dela eta β `rate`. Horretarako, erabili honako g banaketa: $Unif(0, b)$, b balioaren aukera ezberdinak probatuz.
- Irudikatu batera a priori eta a lortutako bi aposteriori banaketak (teorikoa eta MCMC bidez lortutakoa).
- Kalkulatu bi kasuetan (teorikoki eta MCMC bidez) a posteriori itxaropena.