Problem najvećeg kvadrata i Gumbelova distribucija

Ivan Čulin i Joško Kristić

Srpanj 2021

1 Kratki uvod u zadatak

U kvadratu $[0,1]^2$ simuliramo n uniformno distribuiranih točaka. Neka je U neka simulirana točka. Tada definiramo slučajnu varijablu R_U kao stranicu najvećeg kvadrata koji se nalazi unutar $[0,1]^2$ i ne sadrži nijednu drugu točku, a točka U mu se nalazi na donjoj stranici. Prvi korak u zadatku je pronaći algoritam koji će za n točaka iz zadatka pronaći vrijednost od R_{U_i} za svaki $i = 1, 2, \ldots, n$.
Nakon toga, za proizvoljne n i M izvršavamo gornju simulaciju od n točaka ukupno M puta i bilježimo vrijednost

$$M_m = \max_{m=1,\ldots,M} R_{U_i}$$

Tako dobivene vrijednosti M_m sada transformiramo zadanom transformacijom $M_m \to nM_m^2 - \log n - \log\log n$ i uspoređujemo sa Gumbelovom razdiobom. Usporedbu ćemo provesti koristeći Kolgomorov-Smirnovljev test.

2 Algoritam za pronalazak R_U

Neka je (x, y) točka za koju tražimo R_U. Od ostalih točaka su nam u interesu samo točke s većom y koordinatom. Neka je iznad (x, y) ukupno k točaka i neka je vektor svih točaka na početku poredan uzlazno po y koordinati.
Neka nakon i-tog koraka M predstavlja stranicu najvećeg kvadrata koji se može konstruirati ako se iznad točke (x, y) nalazi samo prvih i točaka. Tada je niz vrijednosti koje M poprima po koracima padajući jer dodavanje jedne točke može samo smanjiti ili ostaviti istu stranicu najvećeg kvadrata.
Na početku i-tog koraka u varijable L i R spremimo najbliže x kordinate prvih $i-1$ točaka iznad (x, y) s lijeva i s desna. Prije nego počnemo s prvim korakom postavimo $L = 0$, $R = 10$ i $M = 10 - y$.

Pretpostavimo da se nalazimo u i-tom koraku i radimo na točki (x_i, y_i). Ako je $y_i = y$ prelazimo na sljedeću točku jer i-ta točka ne može promijeniti stranicu

do sada najvećeg kvadrata.
Ako točka ima x koordinatu manju od L ili veću od R ona nikako ne može promijeniti stranicu do sada najvećeg kvadrata jer uvijek dodajemo točke s većom y kordinatom, a točka koja ima veću y kordinatu od svih prijašnjih može jedino smanjiti kvadrat ako je bliža po x koordinati točki (x, y) nego sve prijašnje točke.
Ako točka ima x koordinatu veću od L ili manju od R tu vrijednost spremamo u R ili L i krećemo analizirati tu točku.

Ta točka može smanjiti stranicu do sada najvećeg kvadrata u smislu da se na nju moramo nasloniti s gornjom stranicom ili s nekom od bočnih stranica novog najvećeg kvadrata.
Ako se novi najveći kvadrat na i-tu točku naslanja s gornjom stranicom, onda se prijašnji najveći kvadrat nije na ništa naslanjao s gornjom stranicom (osim eventualno na gornji rub glavnog kvadrata) jer bi to značilo da je stranica najvećeg kvadrata narasla u koraku (jer je $y_i > y_k$ za svaki $k < i$), a to nije moguće.
Dakle, prijašnji najveći kvadrat je bio naslonjen bočno s obe stranice. U ovom se slučaju onda stranica kvadrata smanjuje na $y_i - y$, a to znači da je $y_i - y$ veći od $R - L$ pa nam se to i sprema u varijablu $val = \max(R - L, pY[j] - pY[i])$.
Ako nas i-ta točka ograniči u smislu da se na nju moramo nasloniti bočno tada se stranica smanjuje na R-L i upravo to spremamo u $val = \max(R - L, pY[j] - pY[i])$ jer je $y_i - y$ svakako manji od $R - L$ jer i-ta točka ne leži na gornjoj stranici.
I jedini slučaj koji smo preskočili, ako se prijašnji najveći kvadrat naslanjao na gornju stranicu kvadrata $[0, 1]^2$, tada ako točka (x_i, y_i) ograničava, u $val = \max(R - L, pY[j] - pY[i])$ se svakako sprema ispravna vrijednost. Ako nas i-ta točka neće ograničiti onda se ne dogada ništa jer je onda ili $R - L$ ili $y_i - y$ dovoljno velik da bude veći od M pa $M = \min(val, M)$ ne mijenja vrijednost od M.

Sada opravdavamo liniju $M = \min(val, M)$ koja odlučuje o promjeni M-a u i-tom koraku.
Ako je val veći od M linija $M = \min(val, M)$ vraća M i to je u redu jer se M ne može povećati u koraku. Ako je val manji od M onda $M = \min(val, M)$ mijenja vrijednost od M na val i to je "dobar update" vrijednosti od M. To je dobar update od M jer ako postoji kvadrat veće stranice od val on udara u točku (x_i, y_i). S obzirom da je val manji od M, znači da postoji kvadrat stranice val koji ne udara u prvih $i - 1$ točaka jer postoji i kvadrat stranice M, a $M > val$. S obzirom da želimo maksimizirat M u svakom koraku, upravo treba M postavit na val što i radi linija.
Pokazali smo da ako je u M na početku i-tog koraka spremljeno rješenje za $i - 1$ točaka, dodavanjem i-te točke znamo pravilno promijeniti M. Sada po principu matematičke indukcije kada obradimo sve točke poviše (x, y) dolazimo do rješenja koje je spremljeno u varijabli M.
Složenost algoritma je $\mathcal{O}(n^2)$, a algoritma za provođenje M simulacija $\mathcal{O}(n^2 m)$.

3 Gumbelova distribucija

S obzirom da je glavni dio našeg zadatka usporediti histogram vrijednosti $nM_m^2 - \log n - \log\log n$ s funckijom gustoće tzv. Gumbelove razdiobe, odlučili smo saznati nešto više o Gumbelovoj razdiobi kako bismo odmah na početku pokušali shvatiti hoće li i zašto uopće imati smisla uspoređivati histogram sa spomenutom gustoćom. Gumbelova distribucija dobila je ime po njemačkom matematičaru caru Emil Julius Gumbelu, na temelju njegovih originalnih radova i knjige u kojoj opisuje distribuciju ekstremnih vrijednosti. Gumbelova distribucija se koristi prilikom modeliranja maksimuma (odnosno minimuma) uzoraka raznih distribucija, pa je danas najpoznatija primjena Gumbelove distribucije u modeliranju metereoloških fenomena (npr. maksimalna godišnja razina rijeke) te u predviđanju ekstemnih potresa i drugih prirodnih katastrofa. Gumbelova funkcija distribucije je:

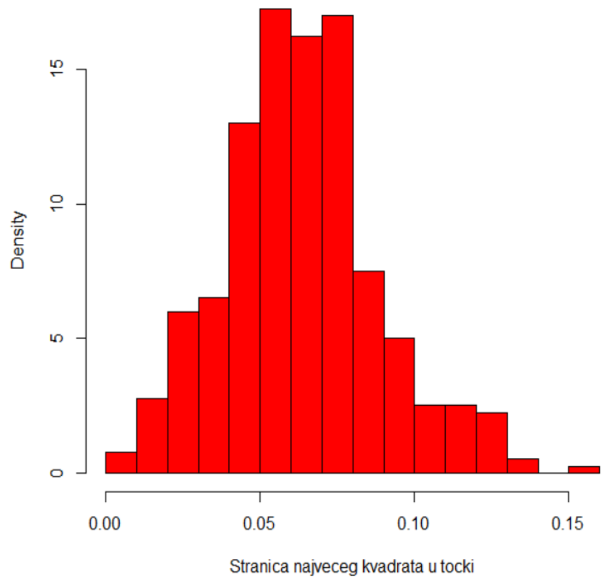
$$F(x;\mu,\beta) = e^{-e^{-\frac{x-\mu}{\beta}}} \qquad \mu \in \mathbb{R},\ \beta \in \mathbb{R}^+$$

U ovome zadatku ćemo se baviti standardnom Gumbelovom distribucijom čija je funkcija gustoće $e^{-(x+e^{-x})}$, a funkcija distribucije $e^{-e^{-x}}$ Dakle iz svega navedenog vidimo da možda i ima smisla uspoređivati zadani histogram s Gumbelovom distribucijom s obzirom da je naš histogram upravo histogram maksimalnih vrijednosti.

4 Histogram vrijednosti R_U za $n = 400$

Simulirali smo 400 uniformno distribuiranih y koordinata na $[0,1]$ te ih sortirali uzlazno. Tada smo simulirali 400 uniformno distribuiranih x koordinata na $[0,1]$ i tako dobili 400 simuliranih točaka na $[0,1]^2$.

Sada provodimo ranije opisani algoritam za svaku od 400 točaka i vrijednosti R_{U_i} spremamo u vektor rez na i-to mjesto. Na sljedećoj slici je prikazan histogram tako dobivenih vrijednosti R_{U_i} za $i = 1, \ldots, 400$:

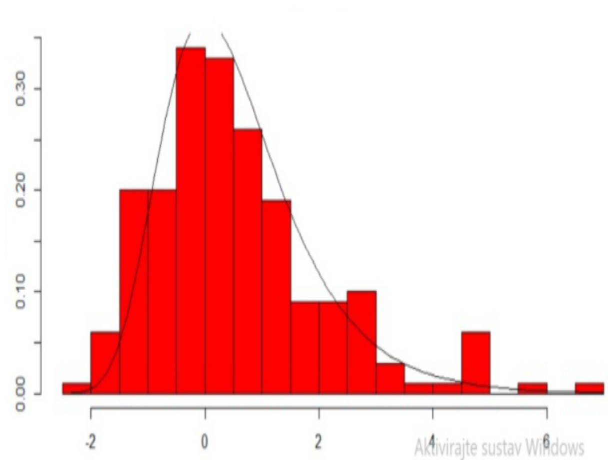


5 Ponavljanje simulacije s 400 točaka 200 puta

Ista simulacija je ponovljena $M = 200$ puta. Nakon i-te simulacije maksimalnu vrijednost iz vektora rez (M_m) spremamo u vektor $maksimum$. Tako dobiveni vektor se sastoji od elemenata M_m za $m = 1, \ldots, 200$. Sada taj vektor transformiramo tako da na svaki M_m djelujemo funkcijom

$$M_m \to nM_m^2 - \log n - \log \log n.$$

gdje je $n = 400$ broj točaka.
Na sljedećoj slici prikazan je histogram transformiranog vektora skupa s grafom funkcije gustoće Gumbelove distribucije:



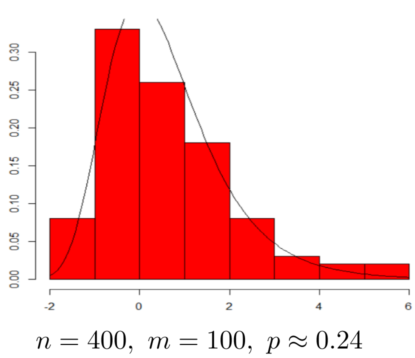
Vidimo da histogram poprilično dobro opisuje površinu ispod grafa funkcije gustoće, što je bilo i za očekivati.
Formalno ćemo naš uzorak usporediti s Gumbelovom distribucijom pomoću Kolgomorov-Smirnovljev testa sa sljedećim hipotezama:

H_0: Opaženi uzorak dolazi iz Gumbelove distribucije
H_1: Opaženi uzorak ne dolazi iz Gumbelove distribucije

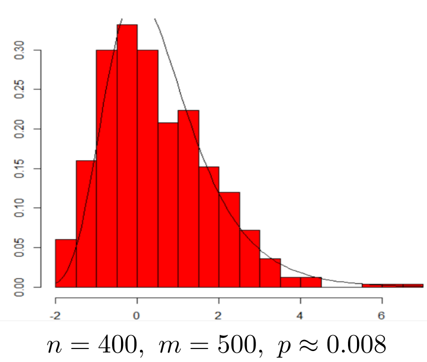
Fiksirat ćemo razinu značajnosti $\alpha = 0.05$ za ovaj i sve buduće testove. p-vrijednost provedenog testa je 0.06, što bi značilo da na razini značajnosti $\alpha = 0.05$ na osnovu uzorka ne možemo odbaciti nul-hipotezu o pripadnosti Gumbelovoj distribuciji.

6 Ponavljanje simulacije s n točaka M puta

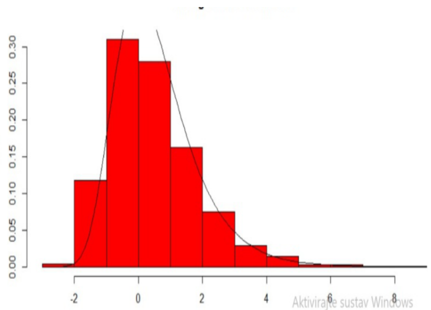
U ovom ćemo paragrafu napraviti isti postupak kao i u prijašnjem za razne vrijednosti od n i m. Priložit ćemo samo histograme i p-vrijednosti testa uz svaki uređeni par (n, m):



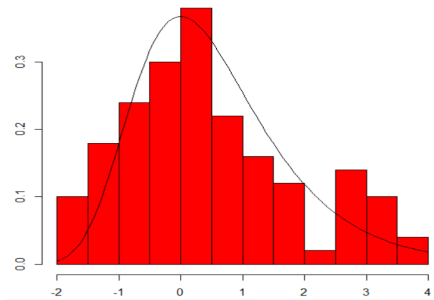
$n = 400,\ m = 100,\ p \approx 0.24$



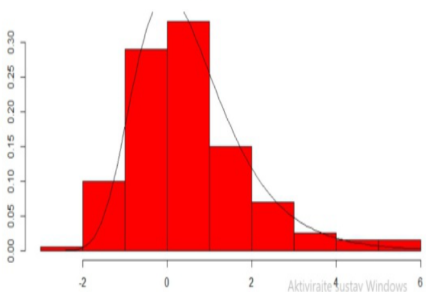
$n = 400,\ m = 500,\ p \approx 0.008$



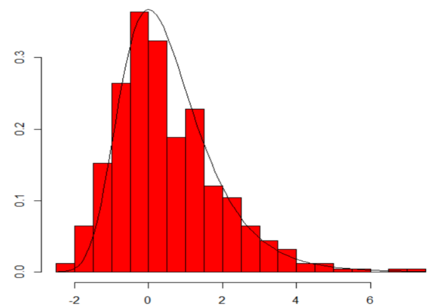
$n = 400,\ m = 1000,\ p \approx 2.4 \cdot 10^{-5}$



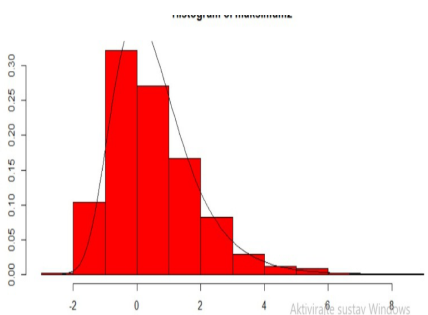
$n = 700,\ m = 100,\ p \approx 0.19$



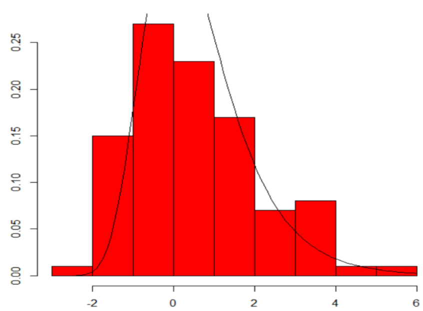
$n = 700,\ m = 200,\ p \approx 0.12$



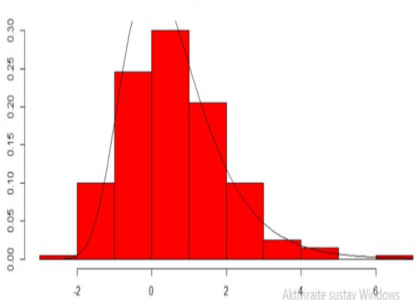
$n = 700,\ m = 500,\ p \approx 0.03$



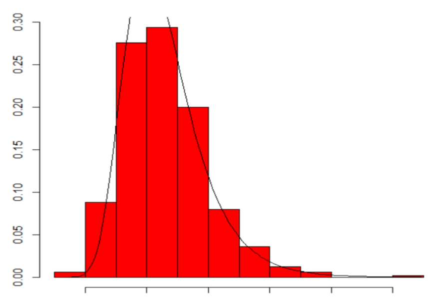
$n = 700,\ m = 1000,\ p \approx 0.08$



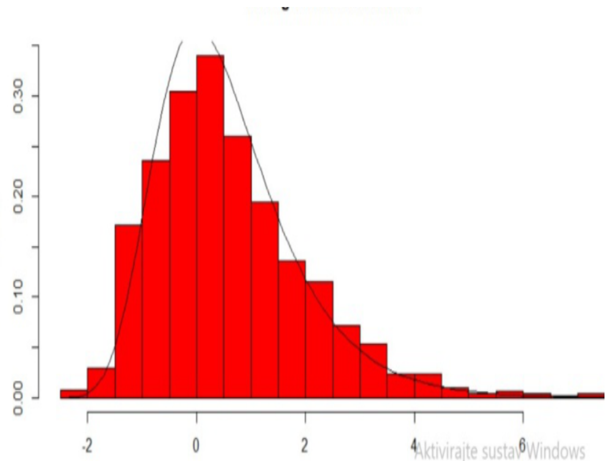
$n = 1000,\ m = 100,\ p \approx 0.27$



$n = 1000,\ m = 200,\ p \approx 0.45$



$n = 1000,\ m = 500,\ p \approx 0.30$



$n = 1000, \ m = 1000, \ p \approx 0.19$

U 9 od 12 na razini značajnosti $\alpha = 0.05$ ne odbacujemo nul-hipotezu da podaci dolaze iz standardne Gumbelove razdiobe.

7 Zaključak

Nakon što smo proveli sve simulacije i nacrtali histograme maksimalnih vrijednosti zajedno s funkcijom gustoće Gumbelove distribucije vidimo da ista jako dobro aproksimirira zadane podatke za sve odabrane n i m, što je dosta zanimljivo s obzirom da za fiksni k vrijednost izraza $nk^2 - \log n - \log \log n$ raste kako n raste, pa bi možda bilo za očekivati da će nam se povećanjem n-a histogram pomaknuti na desno. To se očito nije dogodilo. Razlog je taj što povećanjem broja točaka se stranica najvećeg kvadrata smanjuje (točke su zgusnutije), pa ta vrijednost dobro umiri povećanje n-a. Što se tiče provedenih KS testova u 9 od 12 slučajeva ne bismo odbacili odbacili hipotezu da podaci pripadaju standardnoj Gumbelovoj razdiobi na razini značajnosti $\alpha = 0.05$. Iako rezultate provedenih testova treba uzeti s dozom opreza zbog osjetljivosti KS testa na podatke i same činjenice da su podaci simulirani na slučajan način, to nam je još jedan dokaz koliko zapravo zadana distribucija dobro aproksimira podatke. Sve u svemu, na kraju možemo zaključiti da smo umjesto provođenja ovog „kompliciranog algoritma“ za pronalaženje stranica najvećih kvadrata mogli simulirati Gumbelovu distribuciju, te bismo dobili približno slične rezultate.