

MMC - Gibbs sampler & Hamiltonian Monte Carlo

Michaela Mašková

29. prosince 2020

Protokol se zabývá použitím Monte Carlo metod pro bayesovské odhadování parametrů.

Obsah

1	Markov Chain Monte Carlo	2
1.1	Metropolis Hastings a Gibbs sampler	2
1.2	Hamiltonian Monte Carlo	3
1.3	Porovnání metod	3
2	Definice problému a odvození algoritmu	4
2.1	Toy problem	4
2.2	Bayesovská lineární regrese	5
3	Julia language	6
3.1	Generátory náhodných čísel	6
3.2	Užitečné balíčky	7
4	Toy problem – implementace a řešení	7
4.1	Zadání	7
4.2	Gibbs sampler	7
4.3	Hamiltonian Monte Carlo	7
5	Lineární regrese – implementace a řešení	8
5.1	Zadání	8
5.2	Gibbs sampler	8
5.3	Hamiltonian Monte Carlo	8
6	Výsledky	9
6.1	Toy Problem	9
6.1.1	Gibbs sampler	9
6.1.2	Hamiltonian Monte Carlo	9
6.2	Lineární regrese	9
6.2.1	Gibbs sampler	9
6.2.2	Hamiltonian Monte Carlo	10
6.3	Tabulky	10
7	Závěr	11
8	Příloha - obrázky	12



1 Markov Chain Monte Carlo

Monte Carlo metody je možné použít na nejrozumnější typy úloh – výpočet integrálů, diferenciálních rovnic atd. V pravděpodobnostních úlohách často počítáme střední hodnoty, které jsou právě integrálem typu

$$\mathbb{E}_{x \sim U(0,1)}[f(x)] = \int_0^1 f(x)U(0,1)dx.$$



Střední hodnotu, **tedy daný integrál**, je pak možné odhadnout metodou Monte Carlo, nicméně musíme být schopni generovat náhodně z příslušné hustoty pravděpodobnosti. Pokud tedy máme například pouze naměřená data, která pochází z nějakého modelu, musíme nejdřív získat jejich distribuci a teprve poté z ní můžeme generovat náhodné vzorky, abychom spočítali příslušnou střední hodnotu. Problém nastává, pokud příslušná rozdělení není možné spočítat analyticky – což bude přesně případ našich modelových příkladů.

V tomto protokolu budeme používat dvě Monte Carlo metody

- Gibbs sampler
- Hamiltonian Monte Carlo.

Obě vychází z Markov Chain Monte Carlo.

1.1 Metropolis Hastings a Gibbs sampler

Metropolisův algoritmus funguje na následujícím principu: Místo konkrétní distribuce zvolíme Markovův řetězec, který bude konvergovat k hledané distribuci. Postup má pak následující kroky

1. Vybereme kernel $q(\theta|\theta^{(i)})$.
2. Vygenerujeme vzorek $\theta^* \sim q(\theta|\theta^{(i)})$.
3. S pravděpodobností

$$\min \left[1, \frac{p(\theta^*)q(\theta^{(i)}|\theta^*)}{p(\theta^{(i)})q(\theta^*|\theta^{(i)})} \right]$$

přijímáme $\theta^*: i \leftarrow i + 1, \theta^{(i)} \leftarrow \theta^*$, jinak zamítáme a vracíme se do bodu 2.

Zvoleným kernelem je nějaká náhodná procházka (Gaussův proces) s parametry ϕ .

Gibbs sampler je speciální vícedimenzionální podoba MCMC – Metropolis Hastings algoritmu. Pro distribuce tvaru

$$p(\theta_1, \theta_2, \dots, \theta_k)$$

jsou vzorky generovány postupně jako

1. $\theta_1^{(i+1)} \sim p(\theta_1|\theta_2^{(i)}, \dots, \theta_k^{(i)})$
2. $\theta_2^{(i+1)} \sim p(\theta_2|\theta_1^{(i+1)}, \dots, \theta_k^{(i)})$
- \vdots
- k. $\theta_k^{(i+1)} \sim p(\theta_k|\theta_1^{(i+1)}, \dots, \theta_{k-1}^{(i+1)})$,

kde pravděpodobnost přijetí je 1.

1.2 Hamiltonian Monte Carlo

Hamiltonovské Monte Carlo se jmenuje podle hamiltoniánu. Tato metoda totiž pracuje tak, že si vezme logaritmus hustoty pravděpodobnosti jako potenciální energii

$$U(\theta) = -\log p(\theta) = \frac{\theta^2}{2}.$$

Dále zavedeme kinetickou energii v proměnné p (hybnost)

$$K(p) = \frac{p^2}{2}.$$

Hamiltonovy rovnice říkají, že

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i} \quad \text{a} \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i},$$

kde x_i, p_i jsou i -té komponenty polohy a hybnosti. Jako polohu tedy bereme hledaný parametr θ . Samotný hamiltonián pak má tvar $H(\theta, p) = U(\theta) + \frac{1}{2}mp^2$ a dostáváme

$$\frac{d\theta}{dt} = p \quad \text{a} \quad \frac{dp}{dt} = -\theta.$$

Úloha se tak převede na simulaci diferenciálních rovnic pro vybrané časy $t = 0, \dots, t_N$. Vyžadujeme tak nějaké číslo L , což je počet skoků pro *leapfrog* algoritmus. Na začátku vygenerujeme náhodně hybnost $p^0 \sim \mathcal{N}(0, m)$ a řešíme hamiltonovské rovnice. Pro řešení těchto rovnic se používá numerická metoda *leapfrog integration*. Vždy po Δt časovém úseku jsou pak spočítány nové hodnoty θ a p .

HMC asymptoticky konverguje k pravé distribuci a navíc výsledné θ^* je nezávislé na p^* . Tuto metodu nebudeme implementovat přímo, ale využijeme balíček *Turing.jl* pro pravděpodobnostní programování, kde už je algoritmus naimplementován.

1.3 Porovnání metod

Metoda Gibbs sampleru je už ze zápisu jednodušší na implementaci a navíc vychází z klasické bayesovské inference, což nám umožňuje ji jednoduše využít na řadu různých problémů. Nicméně jednotlivé náhodně vybírané hodnoty jsou určitým způsobem korelované – závisí vždy na předchozím výsledku. HMC tuto nevýhodu eliminuje, čímž dokáže rychleji prozkoumat zkoumaný prostor proměnných. Nevýhodou HMC je však naopak výpočetní náročnost – použitím této metody v podstatě zdvojnásobujeme počet proměnných (přidáváme do modelu hybnost).

2 Definice problému a odvození algoritmu

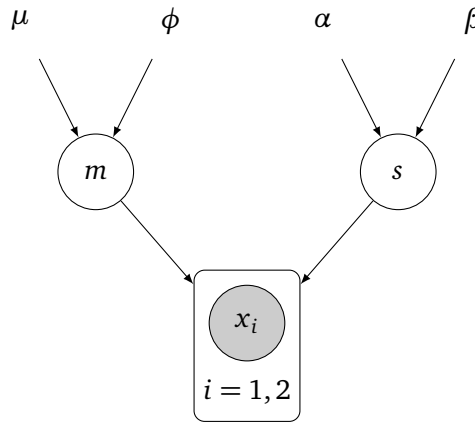
Ve statistice se často snažíme najít odhady parametrů. V bayesovském přístupu však hledaný parametr neodhadujeme bodově jako v klasické statistice. Nehledáme tedy například odhad dané veličiny jako průměr nebo jinou statistiku, ale snažíme se najít pravděpodobnostní rozdělení daných parametrů.


2.1 Toy problem

Představme si, že máme 2 hodnoty x_i , které jsou generovány z rozdělení $p(x|m, s) = \mathcal{N}(x|m, s)$, kde m a s jsou neznámé parametry. Bayesovský přístup spočívá v zavedení tzv. apriorního rozdělení přes tyto hledané parametry. Zde zvolíme jako apriorní rozdělení následující:

- $p(m) = \mathcal{N}(m|\mu, \phi)$
- $p(s) = \mathcal{IG}(s|\alpha, \beta)$.

Počítáme s tím, že α, β, ϕ a μ jsou hyperparametry – tedy jejich hodnoty fixujeme na předem zvolených číslech. Grafickou strukturu takového modelu si můžeme prohlédnout na Obr. 1.




Obrázek 1: Bayesovský model pro toy problem (* )

Problém však je v tom, že v takto definovaném modelu nedokážeme analyticky najít marginální rozdělení pro m a s , kde by jeden parametr nebyl závislý na druhém. Ani s pevnou volbou parametrů α, β a μ, ϕ tak nedokážeme analyticky nalézt hustoty rozdělení $p(m)$ a $p(s)$.

Proto použijeme metodu Monte Carlo.

Odvození pro Gibbs sampler

Pro odvození tvarů příslušných podmíněných marginálních rozdělení využijeme toho, že porovnáme jednotlivé tvary podle bayesovy věty ve tvaru

$$p(\theta|x, \alpha) \propto p(x|\theta)p(\theta|\alpha), \quad \text{$$

kde θ jsou hledané parametry a α představuje parametry apriorních rozdělení (hyperparametry).

Použitím pro tento konkrétní příklad dostáváme

$$p(m, s|x_1, x_2, \mu, \alpha, \beta, \Phi) \propto \prod_{i=1}^2 p(x_i, m, s) p(m|\mu, \Phi) p(s|\alpha, \beta) \quad (1)$$

a úpravou pak získáme

$$\begin{aligned}
 p(m, s | x_1, x_2, \mu, \alpha, \beta, \phi) &\propto \frac{1}{s} \frac{1}{s^{\alpha+1}} \exp \left[-\frac{1}{2} \frac{(m-x_1)^2}{s} - \frac{1}{2} \frac{(m-x_1)^2}{s} \frac{1}{2} \frac{(m-\mu)^2}{\phi} - \frac{\beta}{s} \right] \\
 p(m | s, x_1, x_2, \mu, \alpha, \beta, \phi) &\propto \frac{1}{s} \frac{1}{s^{\alpha+1}} \exp \left[-\frac{1}{2} \frac{(m-x_1)^2}{s} - \frac{1}{2} \frac{(m-x_1)^2}{s} \frac{1}{2} \frac{(m-\mu)^2}{\phi} \right] \\
 &\propto \exp \left[-\frac{1}{2} \left(m^2 \left(\frac{1}{\phi} + \frac{2}{s} \right) \right) - 2m \left(\frac{\mu}{\phi} + \frac{x_1+x_2}{s} \right) \right] \\
 &= \mathcal{N} \left(m \mid \left(\frac{1}{\phi} + \frac{2}{s} \right)^{-1} \left(\frac{\mu}{\phi} + \frac{x_1+x_2}{s} \right), \left(\frac{1}{\phi} + \frac{2}{s} \right)^{-1} \right) \\
 p(s | m, x_1, x_2, \mu, \alpha, \beta, \phi) &\propto \exp \left[-\frac{1}{2} \frac{(m-x_1)^2}{s} - \frac{1}{2} \frac{(m-x_1)^2}{s} - \frac{\beta}{s} \right] \\
 &= i\mathcal{G} \left(s \mid \alpha + 1, \frac{1}{2} (m-x_1)^2 + \frac{1}{2} (m-x_2)^2 + \beta \right)
 \end{aligned}$$

Algoritmus pak bude iterativně probíhat pro $i = 1 : K$, kde K je celkový počet opakování podle vzorců

$$m^{(i+1)} \sim \mathcal{N} \left(m \mid \left(\frac{1}{\phi} + \frac{2}{s^{(i)}} \right)^{-1} \left(\frac{\mu}{\phi} + \frac{x_1+x_2}{s^{(i)}} \right), \left(\frac{1}{\phi} + \frac{2}{s^{(i)}} \right)^{-1} \right) \quad (2)$$

$$s^{(i+1)} \sim i\mathcal{G} \left(s \mid \alpha + 1, \frac{1}{2} (m^{(i+1)} - x_1)^2 + \frac{1}{2} (m^{(i+1)} - x_2)^2 + \beta \right) \quad (3)$$

2.2 Bayesovská lineární regrese

I lineární regresi je možné řešit bayesovsky. V takovém případě předpokládáme, že existuje lineární závislost ve tvaru

$$\mathbf{y} = \boldsymbol{\theta} \cdot \mathbf{X} + \mathbf{e},$$

kde \mathbf{X} je matice sestavená z vektorů \mathbf{x} a $\boldsymbol{\theta}$ je vektor příslušných regresních parametrů. Vektor \mathbf{e} je pak gaussianový náhodný šum.

Pro lineární regresi sestavíme pravděpodobnostní model ve tvaru

$$p(\mathbf{y}, \boldsymbol{\theta} | \mathbf{X}, \alpha) = p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{X}, \omega) p(\boldsymbol{\theta} | \alpha) = \mathcal{N}(\mathbf{X}\boldsymbol{\theta}, \omega \mathbf{I}) \mathcal{N}(\mathbf{0}, \alpha^{-1} \mathbf{I}) \propto \exp \left[-\frac{1}{2} \omega \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 - \frac{1}{2} \alpha \|\boldsymbol{\theta}\|^2 \right].$$

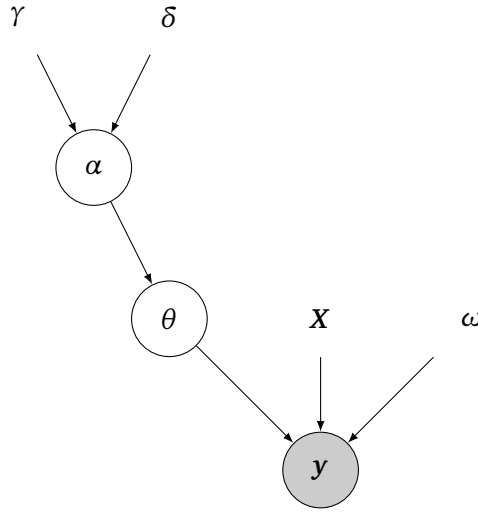
Této metodě se také říká Ridge Regression a řeší problém, kdy matice \mathbf{X} vede k nestabilitě numerické metody řešení.

Pokračujeme v bayesovském přístupu a zavedeme příslušné apriorní rozdělení pro $\alpha = G(\delta, \gamma)$. Parametr α zde zastává úlohu určitého regularizačního členu, který zabráňuje přefitování a vysokým hodnotám parametrů θ_i . Schéma modelu je pak vidět na Obr. 2.

Odvození pro Gibbs sampler

Stejně jako v toy problemu si spočítáme podmíněné distribuce pro $\boldsymbol{\theta}$ a α . Sdružená hustota pravděpodobnosti má tvar

$$\begin{aligned}
 p(\mathbf{y}, \boldsymbol{\theta}, \alpha | \mathbf{X}) &= p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{x}, \omega) p(\boldsymbol{\theta} | \alpha) p(\alpha) = \mathcal{N}(\mathbf{X}\boldsymbol{\theta}, \omega \mathbf{I}) \mathcal{N}(\mathbf{0}, \alpha^{-1} \mathbf{I}) G(\delta, \gamma) \\
 \implies p(\boldsymbol{\theta}, \alpha | \mathbf{y}, \mathbf{X}) &\propto \alpha^{\delta+d/2-1} \exp \left[-\frac{1}{2} \omega \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 - \frac{1}{2} \alpha \|\boldsymbol{\theta}\|^2 - \gamma \alpha \right]
 \end{aligned}$$



Obrázek 2: Bayesovský model pro lineární regresi (+).

a tedy jednotlivé marginální hustoty pravděpodobnosti závislé na příslušných parametrech budou

$$p(\theta|\alpha, \mathbf{y}, \mathbf{X}) = \mathcal{N}(\hat{\theta}, (\mathbf{X}^T \mathbf{X} + \alpha \mathbf{I})^{-1})$$

$$p(\alpha|\theta, \mathbf{y}, \mathbf{X}) = \mathcal{G}\left(\delta + \frac{d}{2}, \gamma + \|\theta\|^2\right).$$

Vzorce pro algoritmus Gibbs sampleru tak budou mít tvar

$$\theta^{(i+1)} \sim \mathcal{N}(\hat{\theta}, (\mathbf{X}^T \mathbf{X} + \alpha^{(i)} \mathbf{I})^{-1}) \quad (4)$$

$$\alpha^{(i+1)} \sim \mathcal{G}\left(\delta + \frac{d}{2}, \gamma + \|\theta^{(i+1)}\|^2\right) \quad (5)$$

3 Julia language

3.1 Generátory náhodných čísel

Julia language je programovací jazyk určený převážně pro vědecké výpočty se zaměřením hlavně na rychlost a jednoduchost.

Pro generování náhodných čísel z rozdělení $U(0, 1)$ je použit Mersenne Twister s použitím speciální knihovny Merseen Twister Library. V protokolu je nutné generovat náhodná čísla z několika různých rozdělení. Generování z $\mathcal{N}(0, 1)$ probíhá pomocí **Zigurratova** algoritmu. Pro získání rozdělení s jinou střední hodnotou nebo rozptylem jsou použity známé transformace rozdělení.

Julia má implementovány metody pro generování náhodných čísel z většiny běžně používaných rozdělení pro vlastní nastavení parametrů. Základní generování pak probíhá v jednoduchém příkazu

```

1 x = rand(distribution(parameters), size)
2
3 a = rand(Normal(3, 5), 10)
4 B = rand(Gamma(1, 2), 5, 5)

```

kde **a** je vektor o velikosti 10 vygenerovaný z rozdělení $\mathcal{N}(3, 5)$ a **B** matice rozměru 5×5 , jejíž prvky jsou náhodně vygenerovány z $G(1, 2)$.

3.2 Užitečné balíčky

Pro implementaci Hamiltonian Monte Carlo byl použit balíček Turing.jl, který je určený pro pravděpodobnostní programování. Skrývá v sobě implementované metody pro mnoho Monte Carlo metod včetně HMC, která bude použita v tomto projektu.

Zde se hodí podotknout, že pro Hamiltonian Monte Carlo může být náročné najít správné nastavení parametrů – hlavně délku kroku L , která je pro tuto metodu klíčová. Existuje rozšíření nazývané U-Turn Sampler, které dynamicky a automaticky najde správnou hodnotu L . V balíčku Turing.jl je No U-Turn implementován v metodě NUTS(). Pomocí NUTS metody tak byla zjištěna příslušná hodnota kroku L , která byla následně použita pro klasickou metodu HMC().

4 Toy problem – implementace a řešení

4.1 Zadání

Základní problém bude definován pro dva body, pro které platí

$$x_1 = 1, x_2 = 2.$$

Zvolíme hyperparametry modelu

$$\alpha = 2$$

$$\beta = 3$$

$$\Phi = 1$$

$$\mu = 5.$$



4.2 Gibbs sampler

Pro Gibbs sampler vytvoříme cyklus využívající odvozené vzorce (2) a (3), který pro zadané parametry a počet iterací vrátí odhadované hodnoty m a s jako vektor. Tyto hodnoty už jsou získané vzorky z hledaných rozdělení.

```

1 mi = zeros(iter+1)
2 si = zeros(iter+1)
3 mi[1] = m
4 si[1] = s
5 for i in 1:iter
6     E = (1/phi + 2/si[i])^(-1) * (mu/phi + (x1 + x2)/si[i])
7     var = (1/phi + 2/si[i])^(-1)
8     mi[i+1] = rand(Normal(E, var))
9     a = alpha + 1
10    b = 0.5*(mi[i+1] - x1)^2 + 0.5(mi[i+1] - x2)^2 + beta
11    si[i+1] = rand(InverseGamma(a, b))
12 end

```

4.3 Hamiltonian Monte Carlo

Pro implementaci HMC využijeme balíček Turing.jl v jazyce Julia. Pak stačí definovat model jako

```

1 @model toy(x, y, alpha, neta, phi, mu) = begin
2     s ~ InverseGamma(alpha, beta)
3     m ~ Normal(mu, sqrt(phi))
4     x ~ Normal(m, sqrt(s))

```

```

5  y ~ Normal(m, sqrt(s))
6  return s,m
7  end

```

Model je definovaný tak, že vybírá náhodně $s \sim G(\alpha, \beta)$, $m \sim \mathcal{N}(\mu, \Phi)$ a následně příslušné x, y z rozdělení závislého na získaných parametrech. Provedení samotné simulace je pak vyřešeno příkazem

```

1 chain = sample(toy(x1, x2), HMC(0.1, 5), iter)

```

kde do modelu vstupují proměnné x_1, x_2 , používáme HMC a počet iterací je `iter`. Z výsledného řetězce pak dostaneme znovu vektor získaných vzorků m a s , pro které můžeme odhadnout rozdělení a získat i rozdělení $p(m, s | x_1, x_2)$.

5 Lineární regrese – implementace a řešení

5.1 Zadání

Lineární regrese je poněkud zajímavějším příkladem použití metod na reálný problém. Definujeme problém dvourozměrné lineární regrese pro 100 hodnot $x_1 \in (-5, 15)$ a $x_2 \in (-7, 7)$. Závislost $y(x_1, x_2)$ bude daná předpisem

$$y = 4.1 - 5.8x_1 - 2.3x_2 + e,$$

kde $e \sim \mathcal{N}(0, 3)$ bude gaussovský šum. To znamená, že reálné hodnoty parametru $\theta = (4.1, -5.8, -2.3)$.

Volba hyperparametrů apriorního rozdělení α pak bude $\delta = 1.1$ a $\gamma = 0.5$.

5.2 Gibbs sampler

Pro implementaci Gibbsova sampleru použijeme stejný postup jako v předchozím případě, použijeme odvozené rovnice (4) a (5). Odhadujeme tedy parametry lineární regrese θ za pomoci apriorních rozdělení.

Iterační cyklus tak bude vypadat jako

```

1 for i in 1:iter
2   E = (X'*X)^(-1)*X'*y
3   D = Symmetric((X'*X + alpha[i]*I)^(-1))
4   theta[i+1,:] = rand(MvNormal(E,D))
5
6   a = delta + d/2
7   b = gamma + norm(theta[i+1,:])^2
8   alpha[i+1] = rand(Gamma(a,b))
9 end

```

5.3 Hamiltonian Monte Carlo

Pro HMC budeme postupovat podobně, nicméně zůstaneme u klasického hierarchického modelu, kde budeme mít $\alpha \sim G(\gamma, \delta)$ a $\theta \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}I)$. Výsledný model tak bude mít formu

```

1 @model linreg(X,y,gamma,delta,E) = begin
2   alpha ~ Gamma(gamma,delta)
3   D = Symmetric((X'*X + alpha*I)^(-1))
4   theta ~ MvNormal(E,D)
5   sigma ~ truncated(Normal(0, 100), 0, Inf)
6   y ~ MvNormal(X*theta,sqrt(sigma))
7   return theta
8 end

```

Využijeme toho, že o bodech y nemáme žádnou informaci a použijeme neinformativní formu rozptylu, která je reprezentována právě jako $\sigma \sim \text{truncated}(\text{Normal}(0, 100), 0, \text{Inf})$.

6 Výsledky

Simulace byla provedena pro různé počty náhodně vygenerovaných vzorků: 50, 500 a 5000. Vzhledem k tomu, že obě metody jsou založeny na náhodném generování čísel, byla simulace pro každou metodu a nastavení parametrů provedena 10×, aby bylo možné porovnávat **náhodnost** jednotlivých řešení daného problému.

Poznámka: Všechny grafy jsou uvedené na konci dokumentu, aby bylo zachováno jejich pořadí a bylo možné je lépe sledovat.

6.1 Toy Problem

V tomto případě hledáme rozdělení dvou parametrů, což nám umožňuje vykreslit si sdruženou hustotu pravděpodobnosti $p(m, s | x_1, x_2)$. Pro různé počty iterací (vzorků) **byly spočítány** konfidenční intervaly hodnot m a s , **kde můžeme navíc sledovat** střední hodnoty těchto parametrů pro různé realizace simulace.

6.1.1 Gibbs sampler

Nejdříve se podíváme na vliv počtu iterací pro počáteční hodnoty parametrů $m = 5, s = 2$. Na Obr. 3 si můžeme všimnout, že pro 50 iterací se výsledná rozdělení parametrů m a s významně liší. Pro 500 iterací již vidíme na Obr. 5, že se rozdělení více podobají navzájem, střední hodnoty jednotlivých parametrů se také přibližují, což můžeme vizuálně porovnat na Obr. 6. Nakonec můžeme na Obr. 7 vidět, že pro 5000 iterací už jsou rozdělení velmi podobná a vliv „náhody“ byl tak minimalizován.

6.1.2 Hamiltonian Monte Carlo

Stejně závěry jako výše budeme chtít udělat pro HMC a zjistit, jestli se Gibbs sampler a HMC liší v průzkumu distribučního prostoru. Na Obr. 9 jsou sdružená rozdělení velmi odlišná jedno od druhého. Situace se s rostoucím počtem iterací mění, což je vidět na obrázcích 11 a 13. Rozdělení pro 5000 iterací se pro různé simulace již výrazně neliší. Stejně závěry můžeme pozorovat na 95% konfidenčních intervalech pro parametry m, s na Obr. 10, 12 a 14, kde vidíme postupné sjednocování konfidenčních intervalů i středních hodnot parametrů.

6.2 Lineární regrese


V lineární regresi bylo cílem odhadnout správně koeficienty pro problém definovaný jako $y = 4.1 - 5.8x_1 - 2.3x_2 + e$, kde $e \sim \mathcal{N}(0, 3)$. Hledali jsme tedy odhady tří parametrů $\theta = (\theta_1, \theta_2, \theta_3)$. Vizualizace výsledků proto proběhla tak, že byla vykreslena rozdělení jednotlivých parametrů θ_i , a to jejich odhad pomocí *kernel density estimation* a jako MLE odhad parametrů normálního rozdělení. Pro parametry byla navíc do grafu vykreslena skutečná hodnota parametru.

Pro každou hodnotu počtu iterací (50, 500, 5000) budou vykresleny odhady rozdělení parametrů θ_1, θ_2 a θ_3 **2×**, aby bylo možné porovnat průběhy rozdělení pro jednotlivé průběhy simulací. Zároveň jsou vykresleny 95% konfidenční intervaly parametrů pro daný počet iterací a všech 10 simulací.

Na obrázcích si budeme moci všimnout, že oba samplery dávají velmi podobné výsledky, které se mírně liší od pravých hodnot parametrů, což je dáno hlavně rozptylem hodnot y .

6.2.1 Gibbs sampler

Byly zvoleny počáteční hodnoty parametrů $\alpha = 0.5$ a $\theta = (0, 0, 0)$, jelikož v lineární regresi obecně testujeme, jestli jsou výsledné parametry různé od nuly a nemáme navíc žádnou průvodní informaci o hodnotách parametrů.

	iterations	$\mathbb{E}[m]$	 m	$\mathbb{E}[s]$	$\sigma(s)$
Gibbs sampler	50	3.553	0.178	4.009	0.668
	500	3.541	0.050	4.067	0.220
	5000	3.536	0.018	4.060	0.101
HMC	50	3.165	0.432	3.560	1.361
	500	3.461	0.107	3.981	0.928
	5000	3.547	0.069	4.210	0.303

Tabulka 1: Tabulka charakteristik pro různé počty iterací - Toy problem.

Postupně na obrázcích 15, 17 a 19 můžeme vidět, jak se odhad hustoty pomocí *kernel density estimation* přibližuje k MLE odhadu hustoty pravděpodobnosti z dat. Více vzorků tak zajišťuje hladší průběh odhadované hustoty pravděpodobnosti. Na obrázcích 16, 18 a 20 vidíme, jak se měnily konfidenční intervaly pro jednotlivé simulace při různém potu iterací algoritmu. Opět si můžeme všimnout, že pro 5000 iterací jsou výsledky jednotlivých simulací téměř identické.

Průběh celé simulace ($10\times$ pro 50, 500 a 5000 vzorků) trval ≈ 20 vteřin.

6.2.2 Hamiltonian Monte Carlo

Výsledky pro HMC algoritmus a lineární regresi se od Gibbs sampleru liší tím, že častěji odhadují více-modální rozdělení, což můžeme vidět hned na příkladu na Obr. 21 pro odhad parametru θ_1 (interceptu). Odhad pomocí *kernel density estimation* se od MLE odhadu liší výrazněji než u Gibbs sampleru. Nicméně i tak je vidět na Obr. 23 a 25, že se oba odhady přibližují a s více vzorky dochází k hladšímu odhadu hustoty pravděpodobnosti.

Konfidenční intervaly na obrázcích 22, 24 a 26 ukazují, že hlavně odhad parametru θ_1 je nejméně přesný a nejvíce náchylný k rozdílům pro různé simulace. S rostoucím počtem iterací se však konfidenční intervaly liší stále méně.

Průběh celé simulace ($10\times$ pro 50, 500 a 5000 vzorků) trval ≈ 32 vteřin.

6.3 Tabulky

Pro numerické porovnání metod byly vytvořeny tabulky, které shrnují pro každý počet iterací (50, 500, 5000) střední hodnotu a standardní odchylku přes 10 simulací.

V tabulce 1 lze nahlédnout výsledky pro Toy problem. Pro obě metody vidíme, že s rostoucím počtem iterací (generovaných vzorků) se snižuje rozptyl středních hodnot odhadovaných parametrů. Zatímco střední hodnoty obou parametrů jsou pro Gibbs sampler víceméně stejné, u HMC se jejich hodnoty liší a zároveň zpřesňují. Nicméně pro Toy problem není cílem určit správnou hodnotu parametru, ale spíše ilustrovat vliv apriorní informace na výsledek.

Pro lineární regresi už můžeme porovnávat s reálnými hodnotami parametrů. V tabulce 2 si můžeme všimnout, že rozptyl (standardní odchylka) středních hodnot parametrů je velmi nízký, což indikuje, že rozdíl mezi jednotlivými simulacemi je minimální. Standardní odchylka se taktéž zmenšuje. Gibbs sampler dává stejné výsledky pro jednotlivé simulace už pro 50 vzorků, HMC rozptyl postupně klesá až téměř na 0. Výsledné hodnoty pro jednotlivé samplery se významně neliší.

	iterations	$\mathbb{E}[\theta_1]$	$\sigma(\theta_1)$	$\mathbb{E}[\theta_2]$	$\sigma(\theta_2)$	$\mathbb{E}[\theta_3]$	$\sigma(\theta_3)$
Gibbs sampler	50	3.7656	0.0001	-5.7840	0.0	-2.3426	0.0
	500	3.7651	0.0	-5.7842	0.0	-2.3402	0.0
	5000	3.7670	0.0	-5.7843	0.0	-2.3408	0.0
HMC	50	3.7105	0.0084	-5.7792	0.0001	-2.3417	0.0
	500	3.7810	0.0070	-5.7857	0.0001	-2.3414	0.0
	5000	3.7612	0.0003	-5.7838	0.0	-2.3405	0.0

Tabulka 2: Tabulka charakteristik pro různé počty iterací - lineární regrese.

7 Závěr

Cílem protokolu bylo vyzkoušet metodu Monte Carlo na bayesovské odhady parametrů. Byly demonstrovány dva přístupy: Gibbs sampler a Hamiltonian Monte Carlo. Oba přístupy je možné použít v odhadování pravděpodobnostních distribucí parametrů.

Ukázalo se, že Hamiltonian Monte Carlo je metodou, která má obecně vyšší rozptyl a rychleji tedy prozkoumává celý parametrický prostor. Pro menší počet iterací také častěji odhaduje více módů rozdělení než Gibbs sampler. Je nicméně výpočetně náročnější a pomalejší než Gibbs sampler – zatímco generování 10×5550 vzorků společně s ukládáním dat a vykreslením všech grafů trvalo pro Gibbs sampler ≈ 20 vteřin, pro HMC trvala stejná simulace ≈ 32 vteřin. Pro velké simulace, kde by bylo třeba generovat více vzorků by se tak více vyplatilo použít Gibbs sampler.

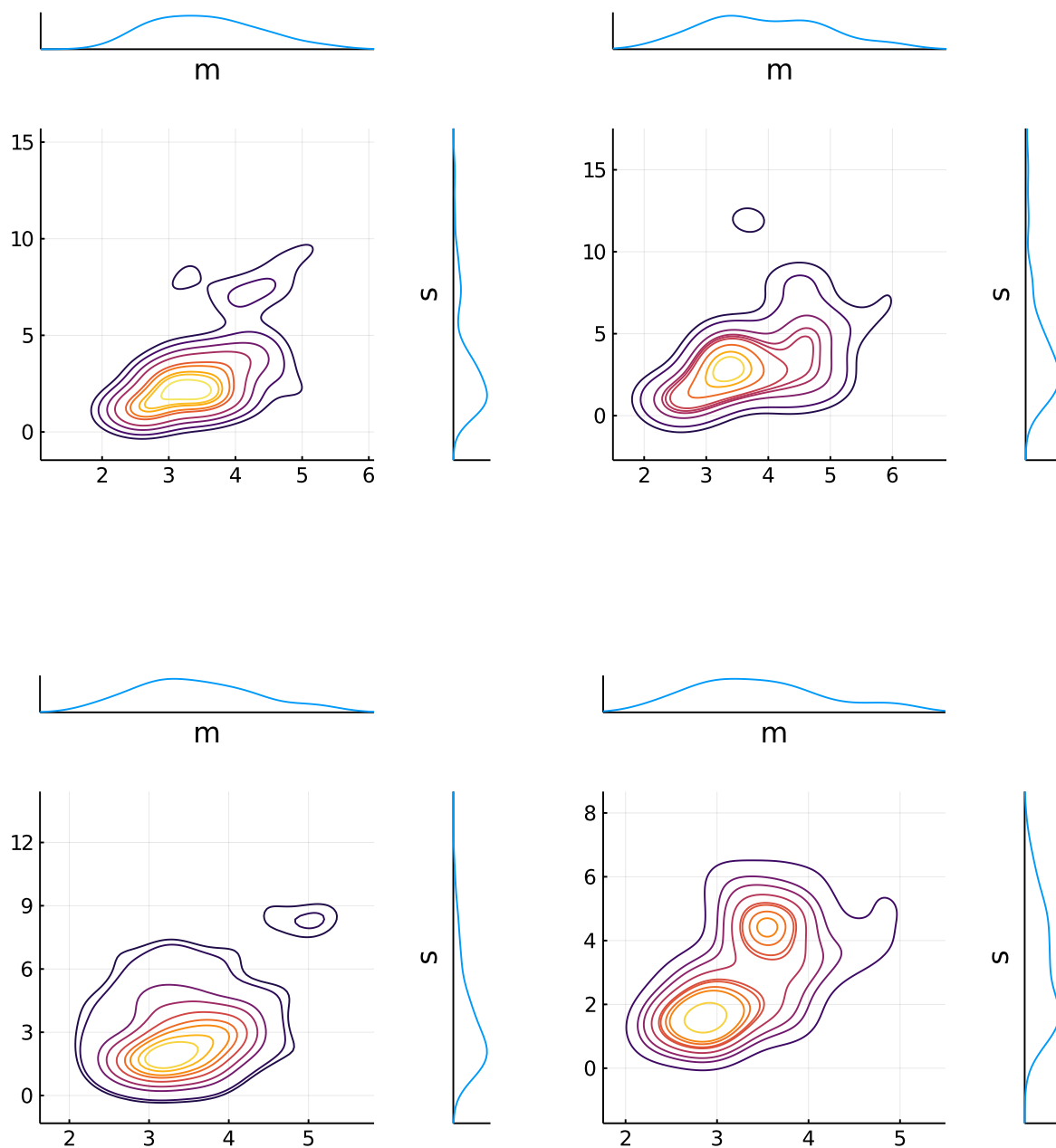
Pro lineární regresi byly odhadnuty koeficienty θ , které odpovídaly reálným hodnotám parametrů, ze kterých byla data generována. Ukázalo se, že jak Gibbs sampler tak HMC odhadují velmi podobné hodnoty parametrů i jejich rozdělení. Obě metody tedy lze použít pro odhad parametrů v problému lineární regrese.

Ukázalo se, že více iterací (vygenerovaných vzorků) v důsledku znamená, že metoda dává víceméně stejné výsledky pro větší počet simulací.

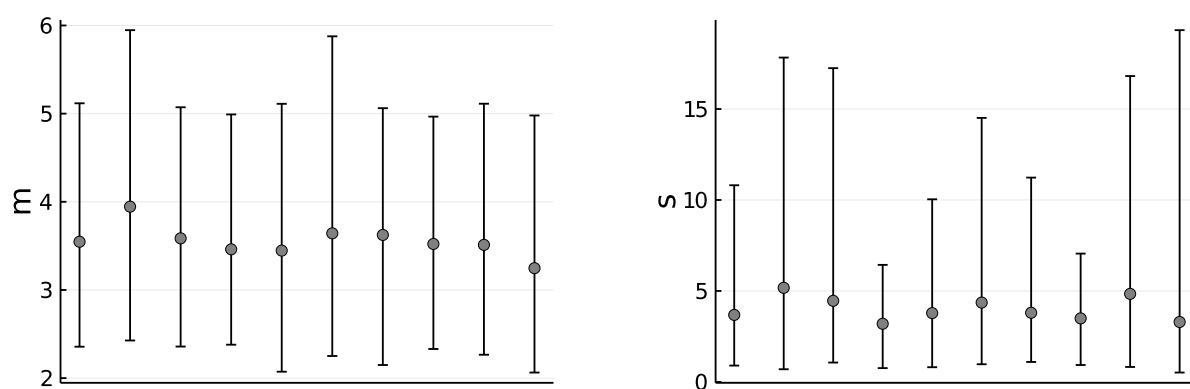
Pro další otestování metod by bylo možné je použít na reálná data, jelikož všechny simulace byly provedeny pouze pro uměle vygenerovaná data.

8 Příloha - obrázky

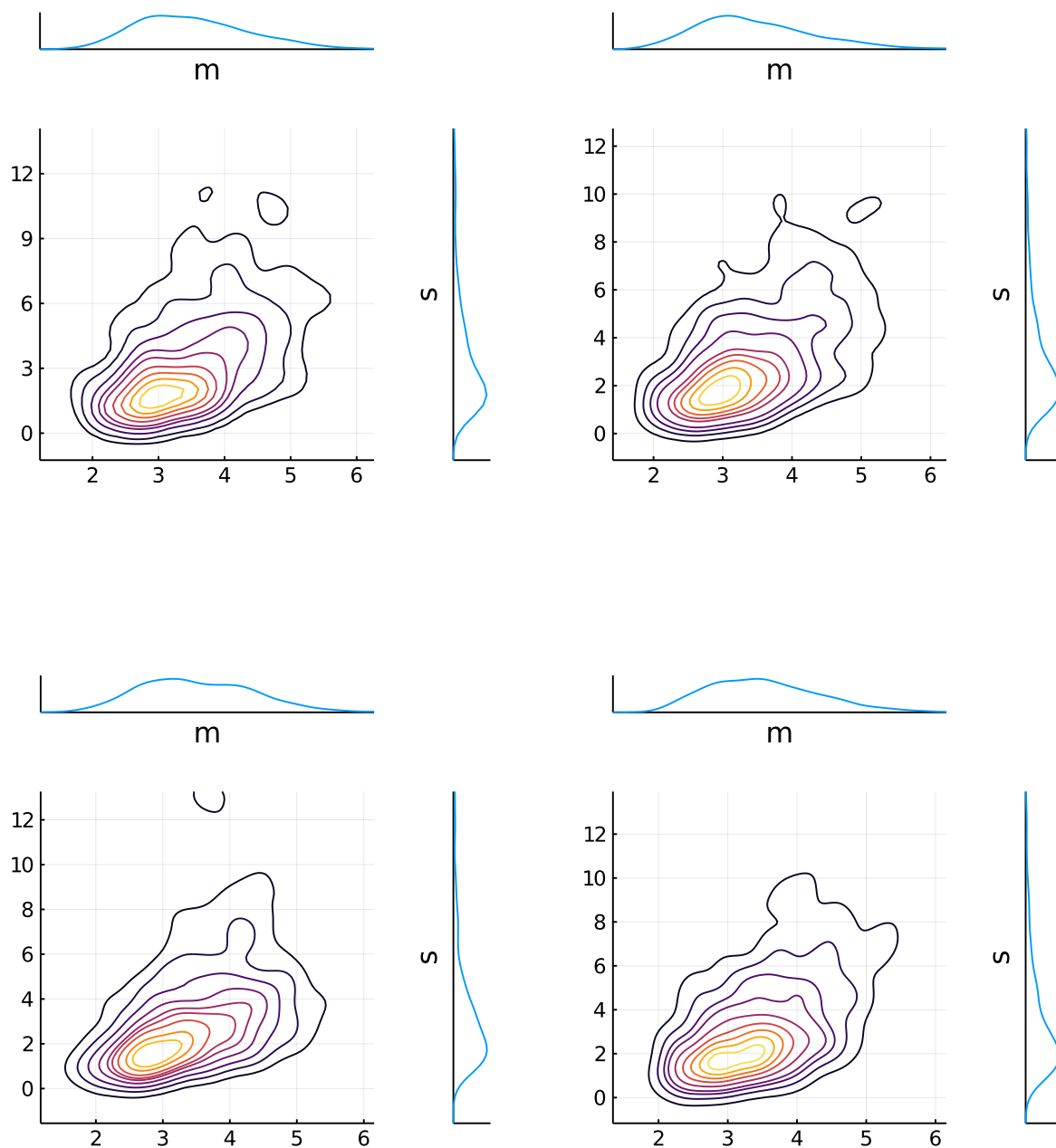
Příloha obsahuje všechny obrázky referencované v sekci Výsledky.



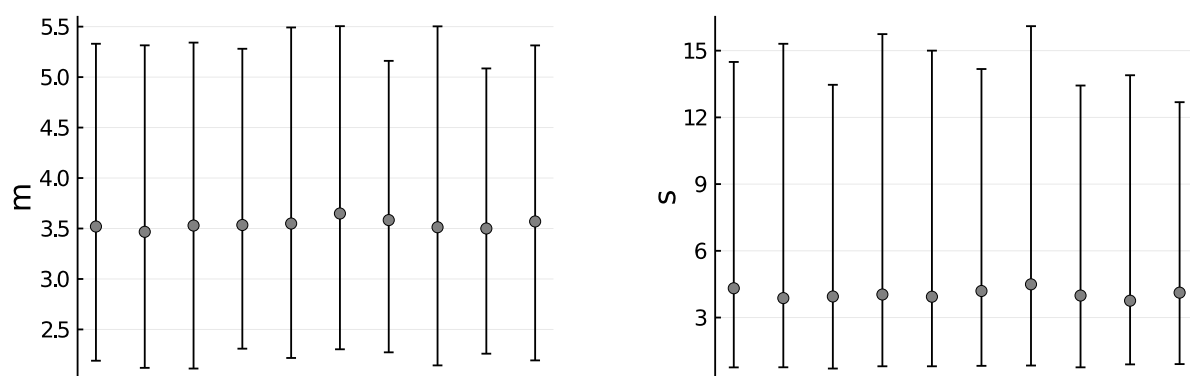
Obrázek 3: Ukázka 4 výsledných rozdělání parametrů m a s pro 50 iterací, Gibss sampler.



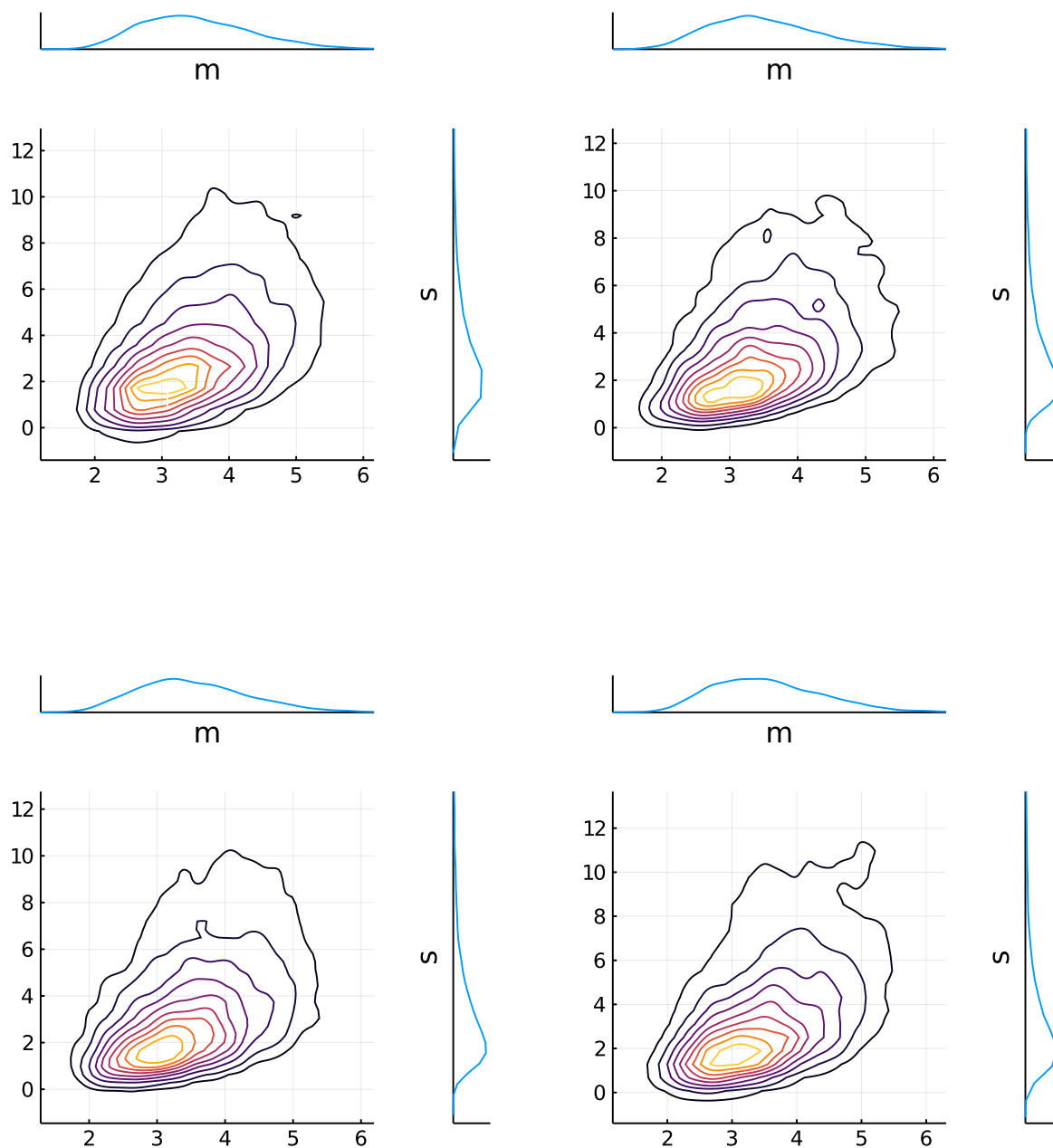
Obrázek 4: 95% konfidenční intervaly pro parametry m , s , 50 iterací, 10 realizací simulace, Gibss sampler.



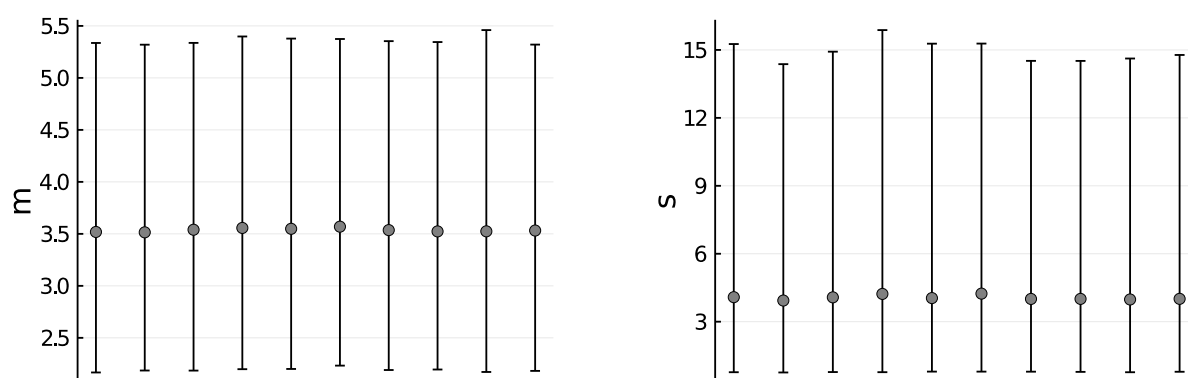
Obrázek 5: Ukázka 4 výsledných rozdělání parametrů m a s pro 500 iterací, Gibss sampler.



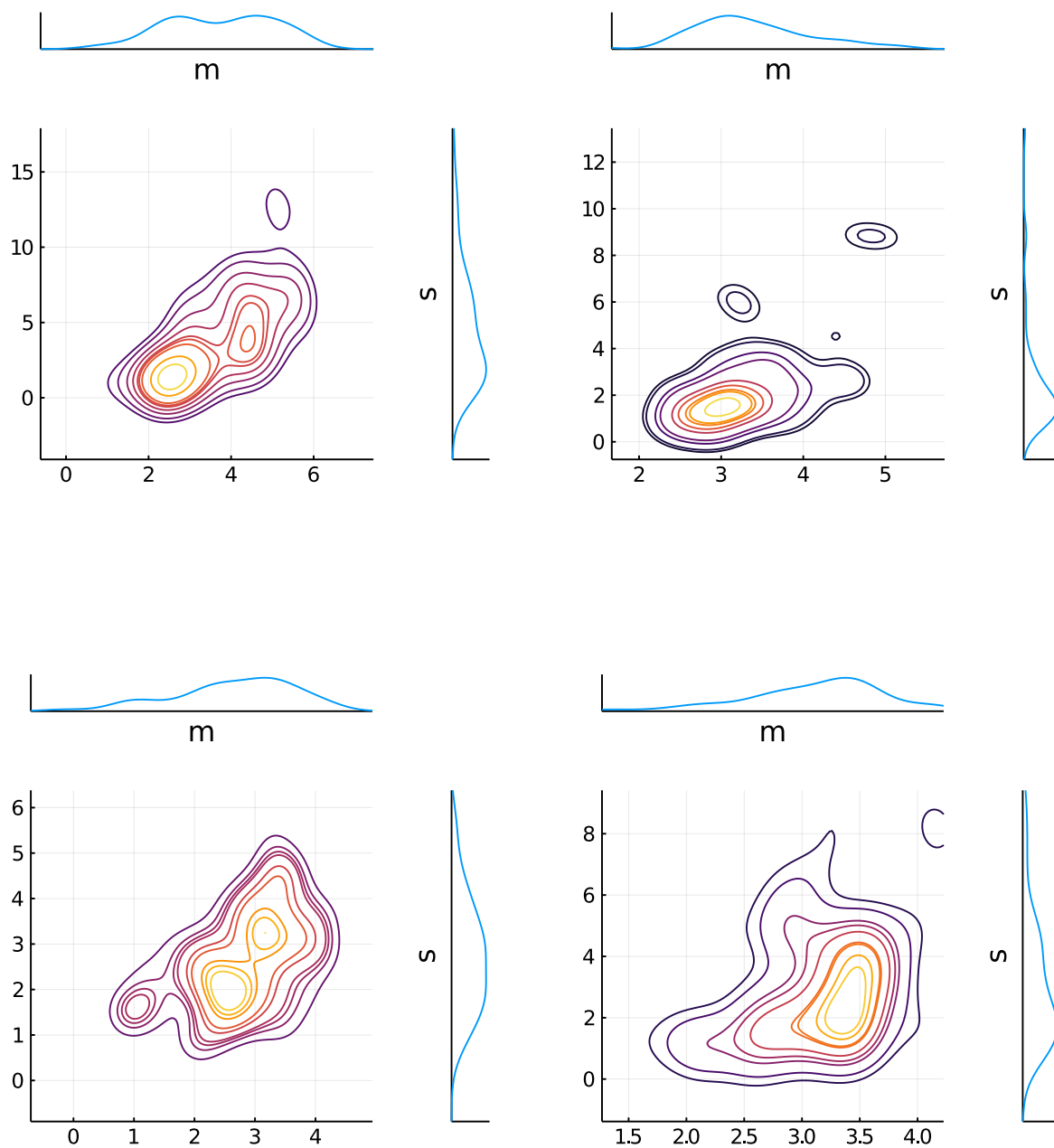
Obrázek 6: 95% konfidenční intervaly pro parametry m, s , 500 iterací, 10 realizací simulace, Gibss sampler.



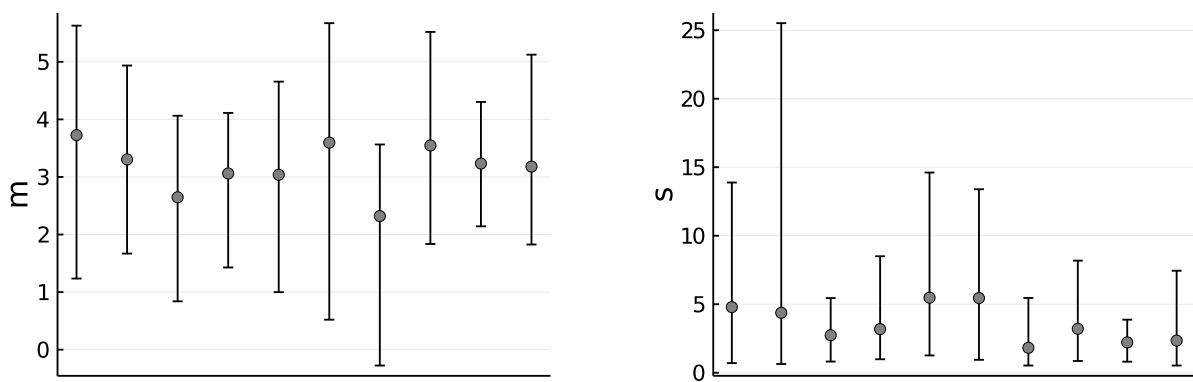
Obrázek 7: Ukázka 4 výsledných rozdělání parametrů m a s pro 5000 iterací, Gibss sampler.



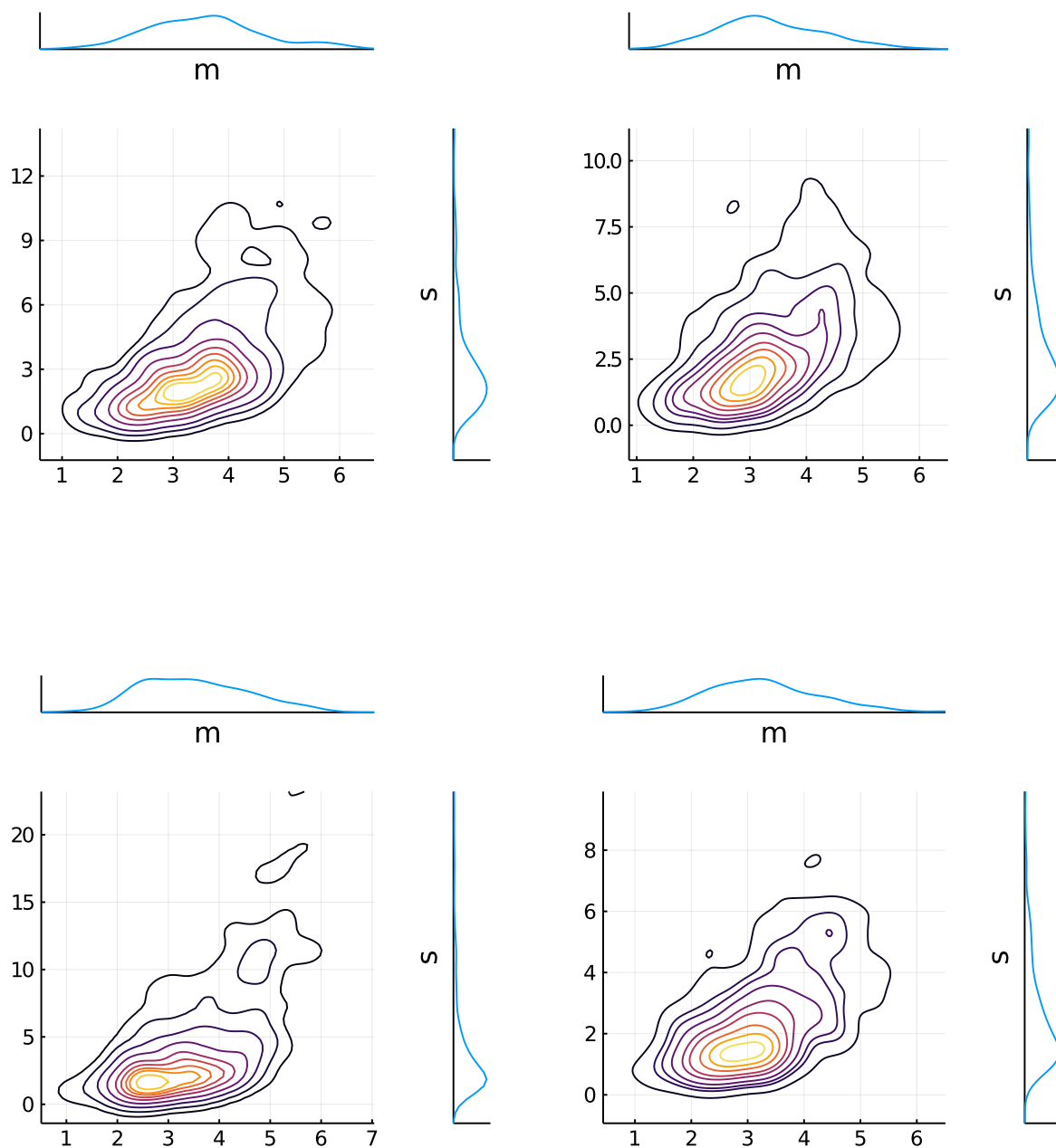
Obrázek 8: 95% konfidenční intervaly pro parametry m , s , 5000 iterací, 10 realizací simulace, Gibss sampler.



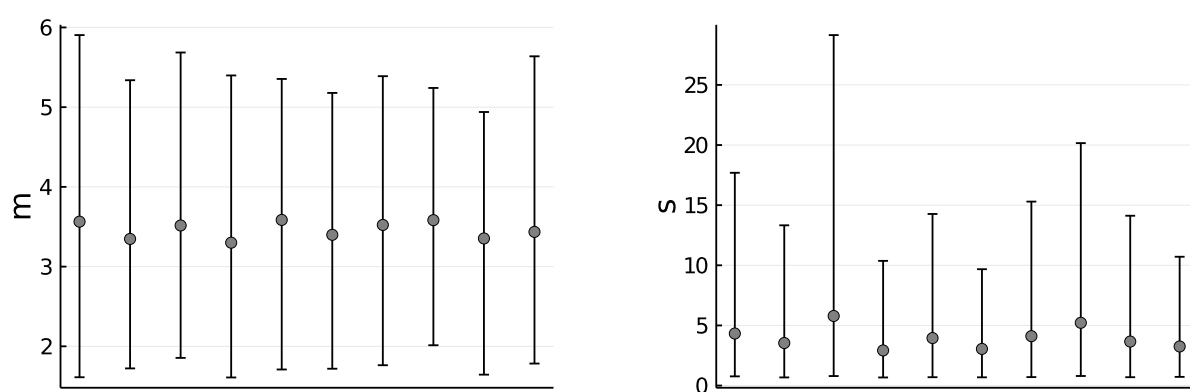
Obrázek 9: Ukázka 4 výsledných rozdělání parametrů m a s pro 50 iterací, HMC sampler.



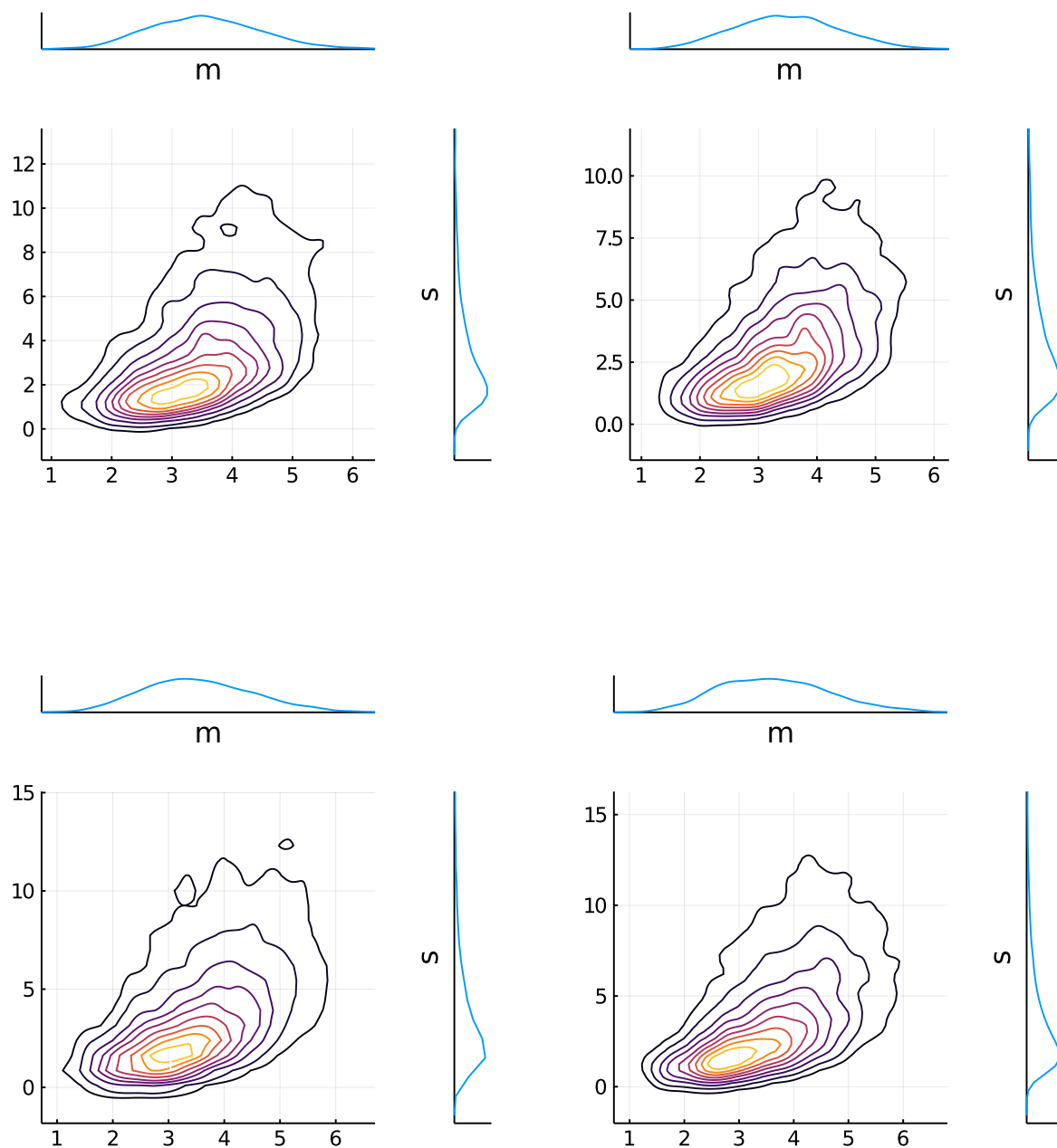
Obrázek 10: 95% konfidenční intervaly pro parametry m, s , 50 iterací, 10 realizací simulace, HMC sampler.



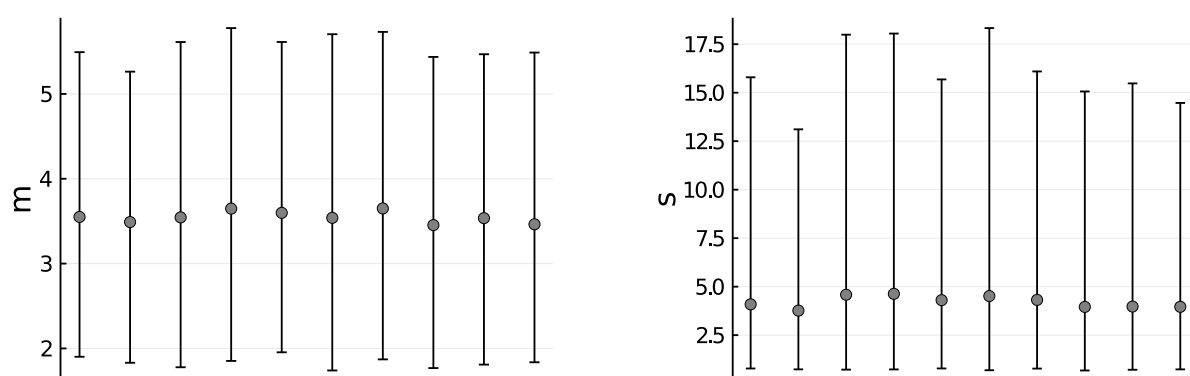
Obrázek 11: Ukázka 4 výsledných rozdělání parametrů m a s pro 500 iterací, HMC sampler.



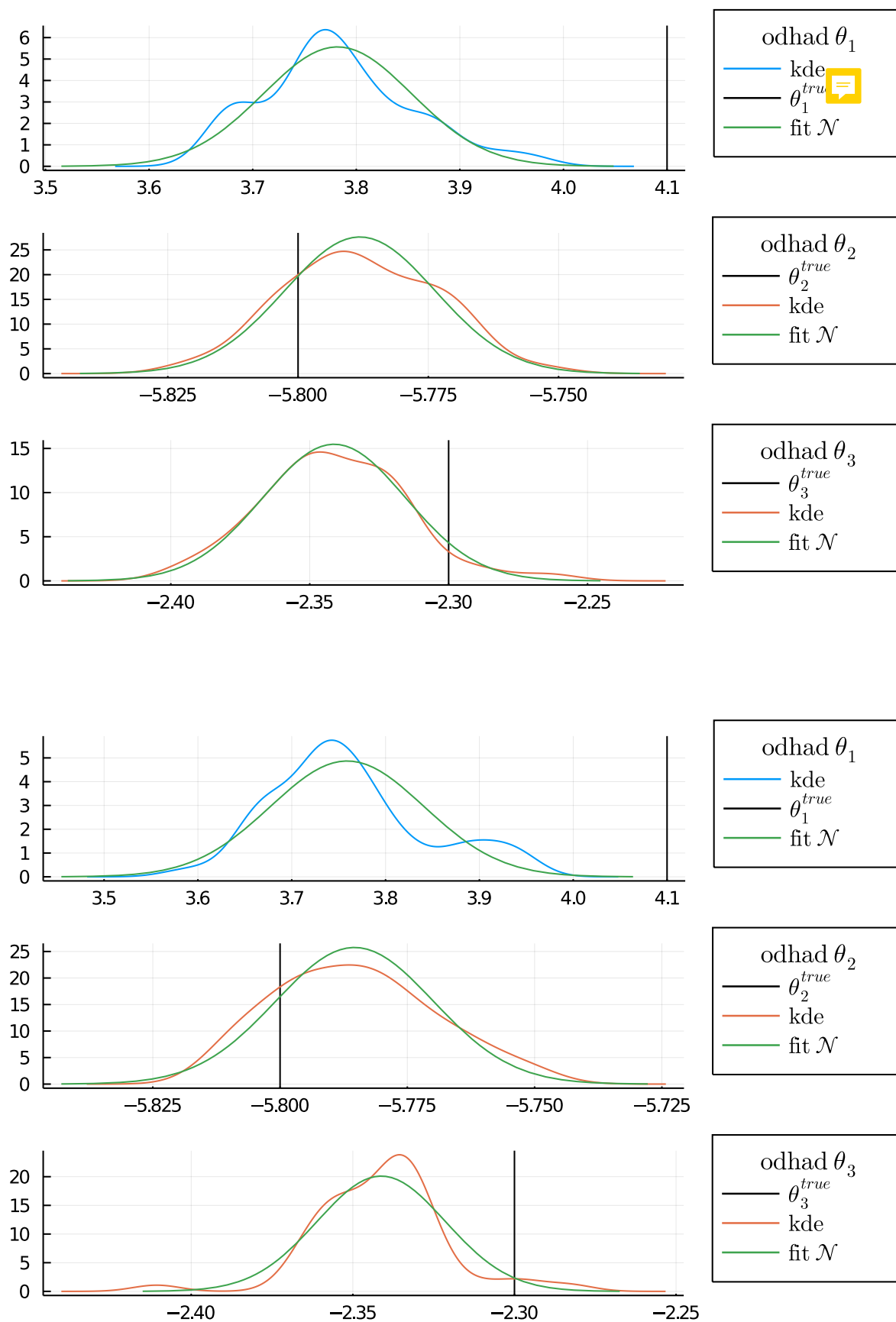
Obrázek 12: 95% konfidenční intervaly pro parametry m , s , 500 iterací, 10 realizací simulace, HMC sampler.

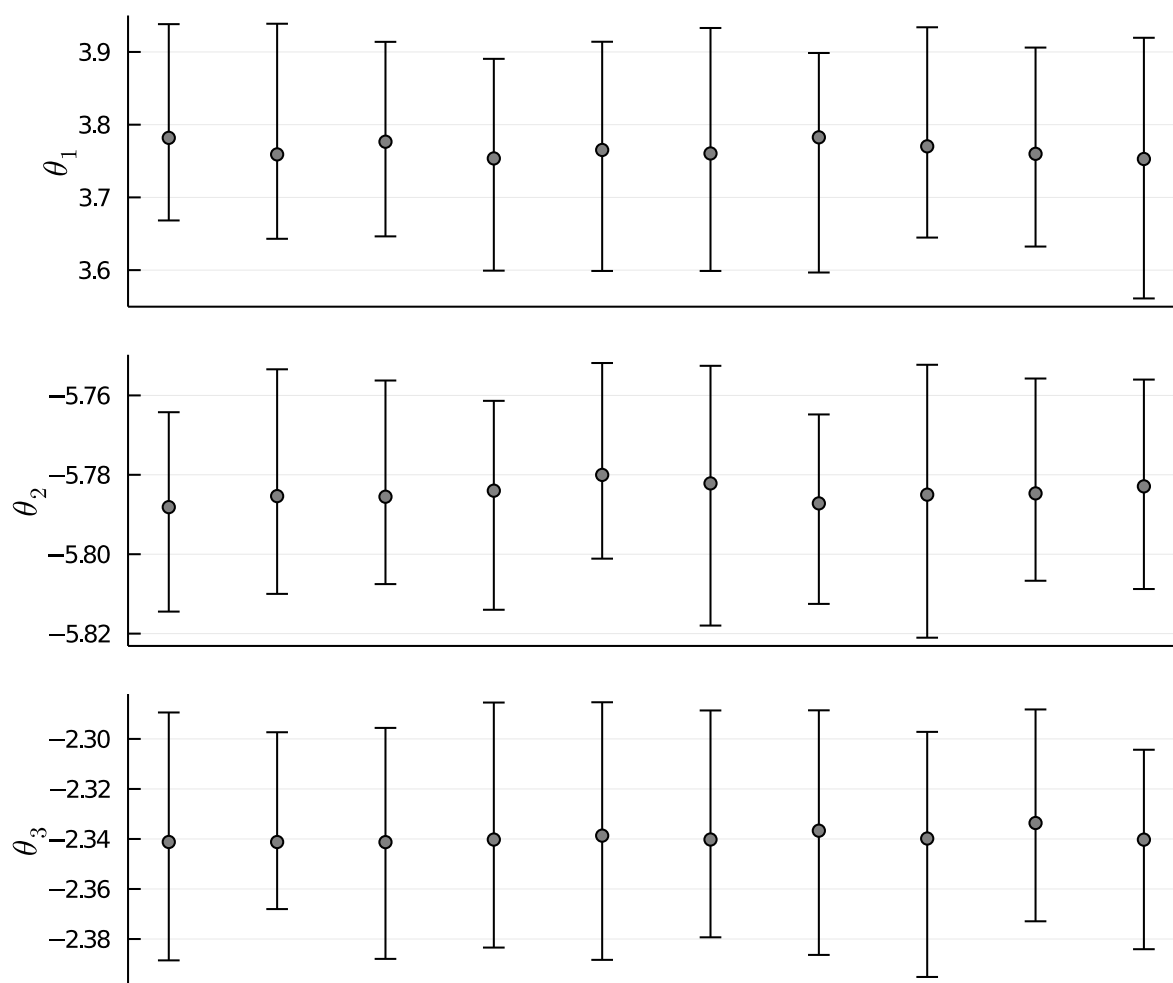


Obrázek 13: Ukázka 4 výsledných rozdělání parametrů m a s pro 5000 iterací, HMC sampler.

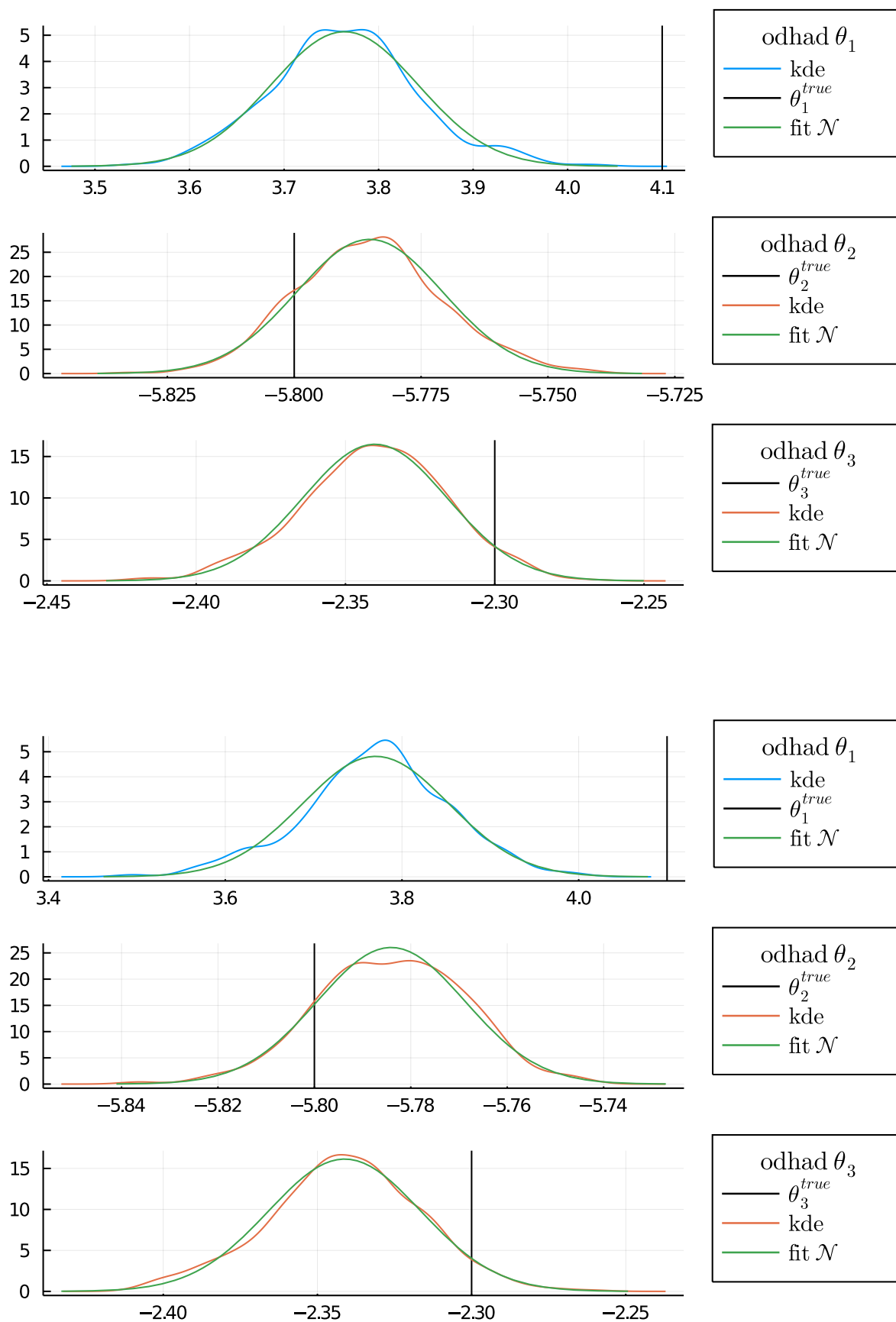


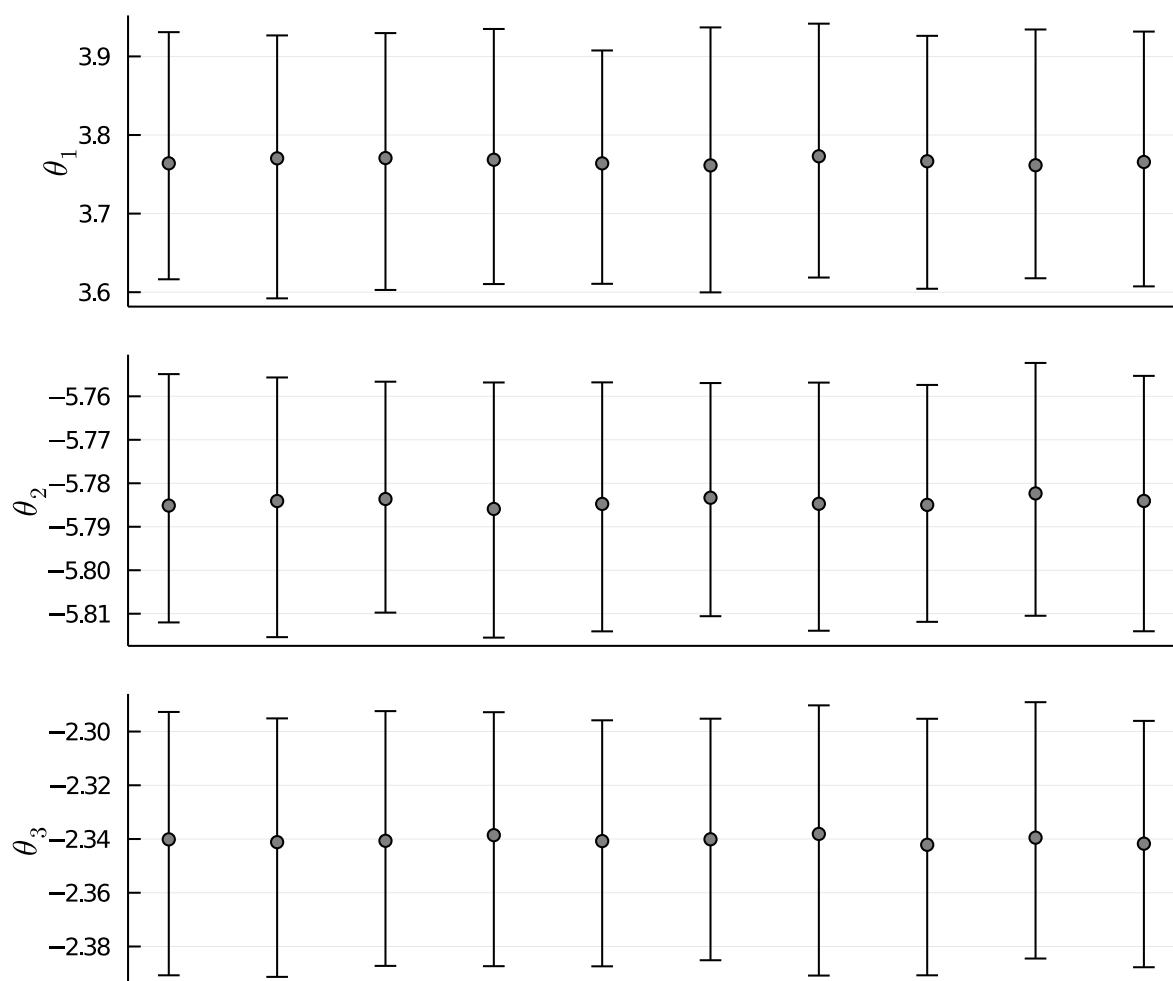
Obrázek 14: 95% konfidenční intervaly pro parametry m , s , 5000 iterací, 10 realizací simulace, HMC sampler.

Obrázek 15: Výsledná rozdělení parametrů θ_1, θ_2 a θ_3 ($2\times$) pro Gibbs sampler, 50 iterací.

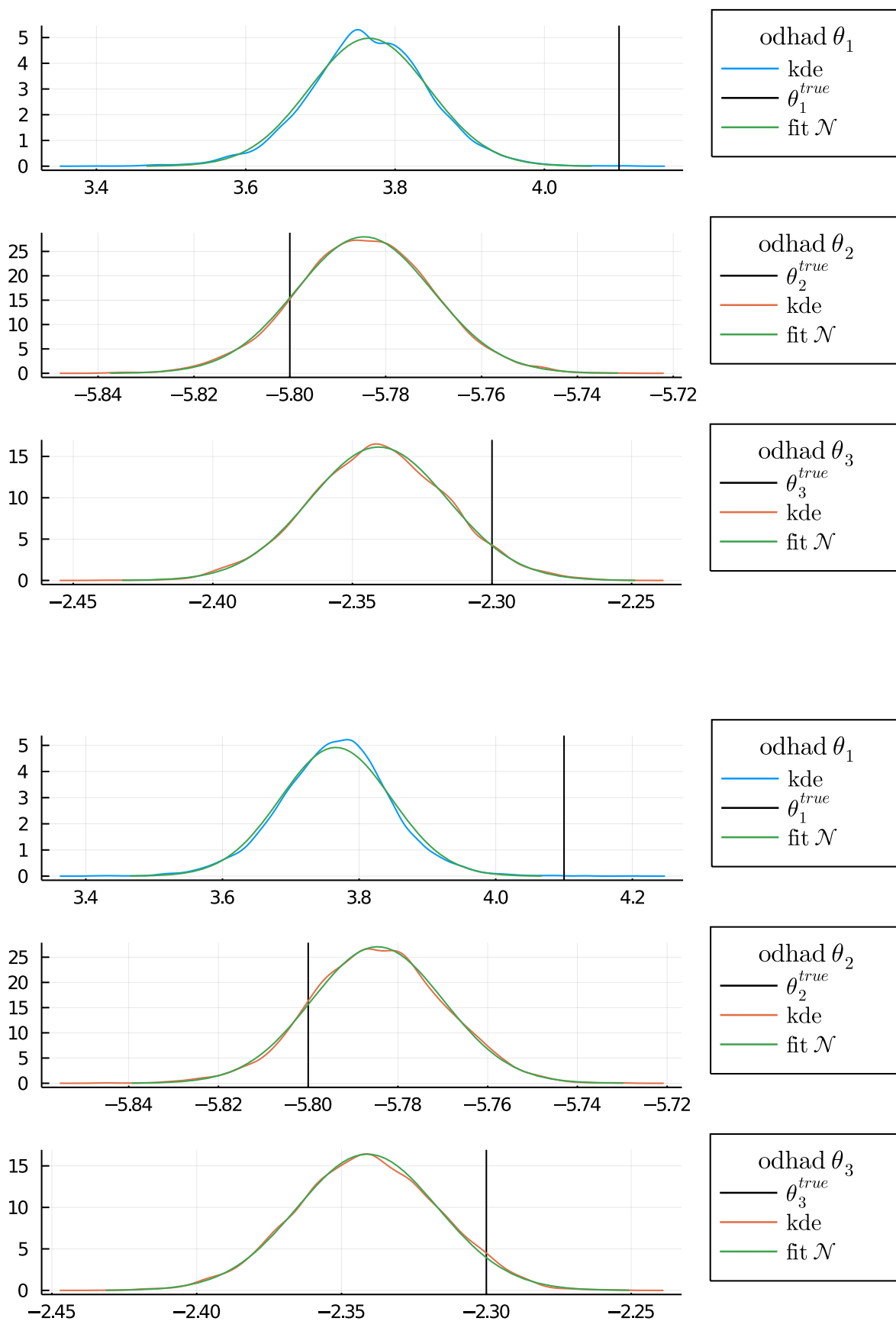


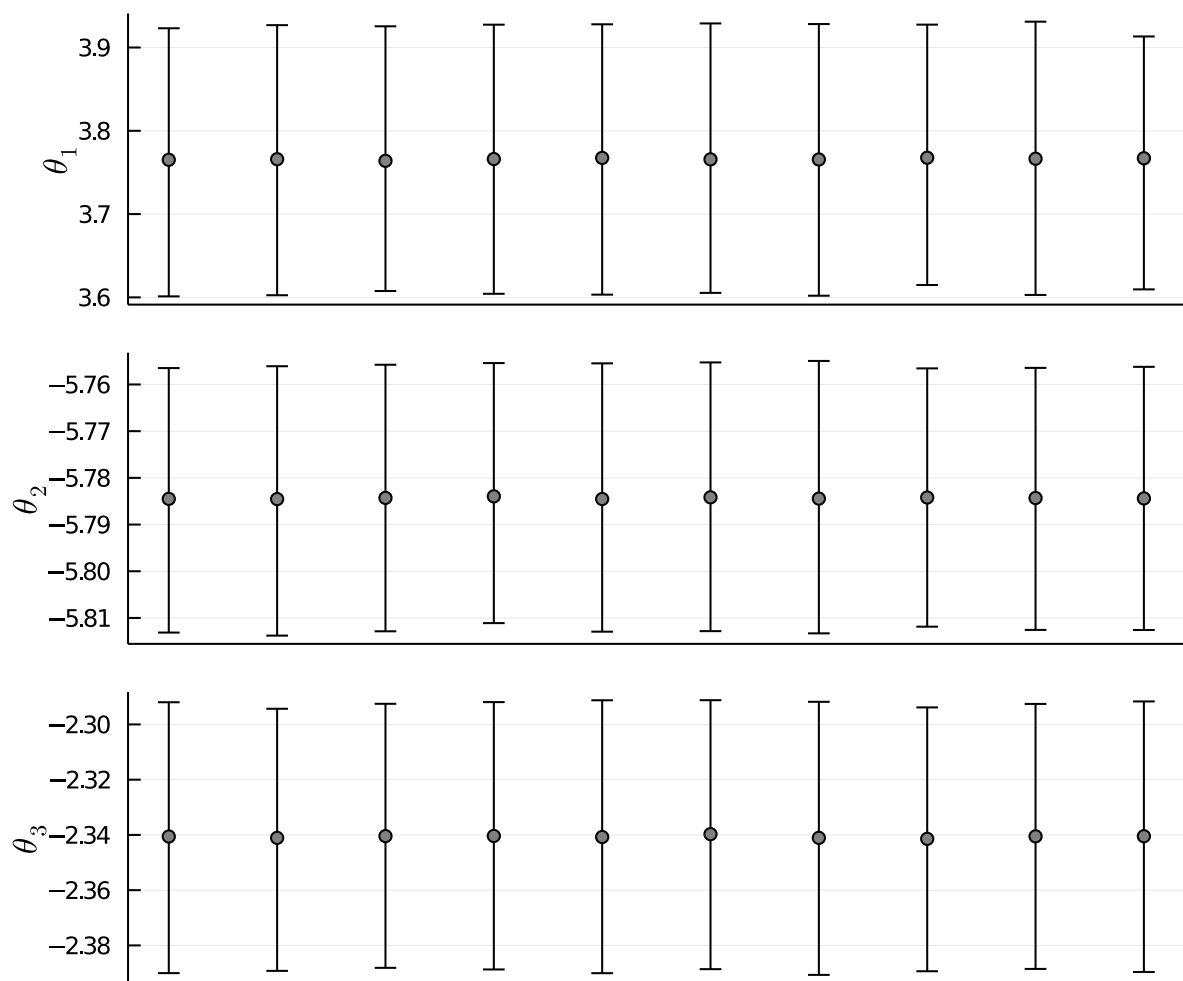
Obrázek 16: Konfidenční intervaly (95%) pro parametry θ , 50 iterací, 10 realizací simulace, Gibbs sampler.

Obrázek 17: Výsledná rozdělení parametrů θ_1 , θ_2 a θ_3 (2×) pro Gibbs sampler, 500 iterací.

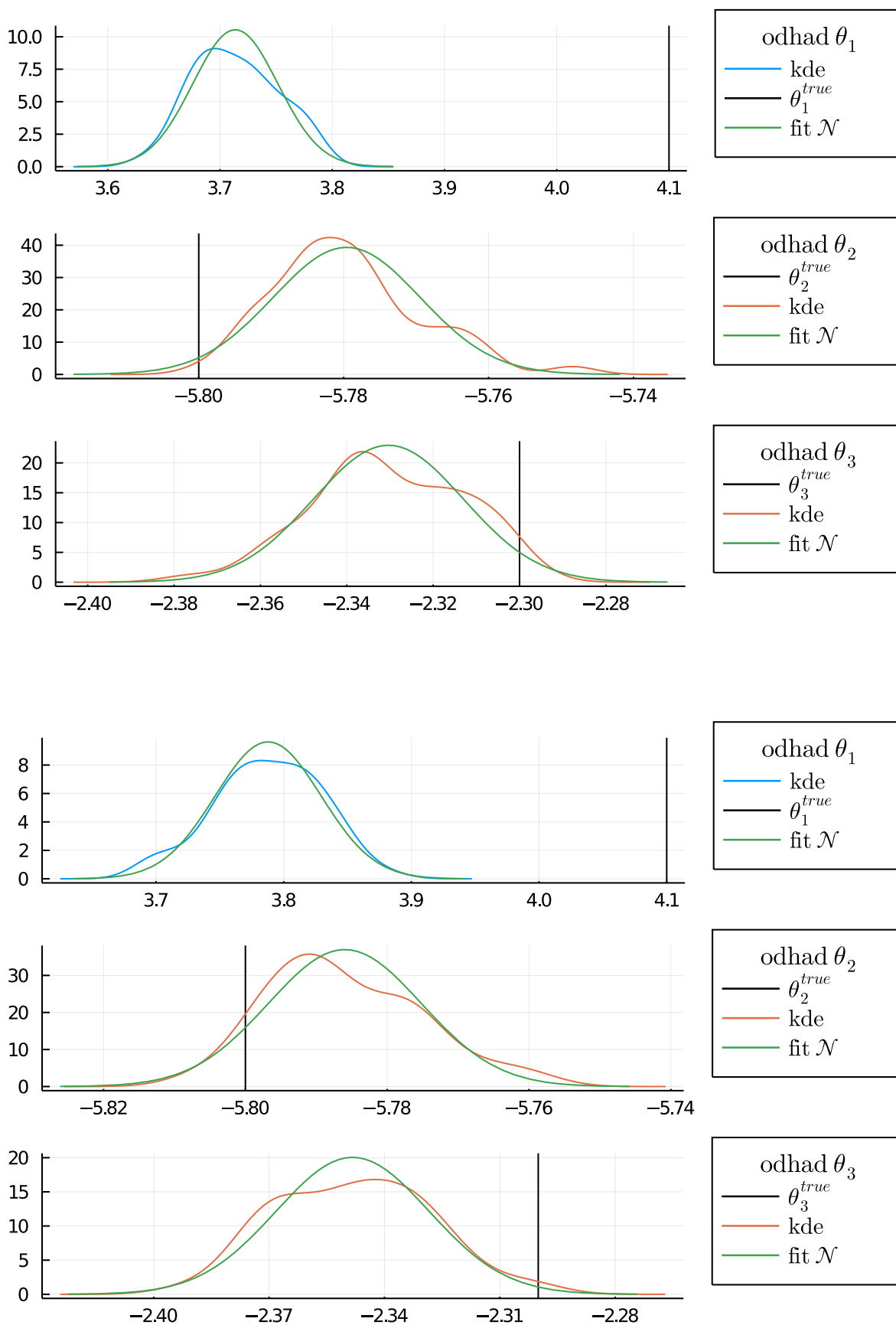


Obrázek 18: Konfidenční intervaly (95%) pro parametry θ , 500 iterací, 10 realizací simulace, Gibbs sampler.

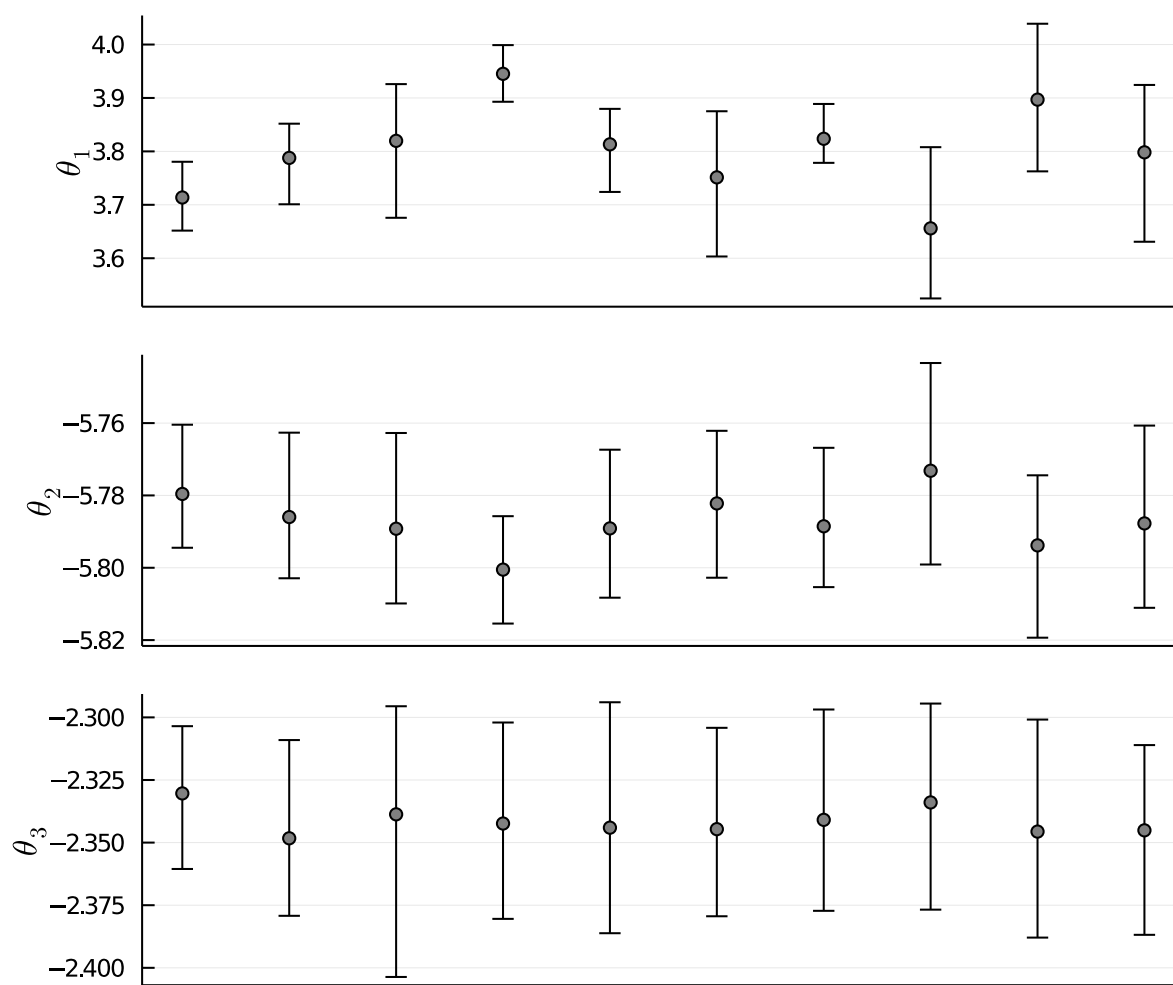
Obrázek 19: Výsledná rozdělení parametrů θ_1, θ_2 a θ_3 (2×) pro Gibbs sampler, 5000 iterací.



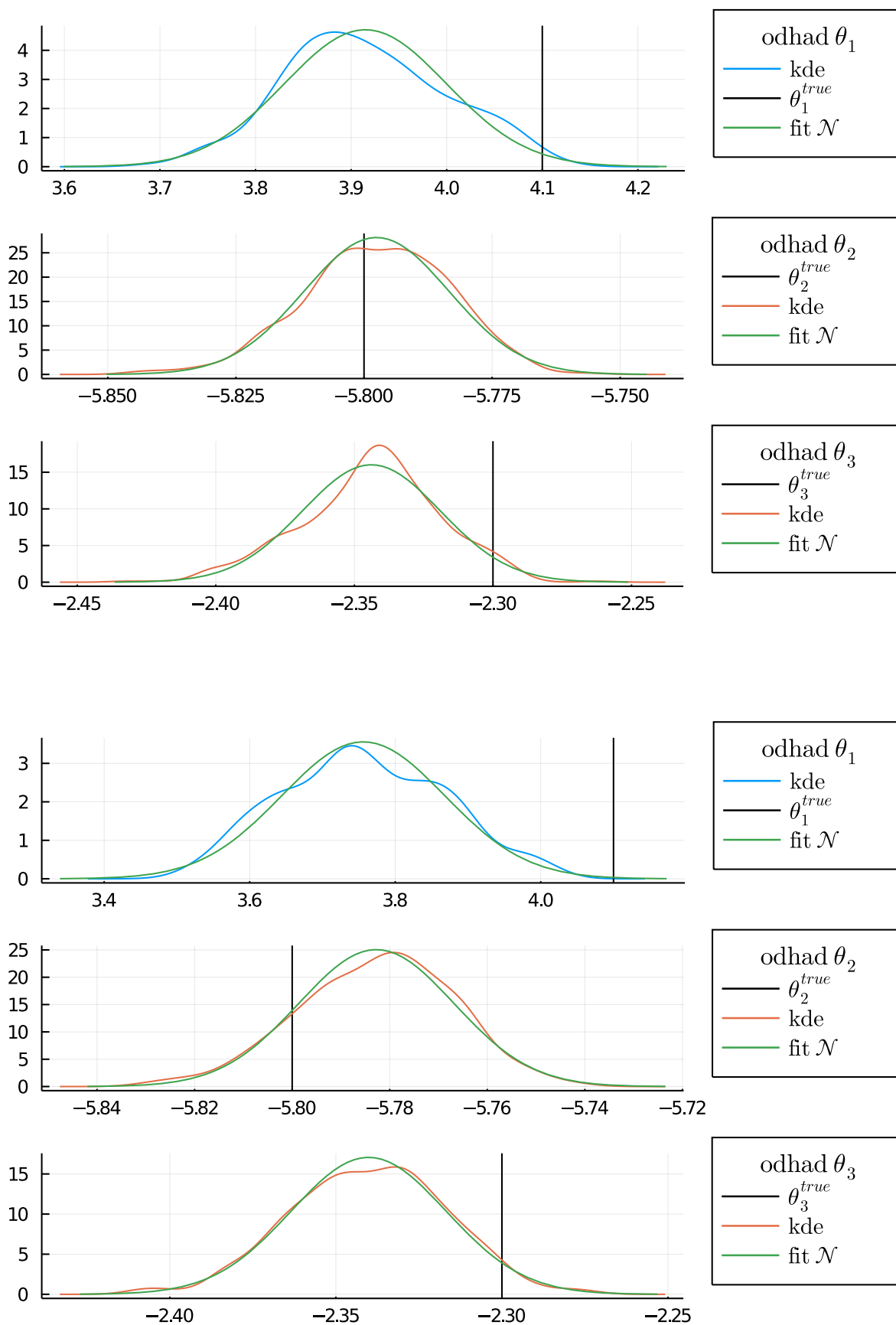
Obrázek 20: Konfidenční intervaly (95%) pro parametry θ , 5000 iterací, 10 realizací simulace, Gibbs sampler.

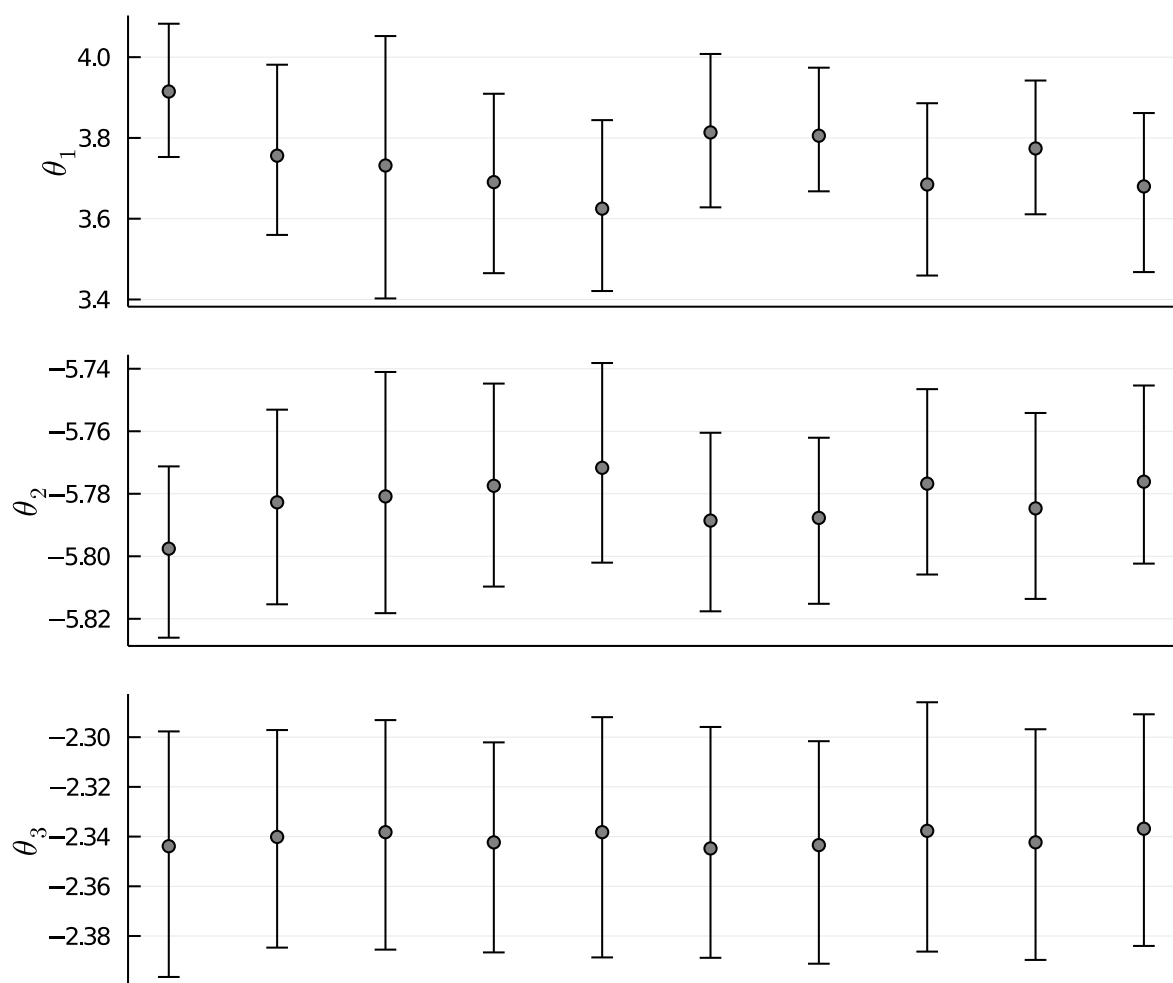


Obrázek 21: Výsledná rozdělení parametrů θ_1, θ_2 a θ_3 ($2\times$) pro HMC sampler, 50 iterací.

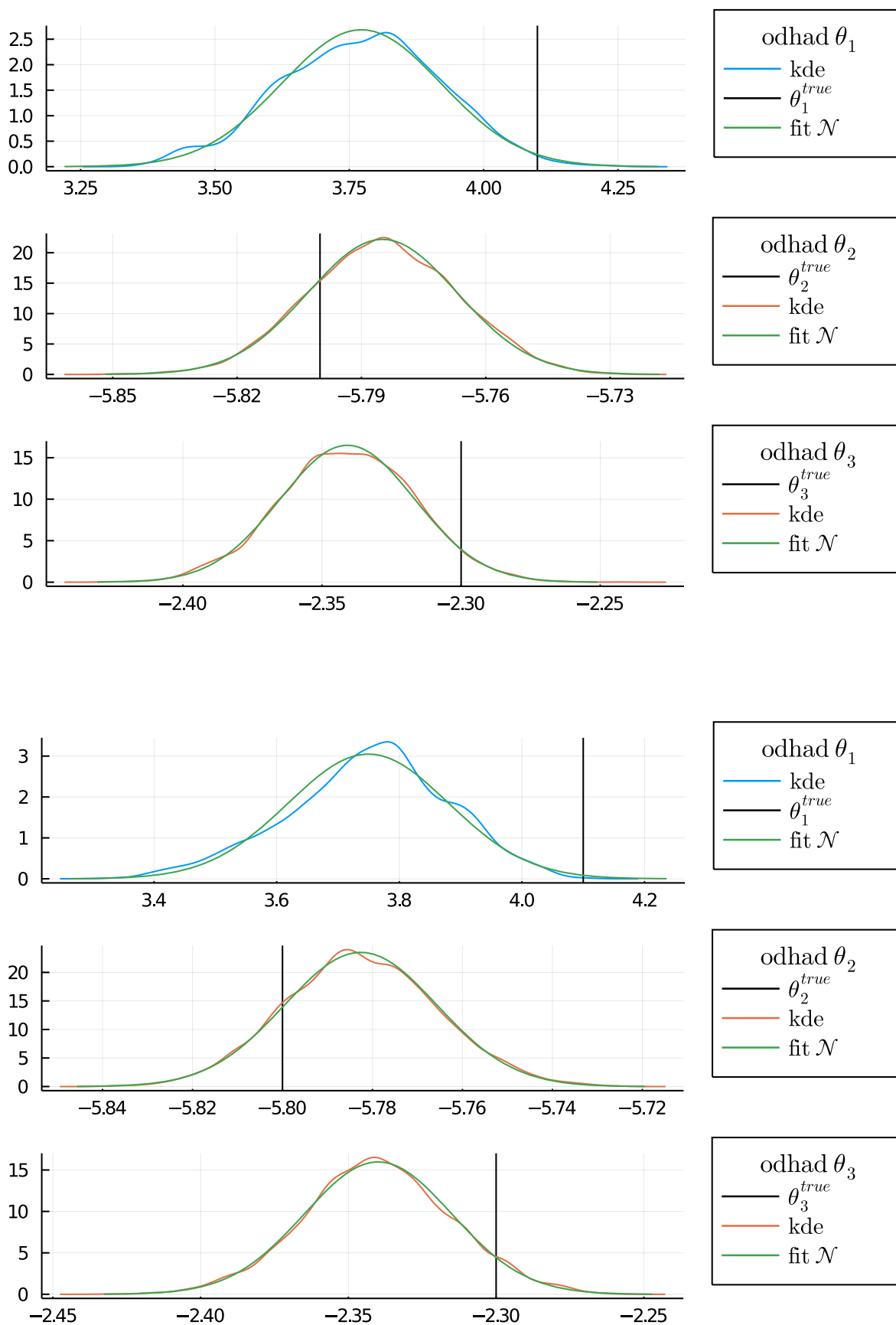


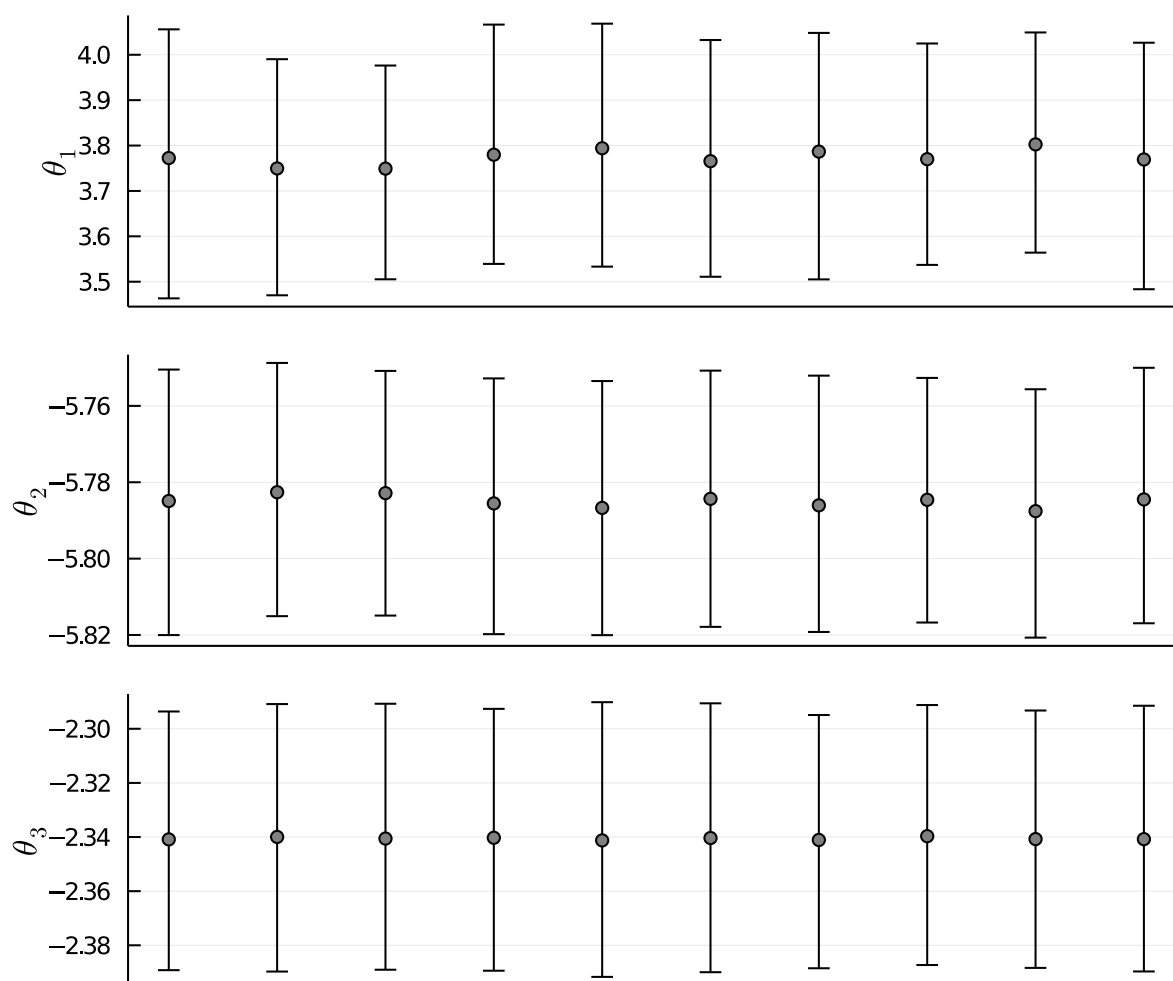
Obrázek 22: Konfidenční intervaly (95%) pro parametry θ , 50 iterací, 10 realizací simulace, HMC sampler.

Obrázek 23: Výsledná rozdělení parametrů θ_1, θ_2 a θ_3 ($2\times$) pro HMC sampler, 500 iterací.



Obrázek 24: Konfidenční intervaly (95%) pro parametry θ , 500 iterací, 10 realizací simulace, HMC sampler.

Obrázek 25: Výsledná rozdělení parametrů θ_1, θ_2 a θ_3 (2×) pro HMC sampler, 5000 iterací.



Obrázek 26: Konfidenční intervaly (95%) pro parametry θ , 5000 iterací, 10 realizací simulace, HMC sampler.