Domača naloga 5

Neža Kržan, Tom Rupnik Medjedovič

1 Cilj naloge

Imava dano pogojno porazdelitev oz. gostoto porazdelitve, iz katere ne moreva vzorčiti z običajnimi metodami. V ta namen bova uporabila algoritem Metropolis-Hastings, s pomočjo katerega bova vzorčila iz dane porazdelitve (z uporabo gostote). V najnem primeru je potrebno generirati koordinate točk, torej pare (x_i, y_i).

Nato bova še preverila kakšna je verjetnost, da velja $(x_i, y_i) \leq (1, 1)$, za vzorce velikosti 100 in izračunala pokritost 95% intervala zaupanja za to verjetnost. S tem želiva preveriti ali tudi z manjšimi vzorci (velikosti 100) dovolj dobro opišemo dano porazdelitev.

2 Generiranje vrednosti

Z uporabo algoritma Metropolis-Hastings bova generirala vrednosti iz porazdelitve, ki ima gostoto proporcionalno

$$f(x,y) = \begin{cases} x^2y^2e^{-x}e^{-y}e^{-xy}, \text{ kjer } x > 0 \text{ in } y > 0 \\ 0, \text{ sicer} \end{cases}$$

Algoritem je sestavljen iz naslednjih korakov:
Na začetku si izberemo neki začetni vrednosti x_0 in y_0, za kateri mora veljati, da je gostota večja od 0 (je možen izid). Nato na vsakem koraku s pomočjo gostote porazdelitve $g(X_p|X_i = x_i)$ predlagamo novo vrednost (x_p) pogojno na predhodnjo vrednost (x_i). Vendar pa še ne vemo ali predlagano vrednost (x_p) zares sprejmemo. Zato izračunamo verjetnost $\alpha = min\left(\frac{f(x_p)g(x_i|x_p)}{f(x_i)g(x_p|x_i)}, 1\right)$, ki nam pove verjetnost sprejema nove vrednosti, $(1-\alpha)$ pa verjetnost za ohranitev predhodnje na sledeč način

$$x_{i+1} = \begin{cases} x_p, \text{ z verjetnostjo } \alpha \\ x_i, \text{ z verjetnostjo } 1-\alpha \end{cases}$$

Za predlaganje novih vrednosti sva si izbrala

$$x_p = N(x_i, 1)$$
$$y_p = N(y_i, 1)$$

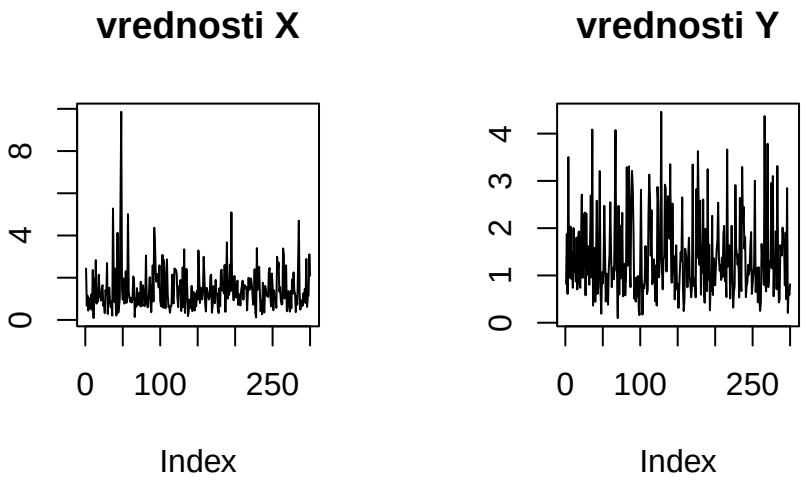
ter za gostoto porazdelitve

$$g((x_p, y_p)|(x_i, y_i)) = f_{N(x_i,1)}(x_p) \cdot f_{N(y_i,1)}(y_p),$$

kjer sta $f_{N(x_i,1)}$ in $f_{N(x_i,1)}$ gostoti $X_i = x_i$ in $Y_i = y_i$ ter s standardnim odklonom 1.

Pri generiranju podatkov moramo določiti še začetno vrednost (izbrala sva $(x_0, y_0) = (2, 2)$) in vrednosti parametrov `burn in` in `step`. Parameter `burn in` nam določi koliko začetnih vrednosti izpustimo iz vzorca

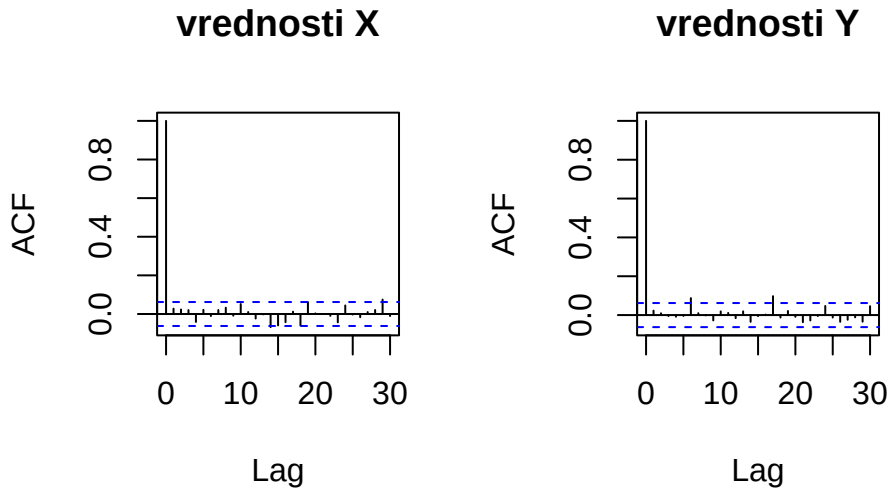
(jih ne vključimo). Vrednost tega sva določila na 100, saj vrednost dokaj hitro skonvergira. To lahko preverimo tudi grafično. Oglejmo si gibanje vrednosti za prvih 300 vrednosti iz vzorca.



Slika 1: Graf gibanja prvih 300 vrednosti za X in Y

Res lahko vidimo, da se vrednosti gibljejo znotraj pričakovanega območja (večina vrednosti je na intervalu $(0, 5)$). Pravzaprav nas v tem primeru zanima bolj ali so se vrednosti že ustalile oz. če v začetnih vrednostih ni drastičnega naraščanja ali padanja.

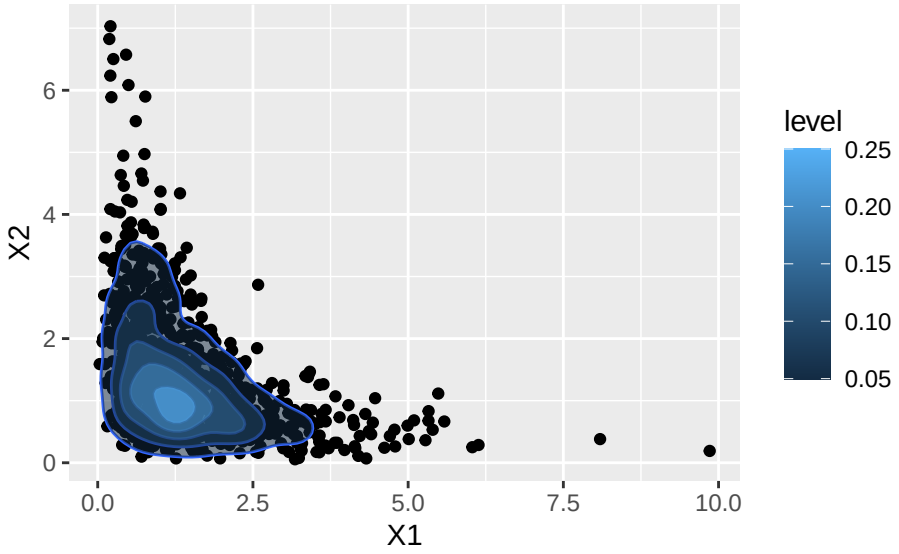
Paramater `step` nam določi koliko zaporednih vrednosti ne vključimo v vzorec. S spreminjanjem te vrednosti želimo izločiti avtokorelacijo med zaporednimi elementi v vzorcu. To vrednost sva določila na 100, saj je bila v vzorcu prisotna visoka avtokorelacija. Tudi to lahko preverimo grafično z avtokorelogramom.



Slika 2: Graf avtokorelacije za X in Y

Vidimo, da je nekaj vrednosti izven 95% intervala zaupanja, vendar se te pojavijo pri kasnejših odlogih, kar nas zares ne skrbi. Z izbrano vrednostjo parametra `step` sva odstranila avtokorelacijo med zaporednimi členi vzorca.

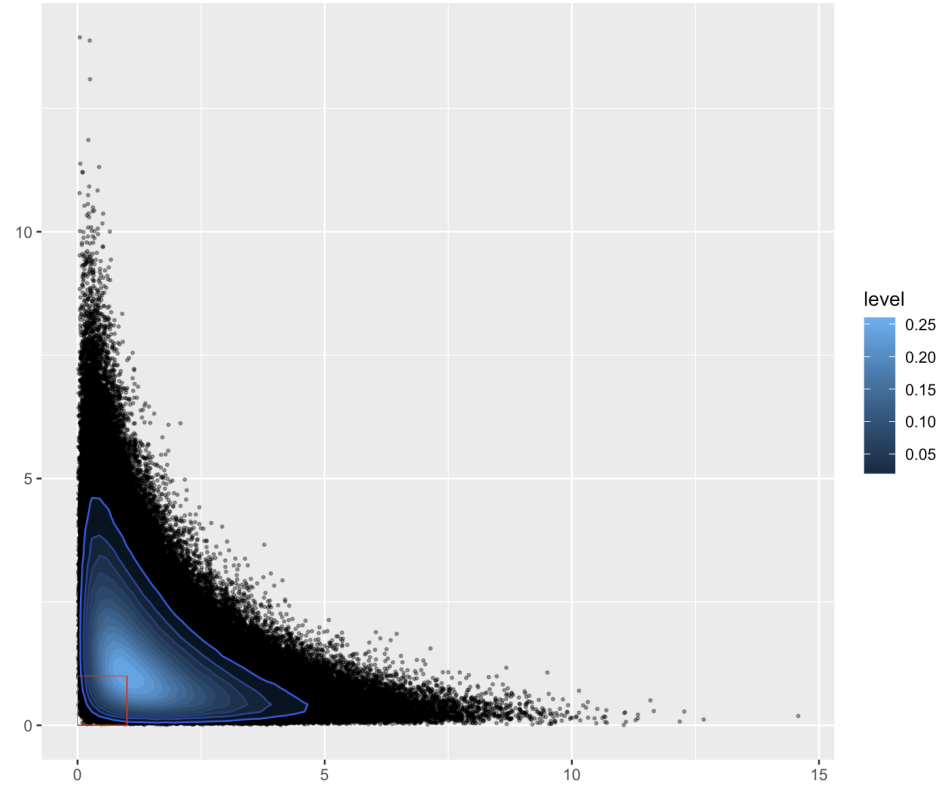
2.1 Prikaz vrednosti vzorca



Slika 3: Prikaz vrednosti iz porazdelitve za vzorec velikosti 1000

Vidimo, da je večina vrednosti tako za X kot tudi Y zgočenih na intervalu $[0, 3]$. To je tudi nekako pričakovano, saj gostota porazdelitve nekoliko spominja na gostoto eksponentne porazdelitve in so temu primerno razporejene tudi točke.

3 Verjetnost



Slika 4: Vrednosti za vzorec velikosti 1000000 z oznacenim obmocjem [0,1]x[0,1]

Želimo oceniti verjetnost, da sta obe vrednosti (X in Y) manjši od 1. Na dovolj velikem vzorcu lahko to vrednost ocenimo kot delež točk, ki se nahajajo znotraj območja $[0,1] \times [0,1]$. Ker je generiranje velikih vzorcev časovno zahtevni proces, sva zgenerirala vzorec velikosti 1000000. Ta vzorec bova uporabila kot "populacijo" iz katere bova izbrala naključne vrednosti in ustvarila manjše vzorce za potrebe simulacij.

Na tem (velikem) vzorcu, sva izračunala željeni delež (obe vrednosti sta manjši od 1) in dobila vrednost 0.091. Ker se nam ta vrednost zdi dokaj majhna glede na zgornji graf vrednosti, si oglejmo naslednjo tabelo.

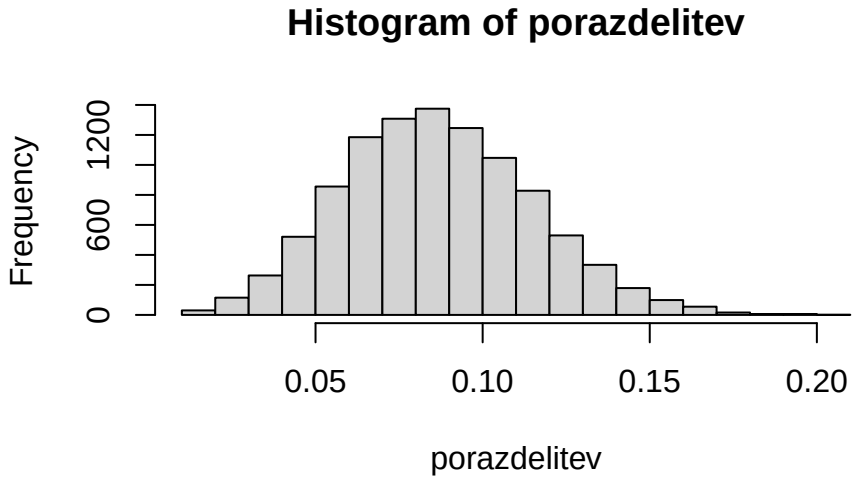
Tabela 1: Tabela razporeditev vrednosti glede na X in Y

	Y<1	1<=Y<2	2<=Y<3	3<=Y
X<1	91475	162263	94515	62827
1<=X<2	162526	155607	43673	10587
2<=X<3	94425	43579	4635	381
3<=X	62530	10534	430	13

Vidimo, da je res velik delež vrednosti, ko je vsaj ena od vrednosti (x ali y) nekoliko višja od 1.

3.1 Vzorci velikosti 100

Poglejmo si kakšna je porazdelitev zgornjega deleža v primeru, da imamo vzorce velikosti le 100. Generiramo veliko število vzorcev (npr. 10000) velikosti 100, na vsakem izračunamo verjetnost da sta tako X kot Y manjša od 1 in narišemo histogram.



Slika 5: Histogram porazdelitve verjetnosti X<1 in Y<1

Vidimo lahko da se vrednosti porazdeljujejo zelo podobno normalni porazdelitvi. V našem primeru imata parametra vrednosti μ=0.0916 in σ=0.0285.

Da se prepričamo o pravilnosti vrednosti parametrov postopek ponovimo 10-krat in rezultate prikazemo v tabeli.

Tabela 2: Tabela vrednosti parametrov za 10000 vzorcev velikosti 100

povprecje	standardni odklon
0.0915	0.0288
0.0916	0.0287
0.0918	0.0289
0.0912	0.0289
0.0914	0.0288
0.0915	0.0287
0.0918	0.0290
0.0919	0.0288
0.0915	0.0286
0.0913	0.0284

Vidimo, da so si vrednosti med seboj zelo podobne, zato smo s tem zadovoljni.

3.1.1 Pokritost

Izračunajmo še pokritost 95% intervala zaupanja za to vrednost. Interval zaupanja za povprečje porazdelitve oziroma verjetnosti, da sta tako X kot Y manjša od 1, bomo izračunali po običajni formuli $\left[\hat{\mu} - 1.96\frac{\hat{s}}{\sqrt{n}},\ \hat{\mu} + 1.96\frac{\hat{s}}{\sqrt{n}}\right]$. Vendar pa nastopi problem, saj ne poznamo "prave" vrednosti verjetnosti, saj ne poznamo parametrov porazdelitve oziroma vrednosti populacije. To lahko rešimo tako, da za "pravo" vrednost vzamemo verjetnosti delež izračunan na velikem vzorcu (v našem primeru ima 1000000 enot).

Prava vrednost deleža je enaka 0.09148.

V ta namen bomo naredili simulacijo kjer določimo število ponovitev (korakov simulacije) in števila vzorcev velikosti 100. V najnem primeru sva izbrala 1000 kot vrednost obeh parametrov. Po izvedeni simulaciji je vrednost pokritja 95% intervala zaupanja enaka 0.957

4 Zaključek

Z algoritmom Metropolis-Hastings lahko dobro generiramo vrednosti iz pogojne porazdelitve. Za izračun potrebujemo le gostoto želene porazdelitve. Pri tem moramo paziti le na pravilno izbiro začenih vrednosti (v najnem primeru (x_0, y_0)) in parametrov `burn in` ter `step`. Tudi v primeru manjših vzorecev (velikosti 100) dovolj dobro opišemo dano porazdelitev. To sva dodatno preverila s simulacijami, kjer smo ocenjevali verjetnost, da sta obe vrednosti manjši od 1. Pokritost 95% intervala je bila zelo blizu željeni vrednosti (izračunana vrednost je enaka 0.957), torej smo z izidom zadovoljni.