108: Generatorji slučajnih števil

Peter Rupnik

13. december 2018

1 Prva naloga

1.1 Naloga

Med generatorji gaussovskih slučajnih števil najdemo Box-Mullerjev generator in konvolucijski generator (6 prištetih in 6 odštetih naključnih števil iz intervala $[0, 1)$. Preizkusi oba generatorja s testom Kolmogorov-Smirnova ter χ^2-testom. Izvedi teste za različne velikosti vzorcev $(100, 1000, 10000, \ldots)$ in primerjaj rezultate. Za oba primera lahko gledaš tudi porazdelitev statistike. Izmeri tudi časovno učinkovitost algoritmov.

1.2 Implementacija

Za Box-Mullerjev generator vzamemo dve naključni vrednosti $\{x_1, x_2\}$, ki sta porazdeljeni z enakomerno porazdelitvijo med 0 in 1:

$$\text{Vzemimo } x_1,\, x_2 \sim \mathcal{U}(0,1), \tag{1}$$

$$y_1 = \sqrt{-2\ln x_1}\cos 2\pi x_2 \tag{2}$$

$$y_2 = \sqrt{-2\ln x_1}\sin 2\pi x_2 \tag{3}$$

$$\text{Tedaj velja} : y_1,\, y_2 \sim \mathcal{N}(0,1) \tag{4}$$

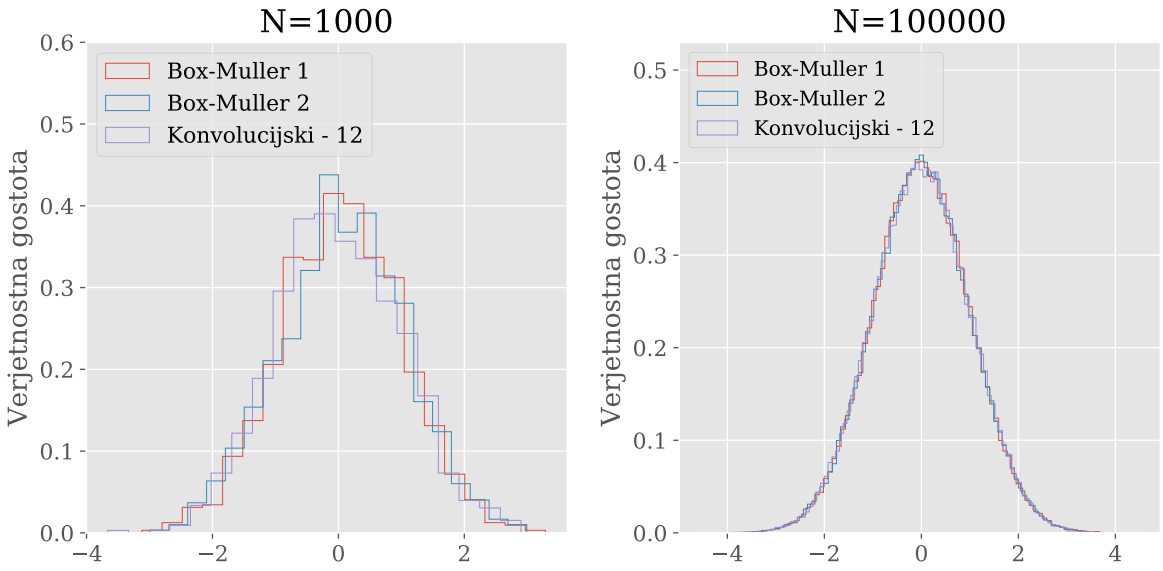
Konvolucijski generator je preprostejši: potrebujemo $2N$ naključnih števil $\{x_1, \ldots x_{2N}\}$, ki jih seštejemo skupaj in nato vsoti odštejemo N:

$$x_1, x_2 \ldots x_{2N} \sim \mathcal{U}(0,1) \tag{5}$$

$$y = \sum_{i=1}^{2N} x_i - N \tag{6}$$

$$y \sim \mathcal{N}(0,1). \tag{7}$$

Oba generatorja sem spisal v programskem jeziku `python`. Najprej sem zagrizel v Box-Mullerjev generator in spisal dvoje verzij; prva uporabi samo eno vejo generiranih normalnih števil (2), druga pa obe. Po [1] vemo, da sta obe veji neodvisni, zato pričakujemo, da velike razlike ne bo, se pa zna pokazati zaradi različne količine generiranih števil: prva verzija za N normalno porazdeljenih števil potrebuje $2N$ enakomerno porazdeljenih števil, druga verzija pa samo N. Pri konvolucijskem generatorju sem dovolil spreminjanje števila enakomerno porazdeljenih števil, s katerimi delamo konvolucijo. Pričakujem, da bo večje število $2N$ boljši generator, a seveda tudi večjo časovno zahtevnost. Privzeta vrednost je $2N = 12$.



Slika 1: Histogram števil, ki jih dobim z generatorjem Box-Mullerja in s konvolucijskim generatorjem. Oznaka v legendi: pri Box-Mullerju pove verzijo generatorja, ki nas trenutno ne briga, pri konvolucijskem generatorju pa pove, da sem za vsako generirano število naredil konvolucijo 12 enakomerno porazdeljenih števil. [Levo:] Generiram 1000 števil, ki jih popredalčkam v 20 enakomernih razdelkov. [Desno:] Generiram 100 000 števil, ki jih popredalčkam v 100 razdelkov (poskusil sem generirati tudi več števil, a so se rezultati s takim binningom tako lepo ujemali drug z drugim in na oko tudi z normalno porazdelitvijo, da že ni bilo več pedagoško, zato sem število generiranih števil zmanjšal.)

generator	n= 10	100	1000	$1{\cdot}10^4$	$1{\cdot}10^5$	$1{\cdot}10^6$
BM	0.202	0.117	0.0485	0.00594	0.00398	0.000619
BM2	0.26	0.0795	0.0513	0.00975	0.00301	0.00108
Konv12	0.326	0.0886	0.0175	0.00797	0.00369	0.00293
Konv24	0.235	0.131	0.0947	0.0914	0.0852	0.0843

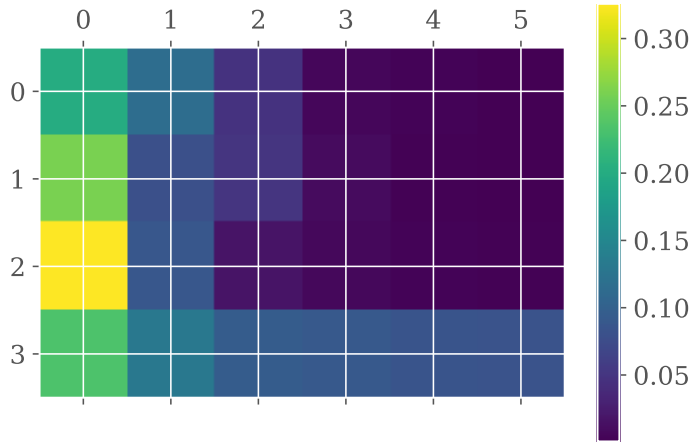


Tabela 1: V tabelo sestavljene vrednosti statistike Kolmogorova–Smirnova D, spodaj prikazane tudi grafično. Pri vseh generatorjih vrednosti padajo s številom generiranih vrednosti.

1.3 Preizkus generatorjev s testom Kolmogorov–Smirnova

Za test sem uporabil metodo `scipy.stats.kstest`, ki sprejme generiran seznam in kar string `'norm'`, pove pa nam vrednost $D_>$ in p-vrednost (iz slednje pa lahko sklepamo o kvaliteti generatorja).

1.4 Časovna zahtevnost

Časovno zahtevnost sem meril z metodo `time.clock()`, ki na Unix sistemih meri čas procesa, torej točno in zgolj tisto, kar nas zanima.

1.5 Preizkus generatorjev s statistiko χ^2

Za preverjanje, ali se generator obnaša kot pričakovano, lahko generirana števila porazporedimo v histogram, velikost stolpcev katerega označim z N_k, pričakovano velikost k-tega stolpca pa z M_k, nato pa sestavim statistiko χ^2 takole:

$$\chi^2 = \sum_k \frac{{N_k - M_k}^2}{M_k}. \tag{8}$$

Pričakovana vrednost porazdelitve in varianca porazdelitve sta enaki

$$\langle \chi^2 \rangle = \nu - 1 \tag{9}$$

$$\langle \chi^2 \rangle^2 = 2\nu - 2, \tag{10}$$

generator	n= 10	100	1000	$1{\cdot}10^4$	$1{\cdot}10^5$	$1{\cdot}10^6$	$1{\cdot}10^7$
BM	0.0012	8.1e-05	0.000313	0.0161	0.0131	0.243	1.67
BM2	0.000192	0.000105	0.00036	0.00488	0.0194	0.18	1.45
Konv12	0.000366	0.00299	0.0307	0.416	3.13	30.3	$3.06{\cdot}10^2$
Konv24	0.000315	0.00299	0.0313	0.31	3.06	31.1	$3.11{\cdot}10^2$

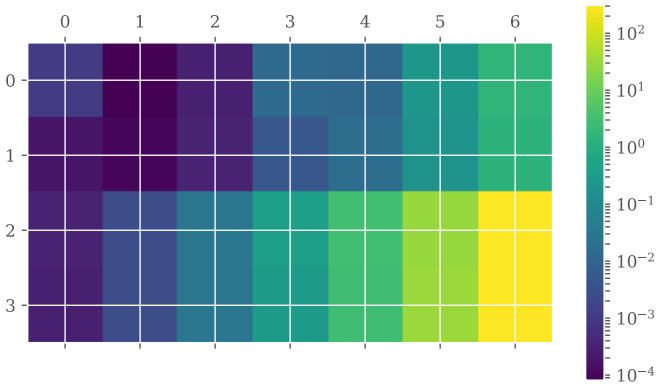
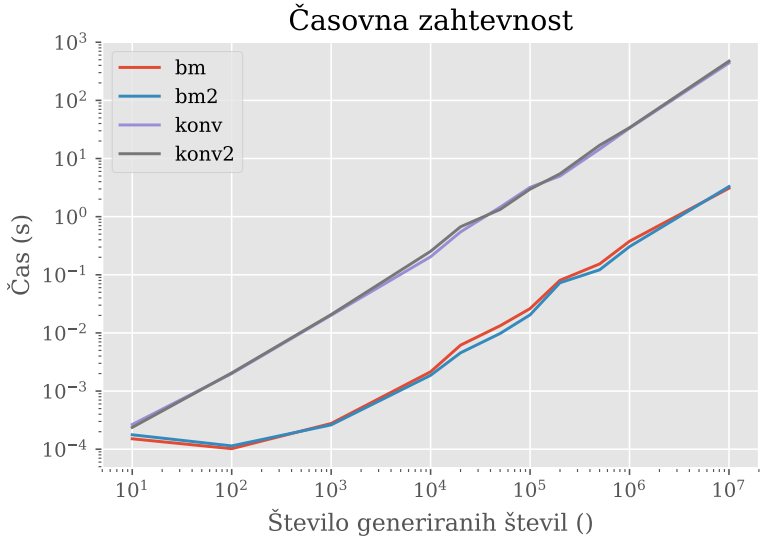
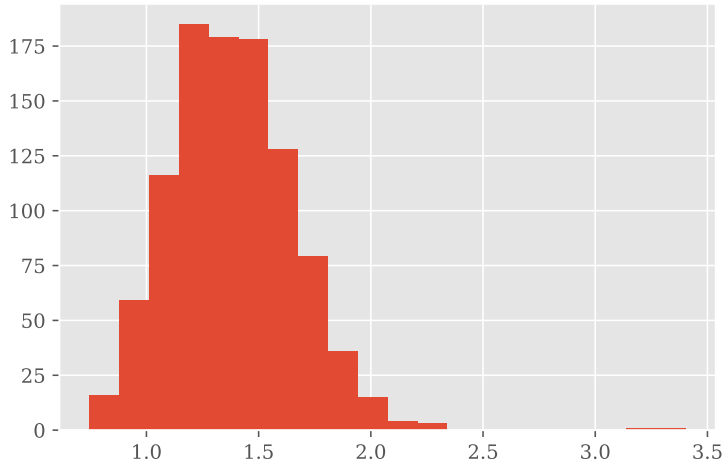


Tabela 2: V tabelo sestavljene vrednosti meritve časa posameznih generacij. [Pod tabelo]: grafični prikaz tabele, barvna skala je v logaritmičnem merilu. Ko povečujemo število generiranih slučajnih števil, konvolucijska generatorja meljeta bistveno dlje časa, pri čemer ni bistvene razlike med konvolucijo 12 ali 24 števil. Spodaj]: v log–log merilu prikazana časovna zahtevnost vseh štirih analiziranih generatorjev.





Slika 2: Porazdelitev statistike χ^2.

kjer je ν število prostostnih stopenj (v našem primeru število razdelkov.) Na žalost mi pričakovanih vrednosti $\langle \chi^2 \rangle$ ni uspelo doseči, moj glavni osumljenec je rutina `numpy.histogram`, zaradi katere sem dobil kvalitativno pričakovano porazdelitev χ^2, le kvantitativno premaknjeno za faktor cca. 10.

2 Druga naloga

2.1 Naloga

Pri dipolnem sevanju porazdelitev fotonov po prostorskem kotu ni enakomerna, pač pa je sorazmerna $\sin^2\vartheta$. Sestavi generator naključnih smeri v prostoru in generator dipolnega sevanja ter ju preizkusi. Namesto razdelitve v predalčke lahko spremljaš tudi nekaj osnovnih momentov, npr. $\langle\cos\vartheta\rangle$, $\langle\cos^2\vartheta\rangle$, oziroma splošno $\langle Y_{lm}\rangle$ za najnižje krogelne funkcije. S kakšno potenco padajo variacije v momentih, ko povečujemo velikost vzorca?

2.2 Matematično ozadje problema

Želimo si porazdelitev, ki bo enakomerno razmazana po obodu sfere. V ta namen zapišemo porazdelitev verjetnosti po površini sfere:

$$\frac{\mathrm{d}P}{\mathrm{d}\Omega} = \frac{\mathrm{d}P}{\sin\vartheta\,\mathrm{d}\vartheta\,\mathrm{d}\phi} = \frac{\mathrm{d}P}{\mathrm{d}\phi\,\mathrm{d}(\cos\vartheta)} = \frac{1}{4\pi}, \tag{11}$$

kar pa lahko faktoriziramo takole:

$$\frac{\mathrm{d}P}{\mathrm{d}\phi} = \frac{1}{2\pi} \implies \phi = 2\pi\mathcal{U}_1 \tag{12}$$

$$\frac{\mathrm{d}P}{\mathrm{d}(\cos\vartheta)} = \frac{1}{2} \implies \vartheta = \arccos(2\mathcal{U}_2 - 1), \tag{13}$$

kjer sta $\mathcal{U}_{1,2}$ enakomerni porazdelitvi na intervalu $[0, 1)$.

Za drugi del, generator slučajnih smeri porazdelitve dipolnega sevanja, lahko postopamo enako in dobimo:

$$\frac{\mathrm{d}P}{\mathrm{d}\phi} = \frac{1}{2\pi} \implies \phi = 2\pi\mathcal{U}_1 \tag{14}$$

$$\frac{\mathrm{d}P}{\mathrm{d}\left(\cos(\vartheta)\right)} = \frac{3}{4}\sin^2\vartheta, \tag{15}$$

kar pa nam da zoprno enačbo

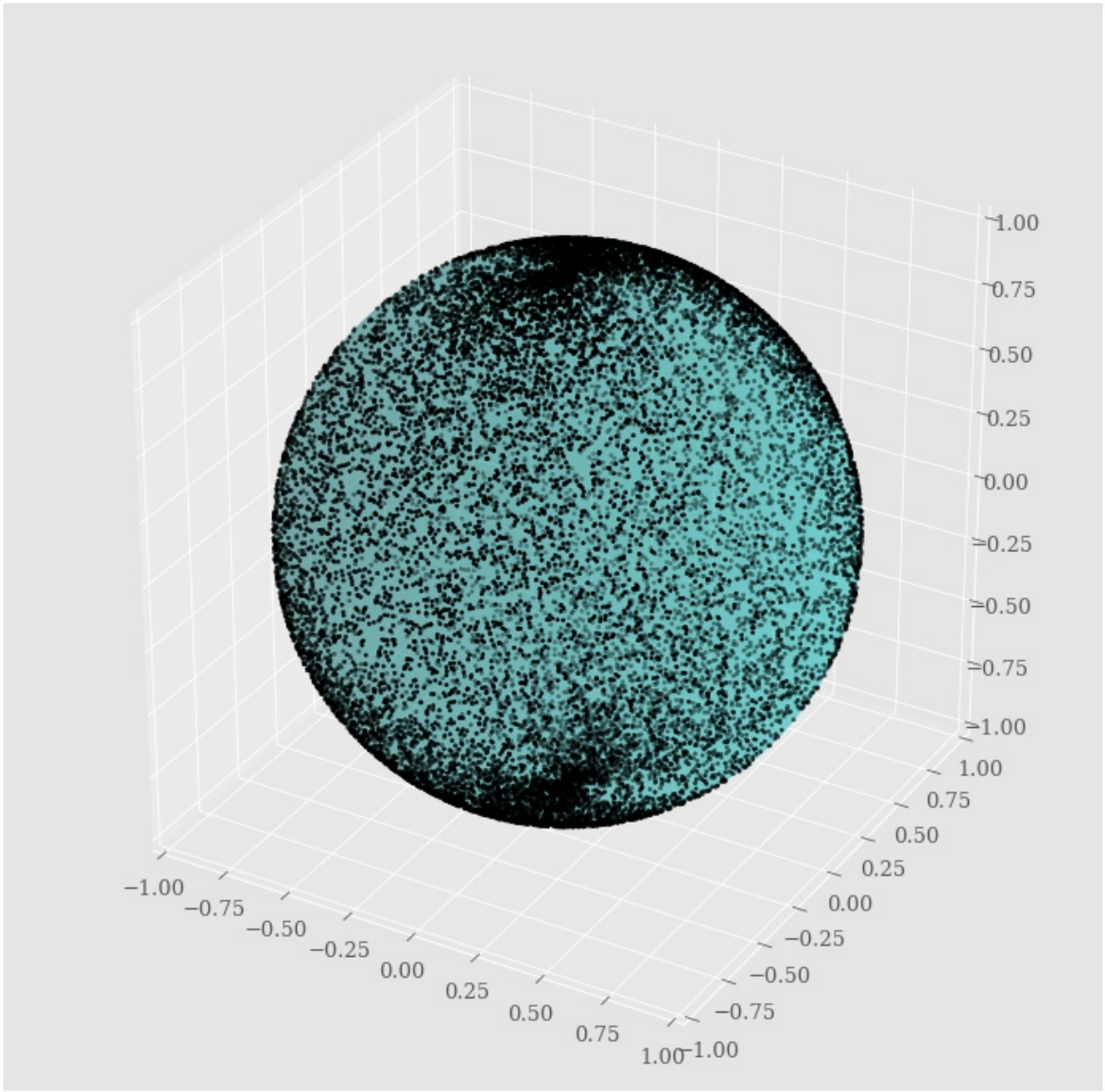
$$\mathcal{U}_2 = \frac{1}{4}\left[3\cos(\vartheta) - \cos^3(\vartheta)\right] + \frac{1}{2}, \tag{16}$$

česar pa analitično ne znam invertirati, lahko pa to težavo premostimo na dva načina: lahko ϑ izračunamo numerično, ali pa posežemo po metodi z zavračanjem, kot smo jo omenili na vajah. Nameravam predstaviti oba procesa.

2.3 Izplen

S simulacijo naključnih smeri v prostoru ni bilo težav. Napisal sem funkcijo, ki mi je po opisanih zvezah iz enakomerne porazdelitve transformirala kota v pravilno porazdelitev, poleg tega pa je generirane smeri pretvorila v kartezično obliko in jih prikazala na enotski krogli. Rezultat žrebanja smeri v prostoru prikazuje slika 3. Če nas zanima, kako dober generator smo naredili, lahko najprej pogledamo povprečje koordinat 100 000 generiranih točk:

$$\begin{bmatrix} \langle x\rangle \\ \langle y\rangle \\ \langle z\rangle \end{bmatrix} = \begin{bmatrix} -0.0005 \\ 0.0011 \\ -0.0002 \end{bmatrix}. \tag{17}$$



Slika 3: Vsaka točka označuje eno izmed 18k smeri v prostoru. Tokrat prvič oddajam sliko v formatu `.jpg`, saj vektorski format izriše prav vsako točko, zaradi česar je velikost vektorskega ekvivalenta te slike narasla za nekaj redov velikosti v primerjavi z običajnimi grafi.

Povprečja se zdijo sumljivo blizu ničle, s čimer smo *zufrieden.*[2]

Še vedno pa se nam morda zdi, da sta 'severni' in 'južni' pol naše enotske krogle bolj kosmata kot preostanek[1]. Da se prepričamo, ali generator deluje, lahko preverimo še porazdelitve polarnega in azimutalnega kota ϕ in ϑ. To sem naredil na sliki 4.

Nadaljujem z generacijo dipolnega sevanja, najprej z *brute force* iskanjem ničle transcendenčne enačbe (16). Tega sem se lotil z bisekcijo in dobil porazdelitev na sliki 5.

Za preverjanje generatorjev popravim funkcije, ki so mi doslej vračale kartezične točke na obodu krogle in jih izrisovale. Zanimajo me povprečja funkcij in količin, ki jih bom izračunal takole:

$$\langle Y \rangle = \frac{\int Y \frac{\partial P}{\partial \Omega} \, \mathrm{d}\Omega}{\int \frac{\partial P}{\partial \Omega} \, \mathrm{d}\Omega}, \tag{18}$$

kjer z Y označujem splošno funkcijo.

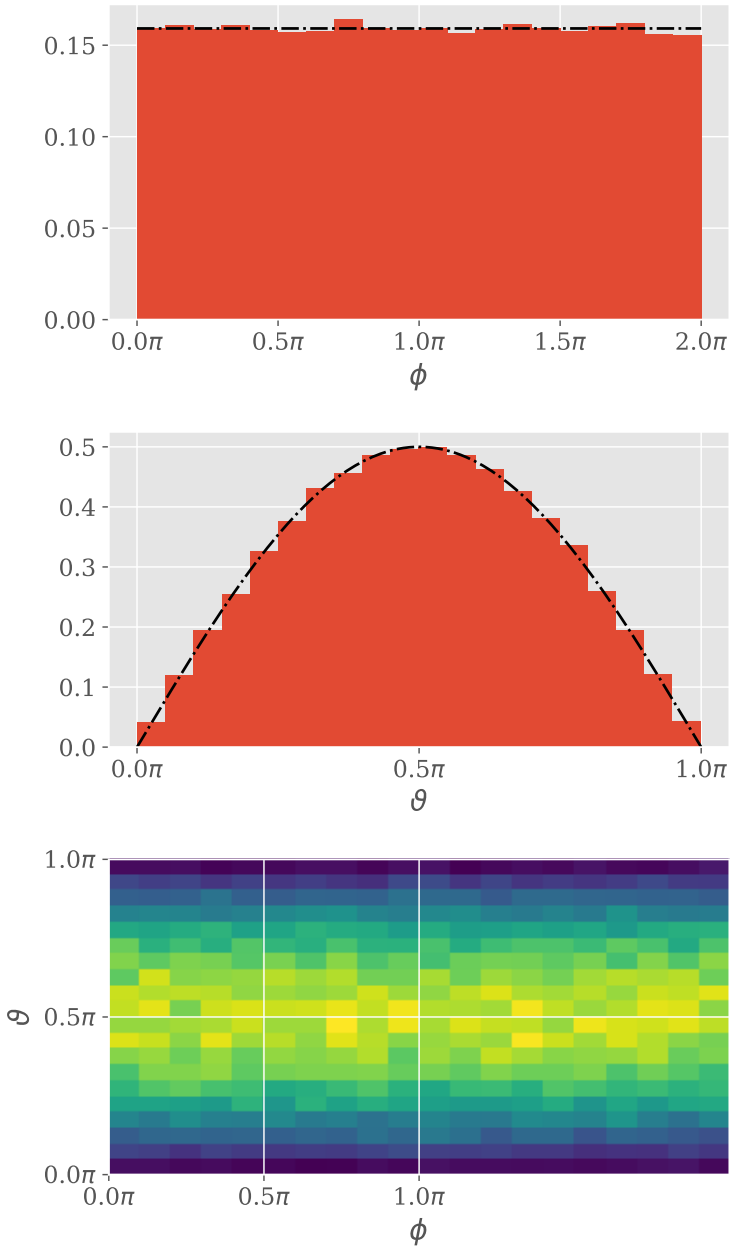
Za generator smeri v prostoru za 100 000 generiranih smeri dobim:

količina	izračunana vrednost	pričakovana vrednost
$\langle \phi \rangle$	3.141598	3.141592
$\langle \vartheta \rangle$	1.57214	1.570796
$\langle \cos(\vartheta) \rangle \propto \langle Y_1^0 \rangle$	0.00253	0
$\langle Y_1^1 \rangle$	$7.916 \cdot 10^{-5} - 0.0011i$	0
$\langle Y_1^1 \rangle$	0.0018	0

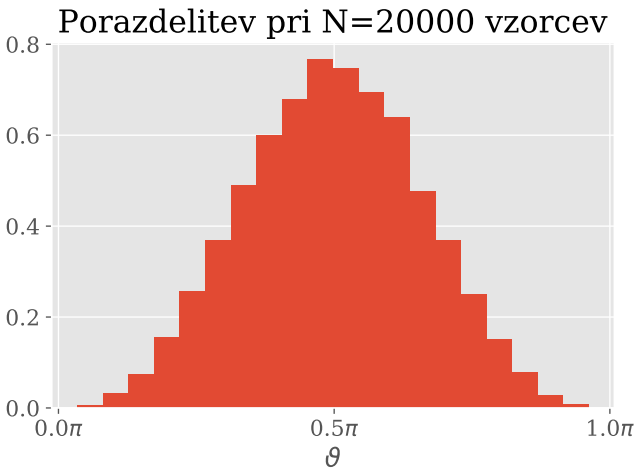
Za generator dipolnega sevanja za 100 000 generiranih smeri pa dobim:

količina	izračunana vrednost	pričakovana vrednost
$\langle \phi \rangle$	3.14282	3.141592
$\langle \vartheta \rangle$	1.571980	1.570796
$\langle \cos(\vartheta) \rangle \propto \langle Y_1^0 \rangle$	-0.00103	0
$\langle Y_1^1 \rangle$	$.00018 - 8.47 \cdot 10^{-5}i$	0
$\langle Y_1^1 \rangle$	-0.00050	0

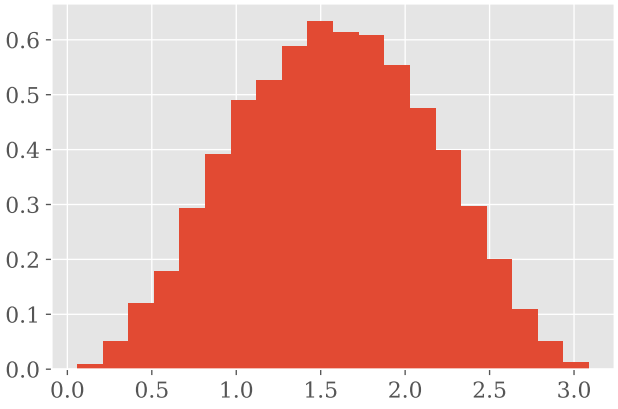
[1] Dejansko izgledata bolj kosmata, vendar je to zaradi prosojnosti turkizne enotske krogle (`alpha=0.3`)



Slika 4: Porazdelitve (normalizirane) generiranih kotov ϕ in ϑ. Število generiranih smeri je 100 000. [Zgoraj:] porazdelitev polarnega kota, s črtkano črto označena pričakovana porazdelitev. [Sredina:] porazdelitev azimutalnega kota, s črtkano črto označena pričakovana porazdelitev. [Spodaj:] porazdelitev obeh kotov, nekesortna Mercatorjeva projekcija gostote točk na naši krogli.



Slika 5: Porazdelitev azimutalnega kota ϑ pri generatorju smeri sevanja dipola. Porazdelitev je bila dobljena z iskanjem inverza funkcije z bisekcijo.



Slika 6: Porazdelitev azimutalnega kota ϑ pri generatorju smeri sevanja dipola. Porazdelitev je bila dobljena z izločanjem pri manjšem številu vzorcev kot slika 5.

3 Tretja naloga

3.1 Naloga

Datoteke `mod_tmxx_yyy.dat` vsebujejo čase oddaje nalog iz Modelske analize (xx – letnik, 10,11,13,14; yyy – številka naloge) v formatu d:hh:mm, merjeno od četrtka ob polnoči. Minus v podatkih pripada številki dneva – -1:20:30 pomeni sredo ob pol devetih zvečer.

S testom Kolmogorov-Smirnova preveri, če so kumulativne porazdelitve za posamezne naloge med seboj statistično enake in če se letniki statistično razlikujejo med seboj. S kolikšno verjetnostjo lahko določimo iz katerega leta prihajajo podatki za izbrano nalogo? Če je potrebno, upoštevaj, da se pogoji za nekatere naloge razlikujejo (novoletni prazniki, zadnja naloga semestra, ...) in jih izloči iz statistike.

3.2 Pristop

Najprej sem prenesel podatke in jih razpakiral iz formata `tar.gz`, in sicer kot tajnice, torej z grafičnim vmesnikom in brez skript. Nekoč bom naštudiral, kako to napraviti v dveh vrsticah programske kode, a danes še ni ta dan...

Nato sem imena vseh datotek razvrstil glede na njihov letnik. Najprej sem pogledal, kako izgledajo kumulativne porazdelitve časov oddajanja nalog za cela leta naenkrat.

Nekaj podatkov:

```
Leto: 2010, dolžina areja: 139, povprečje: -25.6 min
Leto: 2011, dolžina areja: 192, povprečje: -28.3 min
Leto: 2013, dolžina areja: 168, povprečje: -45.6 min
Leto: 2014, dolžina areja: 209, povprečje: -63.2 min

Kolmogorov-Smirnov za leto 2011 in 2010:
        D=0.196, p-vrednost = 0.003369
Kolmogorov-Smirnov za leto 2013 in 2010:
        D=0.248, p-vrednost = 0.0001325
Kolmogorov-Smirnov za leto 2013 in 2011:
        D=0.347, p-vrednost = 4.818e-10
Kolmogorov-Smirnov za leto 2014 in 2010:
        D=0.271, p-vrednost = 6.386e-06
Kolmogorov-Smirnov za leto 2014 in 2011:
        D=0.363, p-vrednost = 3.777e-12
Kolmogorov-Smirnov za leto 2014 in 2013:
        D=0.111, p-vrednost = 0.186
```

P-vrednost v tem testu pomeni stopnjo tveganja, na kateri ne moremo zavreči hipoteze da sta porazdelitvi enaki. Zaradi majhnih p-vrednosti ne morem trditi, da so celoletne porazdelitve časov oddajanja enake, vsaj ne pri razumnih stopnjah tveganja.

Račun ponovim tudi za porazdelitve časov oddajanja za posamezne naloge.

```
Naloga: 101, dolžina areja: 75, povprečje: -41.4 min
Naloga: 102, dolžina areja: 67, povprečje: -46.8 min
Naloga: 103, dolžina areja: 61, povprečje: -30.7 min
Naloga: 104, dolžina areja: 64, povprečje: -36.4 min
```

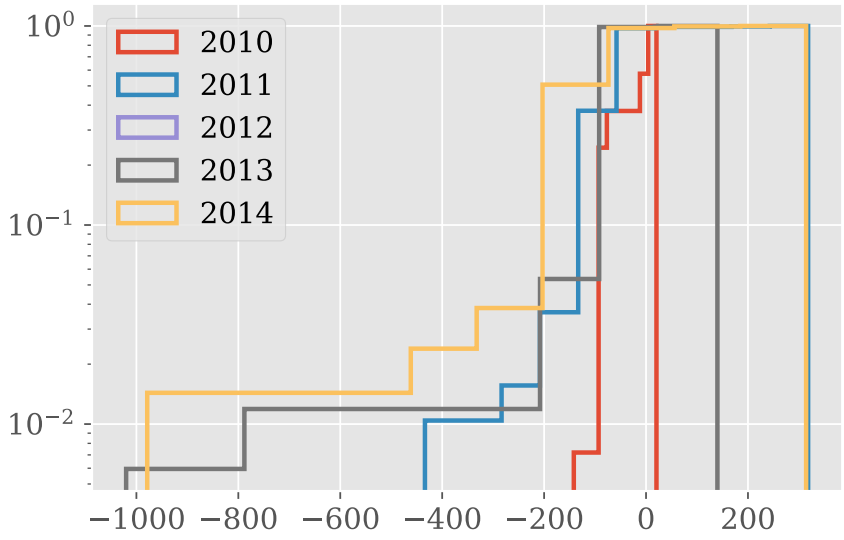
```
Naloga: 105, dolžina areja: 65, povprečje: -29.5 min
Naloga: 106, dolžina areja: 55, povprečje: -33.9 min
Naloga: 107, dolžina areja: 54, povprečje: -38.9 min
Naloga: 108, dolžina areja: 50, povprečje: -14.7 min
Naloga: 109, dolžina areja: 57, povprečje: -53.4 min
Naloga: 110, dolžina areja: 47, povprečje: -40.6 min
Naloga: 111, dolžina areja: 42, povprečje: -66.2 min
Naloga: 112, dolžina areja: 42, povprečje: -32.5 min
Naloga: 113, dolžina areja: 29, povprečje: -1.29e+02 min
```

Tu so rezultati testa Kolmogorova–Smirnova bolj zanimivi: pojavita se oba ekstrema, zelo podobne porazdelitve in zelo različne porazdelitve. Prilagam nekaj kombinacij, ki so dale ekstremne p-vrednosti:

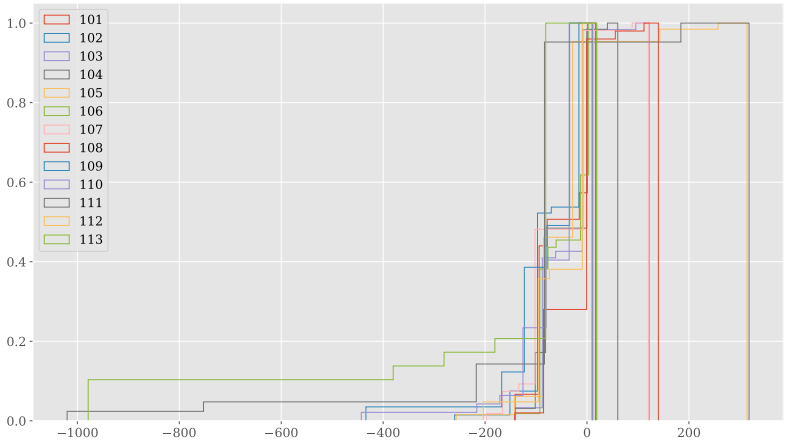
```
naloga 108 in 102:      D: 0.286567, p-vrednost: 0.014125
naloga 109 in 108:      D: 0.281754, p-vrednost: 0.023039
naloga 108 in 101:      D: 0.253333, p-vrednost: 0.034839
naloga 108 in 105:      D: 0.230769, p-vrednost: 0.083878

naloga 110 in  104:     D: 0.095745,  p-vrednost: 0.955799
naloga 107 in  106:     D: 0.093939,  p-vrednost: 0.961589
naloga 106 in  105:     D: 0.088112,  p-vrednost: 0.968082
naloga 110 in  106:     D: 0.089749,  p-vrednost: 0.982248
naloga 107 in  101:     D: 0.079259,  p-vrednost: 0.985718
naloga 107 in  105:     D: 0.081481,  p-vrednost: 0.986129
naloga 110 in  103:     D: 0.081967,  p-vrednost: 0.991760
naloga 106 in  103:     D: 0.075708,  p-vrednost: 0.994889
```

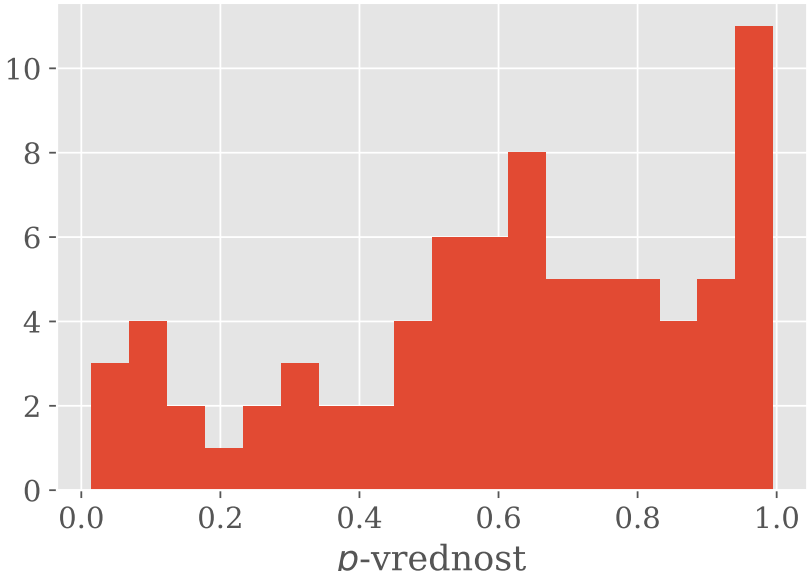
V splošnem tako ne morem trditi, da se časi oddaje pokoravajo enakim zakonitostim, vendar pa obstajajo naloge, katerih časi oddaj so porazdeljeni po zelo podobnih porazdelitvah.



Slika 7: Kumulativna verjetnostna gostota časov oddajanja nalog za posamezna leta. Da se lepše vidijo repi daleč v $t \to -\infty$, rišem ordinato v logaritemskem merilu.



Slika 8: Kumulativna verjetnostna gostota časov oddajanja nalog za posamezne naloge čez vsa leta.



Slika 9: Porazdelitev p-vrednosti za vse kombinacije (razen za kombinacijo, ko primerjamo čase oddaje same s sabo).

Literatura

[1] https://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform, dostopno 13. december 2018.

[2] Modelsko—analitična terminologija prof. Širce s predavanj.