Statistički postupak za procjenu lokacije izvora onečišćenja

Mateljak, Domagoj

SVEUČILIŠTE U RIJECI
FAKULTET ZA FIZIKU

Domagoj Mateljak

STATISIČKI POSTUPAK ZA PROCJENU LOKACIJE IZVORA ONEČIŠĆENJA

Diplomski rad

Rijeka, 2023.

SVEUČILIŠTE U RIJECI
FAKULTET ZA FIZIKU

Diplomski studij Fizika i znanost o okolišu

Domagoj Mateljak

STATISIČKI POSTUPAK ZA PROCJENU LOKACIJE IZVORA ONEČIŠĆENJA

Diplomski rad

Mentor: doc. dr. sc. Darko Mekterović

Rijeka, 2023.

Sažetak

S porastom svijesti o utjecaju čestičnih tvari na zdravlje čovjeka, određivanje lokacije izvora onečišćenja dobiva sve veći značaj. Ova se metoda koristi već desetljećima, no nerijetko su u analizi podataka prisutne pogreške. Jedna od često korištenih metoda jest analiza vjetra pomoću uvjetne vjerojatnosne funkcije (engl. *conditional probability function* - CPF).

U ovom radu, ustanovljeno je da se u mnogim istraživanjima na temu određivanja kvalitete zraka putem CPF-a koriste premali skupovi podataka koji često navode na krive zaključke. Naime, te netočnosti trebaju biti izračunate, prikazane i analizirane. U sklopu ovog rada, ponuđene su dvije metode određivanja intervala pouzdanosti. Prva metoda koristi *bootstrap*, dok je druga metoda temeljena na određivanju intervala pouzdanosti omjera nezavisnih binomnih varijabli. Postupak je proveden za razinu pouzdanosti CL = 68,27 % za skupove od $N = 150$, 300, 1000 i 2500 podataka. Za svaku metodu proučava se broj podataka po smjeru vjetra te se za njih određuju ne samo intervali pouzdanosti, već i prekrivanje kao ocjena tih intervala. Metode su međusobno uspoređene te je pokazano da se rezultati oba postupka poklapaju. Zaključeno je da su oba postupka ispravna, ali i jednostavna za korištenje te da bi se trebali primjenjivati pri određivanju lokacije izvora onečišćenja.

Ključne riječi: Lokacija izvora onečišćenja, kvaliteta zraka, PM_{10}, polarni graf, uvjetna vjerojatnosna funkcija, CPF, *bootstrap*, omjer proporcija, omjer binomnih varijabli, intervali pouzdanosti

Sadržaj

1 Uvod

Ovisno o sastavu čestica u zraku, zrak drugačije utječe na okoliš i ljudsko zdravlje. Ako je prisutno onečišćenje, odnosno ako je prisutno odstupanje od normalnog sastava zraka, zbog prisutnosti neke strane tvari, potencijalno može doći do štetnog učinka na zdravlje organizama koji udišu taj zrak. Iz tog razloga, interesantno je promatrati, odnosno analizirati koncentraciju stranih tvari, odnosno onečišćivala, u zraku. U tu svrhu, postoje mjerne postaje za analizu kvalitete zraka. Na mjernim postajama, kada se mjere koncentracije onečišćivala, ako su zabilježeni podatci o smjeru vjetra, moguće je odrediti ne samo stupanj onečišćenja zraka, nego i odrediti izvor onečišćenja. Cilj mnogih istraživanja je određivanje izvora onečišćenja i doprinos ukupnom onečišćenju. U tom smislu, jedna od pomoćnih metoda koja se koristi je CPF (engl. *conditional probability function*). Naime, u stručnoj literaturi koja se koristi CPF-om kako bi se procijenila lokacija izvora onečišćenja, uopće se ne obraća pozornost na nepouzdanost dobivenih rezultata. Upravo zato je bitno razviti jednostavnu i lako primjenjivu metodu određivanja ocjene nepouzdanosti.

2 Podatci

Praćenje kvalitete zraka u Republici Hrvatskoj provodi se u okviru: državne mreže za trajno praćenje kvalitete zraka i lokalnih mreža za praćenje kvalitete zraka u županijama i gradovima koji uključuju i mjerne postaje posebne namjene [1]. Podatci za analizu kvalitete zraka u Republici Hrvatskoj javno su dostupni na mrežnoj stranici Ministarstva gospodarstva i održivog razvoja - Zavoda za zaštitu okoliša i prirode [2].

Na mrežnoj stranici „Kvaliteta zraka u Republici Hrvatskoj", čijim radom od 2010. godine upravlja Državni hidrometeorološki zavod - DHMZ, pod stručnim nadzorom Ministarstva gospodarstva i održivog razvoja, nalaze se podatci izmjerenih koncentracija onečišćujućih tvari u zraku [3]. Ovi podatci namijenjeni su potrebama Ministarstva gospodarstva i održivog razvoja, inspekcijskih službi, upravnih i javnih tijela, ali i potrebama stručne i znanstvene javnosti. Vlasnik postaje elektroničkim putem, preko mrežne stranice, prenosi izvorne podatke o koncentracijama onečišćujućih tvari u zraku, izmjerenih na satnoj bazi. Na mrežnoj se stranici također nalaze i validirani podatci o koncentracijama onečišćujućih tvari kao i godišnja izvješća o praćenju kvalitete zraka. Izmjerene koncentracije onečišćujućih tvari u zraku koje pristižu na

mrežne stranice u realnom vremenu, nisu službeni podatci ispitnih laboratorija, te se stoga ponekad naknadno izmijenjuju postupkom validacije. Ti se podatci na mrežnoj stranici prikazuju u obliku tablice, ali i grafički te ih je moguće preuzeti u obliku *excel* datoteke.

Za potrebe diplomske radnje, podatci su preuzeti s mrežne stranice Ministarstva gospodarstva i održivog razvoja Republike Hrvatske: http://iszz.azo.hr/iskzl/postajad.html?pid=155&mt=1# [4]. Korišteni podatci izmjereni su pomoću postaje ZAGREB-1. Postaja ZAGREB-1 nalazi se u blizini Koncertne dvorane Vatroslav Lisinski, aktivna je od 11. veljače 2003. godine te je namjenjena procijeni utjecaja na zdravlje ljudi i na okoliš. Postaja ZAGREB-1 mjeri meteorološke parametre temperature (℃), UV-B zračenja, brzine vjetra (m/s), smjera vjetra (°) i relativne vlažnosti (%) [4].

Podatci preuzeti i korišteni za diplomsku radnju su validirani satni podatci o brzini vjetra (m/s), smjeru vjetra (°) i koncentraciji čestica PM_{10} ($\mu g/m^3$), prikupljenih kroz period od 1. siječnja 2017. godine, do 1. siječnja 2022. godine [3]. Čestice PM_{10} (engl. PM - *particulate matter*) je naziv za mikroskopske čestične tvari efektivnog promjera manjeg od 10 μm.

3 Metodologija u literaturi

3.1 CPF - uvjetna vjerojatnosna funkcija

Lokacija izvora onečišćenja može se procijeniti primjenom CPF-a, odnosno uvjetne vjerojatnosne funkcije (engl. *conditional probability function*). CPF funkcija omogućava određivanje smjera iz kojeg dolazi onečišćenje, ali ne i specifičnog geografskog položaja izvora onečišćenja. CPF se definira kao uvjetna vjerojatnost da će izmjerena koncentracija biti iznad neke granične koncentracije, ako vjetar puše iz smjera $\Delta\theta$. CPF se ocjenjuje ocjenom u točki kao:

$$\mathrm{CPF}_{\Delta\theta} = \frac{m_{\Delta\theta}}{n_{\Delta\theta}}, \tag{3.1}$$

gdje je $n_{\Delta\theta}$ broj mjerenja unutar sektora $\Delta\theta$, a $m_{\Delta\theta}$ broj mjerenja unutar sektora $\Delta\theta$ iznad neke granične koncentracije. Izbor granične koncentracije najčešće je 75. percentil koncentracije svih mjerenja, te je iz tog razloga 75. percentil korišten unutar ove diplomske radnje.[5][6][7]. U tom slučaju $m_{\Delta\theta}$ predstavlja broj mjerenja unutar sektora $\Delta\theta$ s koncentracijom iznad 75.

percentila [5][6][7].

Budući da je CPF omjer broja mjerenja iznad neke granične koncentracije unutar sektora $\Delta\theta$ i ukupnog broja mjerenja unutar tog sektora, CPF je bezdimenzijska veličina. U mnogim istraživanjima također se često uklanjaju sva mjerenja gdje je zabilježena brzina vjetra manja od 1 m/s [5][8]. Naravno, u nekim istraživanjima ukljanjaju se i podatci s višim brzinama vjetra, kao npr. podatci s gdje je brzina vjetra $< 1,5$ m/s ili < 2 m/s [6][7]. Za potrebe ovog rada, uklanjaju se samo podatci s brzinom vjetra < 1 m/s.

Iako je korištenjem CPF-a moguće samo odrediti smjer iz kojeg onečišćenje dolazi, uz pomoć prethodnog znanja o izvorima promatranog onečišćenja, moguće je zaključiti koji su najvjerojatniji izvori zagađenja. Osim CPF-a, u mnogim se istraživanjima primjenjuju i drugi postupci određivanja lokacije onečišćenja, poput CBPF-a (engl. *conditional bivariate probability function*), koja je definirana na isti način kao i CPF, uz razliku da se podatci ne segmentiraju samo po smjeru vjetra, nego i po brzini vjetra. Za potrebe ove diplomske radnje, razmatrat će se samo rezultati CPF metode.

3.2 Problemi

Statistika je grana primijenjene matematike koja se bavi prikupljanjem, uređivanjem, analizom, sažimanjem, prezentiranjem i tumačenjem velikog broja podataka i donošenjem zaključaka o pojavama i procesima koje ti podatci predočuju. Podatci se dobivaju promatranjem, odnosno mjerenjem ili iz statističkoga pokusa [9]. Samo jedan podatak, odnosno samo jedno mjerenje nije dovoljno informativno, a ovisno o problemu ni dva, tri, deset ili više mjerenja nije dovoljno za ispravnu analizu. Stoga se provodi veliki broj mjerenja kako bi se dobila šira slika i donijeli valjani zaključci.

Međutim, jedan od problema u promatranju i provođenju mjerenja meteorološkog tipa jest to da ovisno o dobu dana, noći, tjednu, mjesecu ili čak i godini, moguće je izmjeriti vrijednosti koje daju značajno različite rezultate mjerenja. Zbog toga se istraživanja meteorološkog tipa provode na velikim vremenskim skalama te takva mjerenja često traju dugo.

Istraživanja u kojima se koristi CPF kako bi se odredila lokacija onečišćenja u tom pogledu nisu drugačija. Taj tip istraživanja provodi se za periode vremena u rasponu od nekoliko tjedana

do nekoliko godina, te se takva mjerenja najčešće izvode u satnim, trosatnim, šesterosatnim ili dnevnim intervalima. Takva istraživanja ne samo da se izvode kroz duge periode vremena, već su često skupa. Stoga, u mnogim istraživanjima te prirode, često nije prikupljen dovoljan broj podataka za ispravnu statističku inferenciju.

Bitno je naznačiti da na svakoj lokaciji mjerenja postoji nekakva raspodjela učestalosti puhanja vjetra u različitim smjerovima te je tipično da na nekoj lokaciji vjetar ne puše jednako učestalo iz svih smjerova. Štoviše, gotovo uvijek postoje smjerovi iz kojih vjetar slabije puše. Iz tog razloga, bez obzira koliko je velik broj prikuljenih podataka, pošto učestalosti puhanja vjetra nisu jednoliko raspodjeljene, vrlo vjerojatno je da će se pojaviti smjerovi vjetra iz kojih taj vjetar vrlo rijetko puše. Zbog takvih slučajeva je potrebno ocijeniti neodređenost rezultata. Ako ta neodređenost nije ocijenjena, tada može biti teško zaključiti ako je doprinos nekog smjera vjetra značajan ili ako je taj doprinos uzrokovan nasumičnim fluktuacijama.

Na primjer, u slučaju kada bi se samo osam mjerenja nalazilo u nekom od sektora vjetra, očekuje se da će se dva mjerenja naći iznad granične vrijednosti koncentracije te bi tada vrijednost CPF-a iznosila $\mathrm{CPF} = 0,25$. Ako bi umjesto toga samo jedno mjerenje bilo iznad granice, vrijednost CPF-a iznosila bi $\mathrm{CPF} = 0,125$, dok bi CPF iznosio $\mathrm{CPF} = 0,375$ u situaciji kada su tri mjerenja iznad granične vrijednosti. Razmatranjem ovih primjera vidljivo je koliko se potencijalni rezultati ovakvih istraživanja mogu razlikovati, pogotovo za mali broj mjerenja.

Pregledom literature na temu CPF-a i CBPF-a, primijećeno je da se ne obraća nikakva pozornost na procjenu neodređenosti rezultata. Prikupljanjem radova na temu CPF-a i određivanja lokacije izvora zagađenja, prikupljeno je 214 radova u periodu od 2004. do 2021. godine. Od ukupno 214 radova, informacija o korištenom broju podataka uopće nije navedena u 25 rada, dok se u 83 rada navodi da je korišteno 300 ili manje podataka. Osim toga, također je prikupljeno 117 radova u periodu od 2014. do 2021. godine na temu CBPF-a. Od ukupno 117 radova, informacija o korištenom broju podataka uopće nije navedena u 8 rada, dok se u 29 rada navodi da je korišteno 300 ili manje podataka. Pri korištenju CPF-a u svrhu određivanja smjera širenja onečišćenja, odnosno lokacije izvora onečišćenja, podatke se dijeli prema smjerovima vjetra ovisno o tome koliko smjerova vjetra je razmatrano. Podjela na premalen broj sektora smjera vjetra dovodi do zbunjujućih rezultata, pa se često koristi podjela na barem osam sektora, dok je najveća podjela na trideset šest sektora. Pri takvoj podjeli podataka, u svakom se sektoru nalazi još manji broj podataka. U primjeru s 300 mjerenja, pri podjeli na

osam sektora vjetra, očekivano je da se u svakom sektoru nađe 37 do 38 podataka ako vjetar puše s jednakom učestalosti iz svih smjerova. U realističnoj situaciji, to znači da će se u nekim sektorima naći osjetno veći, odnosno osjetno manji broj podataka. Ako se radi o CBPF-u, tada je u primjeru s 300 mjerenja, također potrebno izvršiti i segmentiranje ne samo po smjerovima vjetra, nego i po brzini vjetra, te je tada broj podataka u svakom sektoru još manji te je problem s malim brojem podataka još izraženiji.

4 Metode i teorija

Zbog pojave nepotpune i nezadovoljavajuće metodologije u mnogim istraživanjima koja koriste CPF, potrebno je sastaviti jednostavan primjer ispravnog postupka obrade i analize podataka u nadi da će se istraživanja tog tipa provoditi ispravnije u budućnosti. Kao što je ranije ustanovljeno, veliki problem istraživanja koja koriste CPF je nepostojanje procjene neodređenosti parametra uvjetne vjerojatnosti, što potencijalno dovodi do pogrešne interpretacije podataka. Iz tog razloga je potrebno ispravno analizirati podatke i procijeniti tu neodređenost uz pomoć intervala pouzdanosti. Međutim, konstrukcija tih intervala nije uvijek jednostavna. U sljedećim poglavljima ponuđena su dva jednostavna postupka konstrukcije intervala pouzdanosti. Prvi postupak određuje intervale primjenom *bootstrap* metode, dok se drugim postupkom računaju intervali pouzdanosti za omjer binomnih nezavisnih varijabli.

Naime, to još uvijek nije dovoljno. Potrebno je ocijeniti kvalitetu samih intervala pouzdanosti ovih dviju metoda konstrukcije intervala pouzdanosti. Intervali pouzdanosti su određeni s razinom pouzdanosti (*engl.* CL - *confidence level*). Za približnu konstrukciju intervala pouzdanosti, CL približno odgovara prekrivanju. Prekrivanje je veličina koja opisuje koliko bi intervala pouzdanosti, konstruiranih nekim postupkom konstrukcije, trebalo obuhvaćati pravu vrijednost promatranog parametra. Računanjem prekrivanja se provjerava ispravnost postupka konstrukcije intervala pouzdanosti. Ako se prekrivanje intervala pouzdanosti značajno razlikuje od razine pouzdanosti, tada to ukazuje na potencijalnu grešku u postupku konstrukcije intervala.

4.1 Intervali pouzdanosti

Nad izmjerenim podatcima, moguće je računati razne statistike poput očekivane vrijednost, varijance i slično te se njima dobivaju procjene promatranih parametara. Procjene parametara dobivene su primjenom estimatora, gdje je estimator statistika, odnosno funkcija za određivanje

vrijednosti nepoznatog parametra. Ako je parametar označen s ϑ, tada se estimator, odnosno procjena parametra ϑ označava s $\hat{\vartheta}$. Neka je s $\hat{\vartheta}$ označena statistika, odnosno estimator kojim se procjenjuje vrijednost parametra ϑ. Općenito vrijednost neke statistike $\hat{\vartheta}$, generalno gotovo nikad neće biti jednaka vrijednosti promatranog parametra ϑ. Najjednostavnije je promatrati procjenu u točki, ali tada je nepoznato koliko dobro ta procjena aproksimira pravu vrijednost parametra. Osim procjene u točki, moguće je razmotriti i raspon vrijednosti unutar kojeg se nalazi stvarna vrijednost parametra, tj. interval pouzdanosti [10]. Najbolja procjena parametra ϑ dobiva se promatranjem njene ocjene u točki, $\hat{\vartheta}$ zajedno s intervalima pouzdanosti, $[\hat{\vartheta_L}, \hat{\vartheta_U}]$. Pri određivanju intervala pouzdanosti, prvo je potrebno izabrati razinu pouzdanosti. Razina pouzdanosti postavlja se prije samog procesa određivanja statistike $\hat{\vartheta}$ i koristi se u provjeri ispravnosti postupka.

Intervali pouzdanosti mogu biti uski ili široki te je time određena njihova preciznost. Ako je razina pouzdanosti intervala visoka, a interval uzak, može se reći da je interval precizan, odnosno da je s visokom razinom pouzdanosti poznato da se vrijednost estimatora nalazi unutar uskog skupa mogućih vrijednosti. U slučaju širokih intervala, estimator se potencijalno nalazi unutar šireg skupa vrijednosti, te je time taj interval neprecizniji. Najneprecizniji interval, očito, ima granice od $-\infty$ do $+\infty$ [11].

Neka je dan skup podataka koji se raspodjeljuje po standardnoj normalnoj raspodjeli te neka je izabrana željena razina pouzdanosti, npr. CL= $68,27$ % (CL = *confidence level*). Interval s razinom pouzdanosti CL= $68,27$ % obuhvaća sve podatke koji se nalaze na udaljenosti jedne standardne devijacije σ od srednje vrijednosti skupa podataka, odnosno, razina pouzdanosti CL= $68,27$ % u teoriji obuhvaća $68,27$ % podataka iz tog skupa. Kada bi se više puta izveo eksperiment za koji su određeni intervali pouzdanosti s razinom pouzdanosti od CL= $68,27$ %, tada bi prava vrijednost parametra za koji su izračunati ti intervali pouzdanosti, bila sadržana unutar $68,27$ % tako konstruiranih intervala. Time je ujedno definirana i vjerojatnost prekrivanja, odnosno prekrivanje. Za potrebe ovog rada, intervali pouzdanosti promatrani su u sklopu frekvencionističke paradigme, gdje se vjerojatnost intepretira kao učestalost događaja tijekom višestrukog provođenja eksperimenta. U sklopu ove diplomske radnje, prekrivanje se određuje višestrukim provođenjem simulacija te takva ocjena ima neodređenost koja se iskazuje intervalom pouzdanosti prekrivanja [11].

Neka je nasumično uzet skup od n članova iz populacije za koju p označava udio članova

s nekim svojstvom X. Za potrebe ove radnje, neka je svojstvom X određena situacija kada konstruirani interval sadrži stvarnu vrijednost parametra. Odnosno, neka se članovi tog skupa raspodjeljuju prema Bernoullijevoj raspodjeli s parametrima n i p, te neka je p vjerojatnost da član posjeduje svojstvo X, dok je vjerojatnost da član ne posjeduje svojstvo X dana s $q = (1-p)$ [11].

Neka je estimator za p dan s $\hat{p} = \frac{m}{n}$, gdje je m količina članova tog skupa koji posjeduju svojstvo X. U slučaju kada je nasumično izabran mali skup od n mjerenja u usporedbi s veličinom originalne populacije iz koje su mjerenja preuzeta, tada je m smatran binomnom slučajnom varijablom s očekivanom vrijednosti $E(m) = np$ i standardnom pogreškom $\sigma_m = \sqrt{np(1-p)}$. U slučaju kada je n velik, odnosno kada je $np \geq 10$ i $nq \geq 10$, tada se raspodjela varijable m može aproksimirati normalnom raspodjelom. Očito je da se estimator $\hat{p}$, u slučaju kada je n velik, također raspodjeljuje po normalnoj raspodjeli [11].

Neka je pogreška od p dana s $\sigma_{\hat{p}} = \sqrt{p(1-p)/n}$ te neka se estimator $\hat{p}$ raspodjeljuje prema normalnoj raspodjeli. Frekvencionistička vjerojatnost da se prava vrijednost od p nađe unutar intervala pouzdanosti, odnosno vjerojatnost prekrivanja za postupak konstrukcije intervala pouzdanosti dana je s [11][12] :

$$P(-z_{\alpha/2} < \frac{\hat{p} - p}{\sqrt{p(1-p)/n}} < z_{\alpha/2}) \approx 1 - \alpha. \tag{4.1}$$

Izraz $z_{\alpha/2}$ naziva se z-vrijednost te ona daje procjenu koliko je standardnih devijacija σ neka točka udaljena od srednje vrijednosti nekog skupa podataka za normalnu raspodjelu. U izrazu 4.1, z vrijednost koristi se pri određivanju granica intervala pouzdanosti. Za veliki broj podataka n, mjerenja se raspodjeljuju prema normalnoj raspodjeli, te su za različite razine pouzdanosti z vrijednosti već dobro poznate. Na primjer, neka je CL definiran kao CL $= 1 - \alpha$. Tada za CL $= 68,27$ %, odnosno CL $= 0,6827$ vrijedi $\alpha = 0,3173$. U idealnom slučaju, za frekvencionistički postupak, α predstavlja proporciju puta koji interval ne bi trebao sadržavati pravu vrijednost promatranog parametra p. U slučaju za CL $= 0,6827$, vrijedi da je $z_{\alpha/2} = 1$.

Općenito, u mnogim radovima konstruiraju se intervali s razinom pouzdanosti od 68 %, $68,27$ %, 90 %, 95 %, $95,45$ % ili $99,73$ %. U tim slučajevima je potrebno prilagoditi $z_{\alpha/2}$

vrijednost. Unutar koda u dodatcima, dane su opcije korištenja razine pouzdanosti od $68,27$ %, $95,45$ % i $99,73$ %.

Ukratko, nakon određivanja razine pouzdanosti i pronalaska intervala pouzdanosti, potrebno je ocijeniti njihovu kvalitetu. Kvaliteta intervala pouzdanosti određuje se provjerom prekrivanja. U frekvencionističkoj paradigmi, kada bi se postupak konstrukcije intervala provodio više puta, prekrivanje bi trebalo približno odgovarati razini pouzdanosti.

4.2 *Bootstrap*

Bootstrap jest statistička metoda analize podataka koja se zasniva na višestrukom nasumičnom uzorkovanju iz jednog skupa mjerenja. Prilikom analize izmjerenih podataka, potrebno je ustanoviti raspodjelu mjerenja u svrhu određivanja raznih statistika, uključujući i intervale pouzdanosti. Raspodjela mjerenja, odnosno uzorka, ovisi o raspodjeli populacije, koja je u većini slučajeva nepoznata. Metoda *bootstrap-a* generalno se koristi kada je statistička raspodjela originalnih mjerenja nepoznata ili kada je količina mjerenja nedovoljna da se ustanovi raspodjela, odnosno, kada pretpostavke o normalnosti nisu zadovoljene [13].

Postupkom *bootstrap-a*, moguće je iz mjerenja, odnosno višestrukim uzorkovanjem iz jednog skupa mjerenja, jednostavno i efikasno odrediti aproksimativnu raspodjelu promatrane statistike. Tako dobivena raspodjela nije egzaktna, ali se jednostavno računa te je često vrlo dobra aproksimacija. Metode koje se koriste simulacijama, koriste računalne algoritme za generaciju pseudoslučajnih brojeva, stoga je moguće za isti skup podataka dobiti slične, ali različite konačne rezultate. Dok velik broj statističkih metoda koristi jedan nasumično uzorkovani skup iz populacije u svrhu dobivanja zaključka, *bootstrap* upotrebljava višestruko nasumično uzorakovanje iz originalnog skupa. Ti uzorci se uzorkuju na način da je više puta moguće izabrati isti podatak [13].

Bootstrap se također može shvatiti kao računalna metoda za određivanje mjere točnosti statističkih procjena. Jedan od najjednostavnijih primjera na kojima se *bootstrap* primjenjuje jest procjena točnosti estimatora srednje vrijednosti i standardne pogreške, ali se koristi i za procjenu točnosti mnogih drugih statistika. Iako je moguće *bootstrap-om* procjenjivati točnost raznih kompliciranijih statističkih parametara, taj proces često zahtijeva mnogo veću računalnu snagu, jer proces računanja unutar jedne iteracije *bootstrap* postupka postaje kompliciraniji s

porastom matematičke kompleksnosti naprednijih statistika te kada se uračuna činjenica da će se taj račun ponoviti veliki broj puta, jednom za svaku iteraciju *bootstrap-a*, svo dodatno vrijeme računanja se nakupi. Na sreću, u današnje vrijeme, to više nije toliki problem kao što je bio prije 30, 40 ili više godina. Najjednostavnija ocjena točnosti statistike jest standardna pogreška, ali *bootstrap-om* je moguće izračunati i druge mjere točnosti, poput pristranosti, intervala pouzdanosti itd. [13].

4.2.1 Određivanje intervala pouzdanosti putem *bootstrap-a*

Statistika se često bavi određivanjem svojstava iz nepoznate populacije na temelju nasumičnog uzorkovanja, odnosno estimacijom nekog svojstva vjerojatnosne raspodjele F iz nasumičnog uzorka te raspodjele. Nasumičan uzorak veličine n se definira kao skup n članova $x_1, x_2, ..., x_n$ nasumično izabranih iz konačnog skupa X sastavljenog od N pojedinačnih članova $X_1, X_2, ..., X_N$. Empirijska distribucijska funkcija $\hat{F}$ jest raspodjela podataka dobivenih nasumičnim uzorkovanjem iz populacije X, odnosno postupkom mjerenja, te je ona aproksimacija raspodjele F. Takva empirijska raspodjela dodjeljuje jednaku vjerojatnost pojavljivanja $1/n$ svakom članu $x_i, i = x_1, x_2, ..., x_n$. Jedna od velikih prednosti *bootstrap-a* jest ta da ne zahtijeva ikakvu spoznaju o raspodjeli $\hat{F}$ izmjerenih podataka [13].

Bootstrap uzorak definiran je kao nasumični uzorak veličine n izvučen iz $\hat{F}$ tako da je $\hat{F} \rightarrow (x_1^*, x_2^*, ..., x_n^*,)$, gdje $x^* = \{x_1^*, x_2^*, ..., x_n^*\}$ označava skup od n članova, dobiven nasumičnim uzorkovanjem iz populacije $x = \{x_1, x_2, ..., x_n\}$ na način da je moguće da neki član populacije bude izabran niti jednom ili više puta. Takav način nasumičnog uzorkovanja naziva se nasumičnim uzorkovanjem sa zamjenom [13].

Za potrebe *bootstrap* postupka, neka je dan skup $x = \{x_1, x_2, ..., x_N\}$ nasumičnih uzoraka iz potpuno nepoznate populacijske raspodjele F, te neka skup x predstavlja skup podataka izmjerenih iz neke populacije koja se raspodjeljuje prema F. Neka je parametar ϑ populacijske vjerojatnosne raspodjele F dan s $\vartheta = f(F)$, vrijednost statistike $\hat{\vartheta}$ tada je dana primjenom iste funkcije $f(x)$ na empirijsku distribuciju $\hat{F}$, te je tada $\hat{\vartheta} = f(\hat{F})$. Odnosno, ako $s(x)$ predstavlja funkciju promatrane statistike, tada se estimacija parametra $\vartheta = f(F)$ vrši estimacijom statistike $\hat{\vartheta} = s(x)$ na skupu podataka x. *Bootstrap* uzastopno višestruko vrši ovakvu procjenu statistike. Za svaki takav skup x^*, *bootstrap-om* se određuje estimacija za $\hat{\vartheta}$, odnosno, $\hat{\vartheta^*} = s(x^*)$ [13].

Za veliki broj ponavljanja *bootstrap-a*, može se pretpostaviti da se estimator $\hat{\vartheta}$ raspodjeljuje po normalnoj raspodjeli s nepoznatom očekivanom vrijednošću parametra ϑ. Neka je sada skup x^*, uzorkovan iz $\hat{F}$, $\hat{F} \to x^*$ te neka je za svaku iteraciju *bootstrap-a* određen $\hat{\vartheta}^* = s(x^*)$. Neka je $\hat{G}$ uvjetna distribucijska funkcija (CDF) statistika $\hat{\vartheta}^*$ dobivenih *bootstrap-om*. Tada je $1-\alpha$ percentilni interval definiran s $\alpha/2$ i $1-\alpha/2$ percentilima CDF-a $\hat{G}$, tako da je interval pouzdanosti dan s $[\hat{\vartheta}_L, \hat{\vartheta}_U] = [\hat{G}^{-1}(\alpha/2), \hat{G}^{-1}(1-\alpha/2)]$. Po definiciji, $\hat{G}^{-1}(\alpha/2) = \hat{\vartheta}^{*(\alpha/2)}$ te tada taj interval pouzdanosti iznosi $[\hat{\vartheta}_L, \hat{\vartheta}_U] = [\hat{\vartheta}^{*(\alpha/2)}, \hat{\vartheta}^{*(1-\alpha/2)}]$. R puta generira se skup x^*, odnosno, generira se R skupova x^{*1}, x^{*2}, ..., x^{*R} te se računa statistika $\hat{\vartheta}^*(b)$ za svaki skup x^{*b} tako da je $\hat{\vartheta}^*(b) = s(x^{*b})$, gdje je b = 1, 2, ..., R [13].

Nakon generacije statistike $\hat{\vartheta}^*$ za R iteracija *bootstrap-a*, potrebno ih je sortirati uzlaznim redoslijedom te se traže $\hat{\vartheta}^*(b)$ gdje je $b = R \cdot \alpha/2$ za donju granicu intervala pouzdanosti, dok je $b = R \cdot (1-\alpha/2)$ za gornju granicu intervala pouzdanosti. U slučaju da b nije cijeli broj, zaokružuje se na najbližu cjelobrojnu vrijednost. Neka je tada $\hat{\vartheta}_R^{*(\alpha/2)}$, $R \cdot \alpha/2$. precentil empirijske raspodjele, dok je $\hat{\vartheta}_R^{*(1-\alpha/2)}$, $R \cdot (1-\alpha/2)$. percentil empirijske raspodjele parametra $\hat{\vartheta}^*(b)$. Percentilni interval širine $1-\alpha$ je tada $[\hat{\vartheta_L}, \hat{\vartheta_U}] \approx [\hat{\vartheta_R^{*(\alpha/2)}}, \hat{\vartheta_R^{*(1-\alpha/2)}}]$ [13].

Korištnjem *bootstrap* metode, moguće je odrediti intervale pouzdanosti raznih statističkih parametara. Jednostavan postupak za dobivanje intervala pouzdanosti neke statistike ϑ, za skup podataka od N članova, dan je primjerom [14]:

1. Generacija M uzoraka, veličine N iz skupa mjerenja.
2. Za svaki generirani uzorak, potrebno je izračunati statistiku $\hat{\vartheta}$.
3. Statistike $\hat{\vartheta}_i$, gdje je $i = 1, ..., M$, potrebno je poredati po veličini, uzlazno.
4. Za razinu pouzdanosti CL = 68, 27 %, potrebno je pronaći srednjih 68, 27 % statistika $\hat{\vartheta}_i$. Granice tog intervala su tada ujedno donja i gornja granica intervala pouzdanosti za taj statistički parametar [14].

Na primjer, za razinu pouzdanosti CL = 68, 27 %, potrebno je pronaći srednjih 68, 27 % statistika $\hat{\vartheta}_i$ tako da se donja granica intervala nalazi na poziciji $0,15865 \cdot (M+1)$, a gornja granica intervala na poziciji $0,84134 \cdot (M+1)$, odnosno na poziciji 15, 865 %. i 84, 134 %. percentila. U slučaju da se na pozicijama na kojoj se ti percentili nalaze nisu cjeloborjne vrijednosti, potrebno ih je zaokružiti. Ovaj postupak računanja intervala pouzdanosti jest vrlo jednostavan te ne zahtijeva puno vremena za izračun nakon ispravne implementacije [13].

Iako je dan jednostavan postupak konstrukcije intervala pouzdanosti, javlja se problem nepoznate preciznosti postupka konstrukcije intervala. Iz tog razloga potrebno je izračunati prekrivanje za takav postupak. U frekvencionističkoj interpretaciji, prekrivanje jest proporcija intervala pouzdanosti koji bi obuhvaćali pravu vrijednost parametra ϑ, kada bi taj postupak bio izvršen više puta. U ovom radu, provjerava se postupak konstrukcije intervala pouzdanosti na način da se postupak konstrukcije intervala izvršava jednak broj puta kao i broj iteracija *bootstrap* petlje. Drugim riječima, Iz raspodjele F, koja predstavlja pravu raspodjelu vrijednosti podataka, nasumično se uzorkuje skup podataka te je time simuliran proces mjerenja podataka za potrebe nekog eksperimenta. Zatim se na skupu koji predstavlja izmjerene vrijednosti vrši postupak *bootstrap-a* za parametar ϑ te se određuje jedan interval pouzdanosti za taj parametar koji ili sadrži ili ne sadrži pravu vrijednost parametra ϑ. Taj postupak se provodi mnogo puta te se time određuje proporcija puta koju intervali pouzdanosti sadrže pravu vrijednost parametra ϑ [13].

U teoriji, prekrivanje bi trebalo odgovarati razini pouzdanosti. Ako je izračunato prekrivanje veće od razine pouzdanosti, tada je postupak konstrukcije intervala pouzdanosti konzervativan, dok je u suprotnom slučaju liberalan. Interval pouzdanosti prekrivanja se određuje prema izrazu:

$$\hat{p} \pm z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}. \tag{4.2}$$

Za CL $= 0,6827$ su tada $z^{(\alpha/2)} = z^{(0,15865)}$ = -1 i $z^{(1-\alpha/2)} = z^{(0,84135)} = 1$.

4.3 *Bootstrap* program

Kao što je ranije ustanovljeno, u mnogim istraživanjima na temu procjene lokacije onečišćenja putem CPF-a, prisutan je manjak jasnoće postupka obrade podataka, što često dovodi do otežane provjere ispravnosti istraživanja, a time i do krive interpretacije konačnih rezultata. Iz tog razloga, postupak će u ovom radu biti detaljno raspisan.

Za određivanje intervala pouzdanosti putem *bootstrap* postupka, korišten je program u R-u. Primjer korištenog programa može se naći u dodatcima. Petogodišnji satni podatci s postaje

ZAGREB-1 učitani su u obliku *excel* datoteke u CSV formatu. Podatci moraju biti učitani tako da se u prvom stupcu nalaze podatci o brzini vjetra, u drugom stupcu podatci o smjeru vjetra, a u trećem stupcu podatci o koncentracijama onečišćenja česticama PM_{10}.

Tijekom mjerenja, ponekad dolazi do kvara pri bilježenju ili prijenosu podataka te neki izmjereni podatci tada nedostaju. Stoga, program uklanja redove gdje nedostaje neki od podataka. Također program uklanja podatke s brzinom vjetra manjom od 1 m/s, što je učestalo u mnogim istraživanjima ovog tipa.

U programu je moguće izabrati broj sektora grafičkog prikaza podataka, odnosno podjelu smjerova vjetra. Zadana je podjela na osam sektora. Također, moguće je namjestiti i graničnu koncentraciju, koja je zadana kao 75. percentil koncentracije svih mjerenja. Pomoću petlje, podatci su sortirani u različite sektore te je za svaki sektor izračunat CPF. Sektori su odabrani tako da je izračunata širina sektora θ prema izrazu:

$$\theta = \frac{360^\circ}{\text{broj sektora}}. \tag{4.3}$$

gdje su granice sektora dane s $\pm\theta/2$. Na primjer, ako vjetar puše u iz smjera juga, tada je taj sektor vjetra određen s $\theta = 180^\circ \pm 22,5^\circ$. Ovakvim odabirom sektora, izbjegava se pristranost koja nastaje u programima kao što je funkcija *percentileRose()*, koja dostupna u paketu *openair* [15][16].

Nakon toga, program pohranjuje vrijednosti CPF-a petogodišnjih podataka kao „prave" vrijednosti CPF-a. Skup petogodišnjih mjerenja sadrži veliku količinu podataka koja simulira pravu populaciju iz koje bi podatci bili izmjereni u pravom svijetu za potrebe nekog eksperimenta. Drugi dio programa nasumično uzima manji skup podataka iz skupa „pravih" mjerenja i simulira proces uzorkovanja podataka koji se odvija u stvarnom svijetu. Raspodjela podataka po sektorima, određivanje kvantila i računanje vrijednosti CPF-a za „mjerenja" odvijaju se analogno te se CPF pohranjuje kao zasebna varijabla. Zatim se iz „mjerenja" vrši višestruko nasumično uzorkovanje sa zamjenom, gdje se generiraju skupovi podataka jednake veličine kao i skup simuliranih „mjerenja". Ovi manji skupovi se nazivaju *bootstrap* uzorci. Za svaki *bootstrap* uzorak ponovno se računa kvantil, sortira se po sektorima, izračunava se vrijednosti CPF-a i ta

se vrijednost pohranjuje u kao i-ti član vektora vrijednosti CPF-a dobivenih *bootstrap-om*, gdje je $i = 1, 2, ...R$.

Nakon što se izvede R iteracija *bootstrap* petlje i nakon što je generiran skup od R vrijednosti CPF-a za *bootstrap* uzorke, dobivene vrijednosti CPF-a sortirane su od najmanjih prema najvećim. Tako dobiveni vektor vrijednosti CPF-a, koristi se za određivanje granica srednjih CL $= 1 - \alpha$ vrijednosti *bootstrap* CPF-ova, odnosno za određivanje donje i gornje granice intervala pouzdanosti. Donja granica dobiva se množenjem $(R + 1)$ s $\alpha/2$, gdje je $\alpha = 1-$ CL. Taj umnožak daje redni broj *bootstrap* vrijednosti CPF-a koja predstavlja donju granicu intervala pouzdanosti. Gornja granica intervala pouzdanosti dana je s $(R + 1) \cdot (1 - \alpha/2)$. vrijednosti *bootstrap* CPF-a. Ako dobiveni redni broj nije cijeli broj, tada se zaokružuje na najbližu cjelobrojnu vrijednost. Vrijednost α se također može odabrati na početku programa, odnosno može se odabrati udaljenost od srednje vrijednosti normalne raspodjele σ. Ponuđene opcije jesu $\sigma =$ 1, 2 i 3, što odgovara CL $= 0,6827$, $0,9545$ i $0,9973$ respektivno.

Program provodi postupak za generaciju intervala pouzdanosti putem *bootstrap-a* kao što je ranije obrazloženo. Program zatim generira graf CPF-a simuliranog mjerenja, te prikazuje donju i gornju granicu intervala pouzdanosti kao točke povezane linijama.

Program računa prekrivanje ponavljanjem cijelog postupka izgradnje intervala pouzdanosti R puta, odnosno, izvodi metodu „bootstrap"-a R puta. U svakoj iteraciji generira se novi nasumično odabran uzorak koji predstavlja izmjerene vrijednosti. Pogreška prekrivanja određena je prema izrazu 4.2.

4.4 Bayesov teorem za određivanje CPF-a

Uvjetna vjerojatnost jest vjerojatnost da se ostvari promatrani događaj A, pod uvjetom da je ostvaren događaj B te je taj izraz dan s:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \tag{4.4}$$

gdje je $P(A \cap B)$ vjerojatnost da su se ostvarili događaji A i B, dok je $P(B)$ vjerojatnost da se ostvario događaj B. Vjerojatnost da se dogodio događaj B ako se dogodio događaj A, dana je izrazom:

$$P(B|A) = \frac{P(B \cap A)}{P(A)}, \tag{4.5}$$

Kombiniranjem izraza 4.4 i 4.5, dobiven je izraz:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{4.6}$$

Izraz 4.6 je poznat kao *Bayesov Teorem* te se koristi pri računanju uvjetnih vjerojatnosti [17].

Prije primjene Bayesovog teorema za određivanje CPF-a, potrebno je definirati korištene veličine i parametre. Neka su:

- Q - udio mjerenja s očekivanom koncentracijom iznad granične vrijednosti (npr. ako je granična koncentracija određena s 75. kvantilom, onda je $Q = 0,25$)
- $n_{\Delta\theta}$ - broj mjerenja unutar sektora $\Delta\theta$
- $m_{\Delta\theta}$ - broj mjerenja unutar sektora $\Delta\theta$ s vrijednosti koncentracije iznad granične
- N - ukupan broj mjerenja
- QN - ukupan broj mjerenja s vrijednostima koncentracije iznad granične
- $(1 - Q)N$ - ukupan broj mjerenja s vrijednostima koncentracije ispod granične
- $w_{\Delta\theta}$ - uvjet da se mjerenje nalazi unutar sektora $\Delta\theta$

CPF se ocjenjuje ocjenom u točki prema izrazu 3.1, ali se također može protumačiti kao vjerojatnost da se izmjerena koncentracija nalazi unutar gornjih 25% svih izmjerenih koncentracija, kada se to mjerenje nalazi unutar sektora $\Delta\theta$. Ako je ta vjerojatnost $P(Q|w_{\Delta\theta})$, tada je, primjenom Bayesovog teorema, odnosno primjenom izraza 4.6, dan izraz:

$$P(Q|w_{\Delta\theta}) = \frac{P(w_{\Delta\theta}|Q)P(Q)}{P(w_{\Delta\theta})}. \tag{4.7}$$

Vjerojatnost pronalaska mjerenja unutar sektora $\Delta\theta$ je dana s:

$$P(w_{\Delta\theta}) = P(w_{\Delta\theta}|Q)P(Q) + P(w_{\Delta\theta}|1 - Q)P(1 - Q). \tag{4.8}$$

Uvrštavanjem izraza 4.8 u izraz 4.7, dobiva se izraz:

$$P(Q|w_{\Delta\theta}) = \frac{P(w_{\Delta\theta}|Q)P(Q)}{P(w_{\Delta\theta}|Q)P(Q) + P(w_{\Delta\theta}|1-Q)P(1-Q)}, \tag{4.9}$$

odnosno izraz:

$$P(Q|w_{\Delta\theta}) = \frac{1}{1 + \frac{P(w_{\Delta\theta}|1-Q)}{P(w_{\Delta\theta}|Q)}\frac{P(1-Q)}{P(Q)}}. \tag{4.10}$$

Neka je:

$$P(w_{\Delta\theta}|1-Q) = \frac{n_{\Delta\theta} - m_{\Delta\theta}}{(1-Q)N}, \tag{4.11}$$

$$P(w_{\Delta\theta}|Q) = \frac{m_{\Delta\theta}}{QN}. \tag{4.12}$$

Tada, uvrštavanjem izraza 4.11 i 4.12 u izraz 4.10, dobiva se izraz:

$$P(Q|w_{\Delta\theta}) = \frac{1}{1 + \frac{\frac{n_{\Delta\theta} - m_{\Delta\theta}}{(1-Q)N}}{\frac{m_{\Delta\theta}}{QN}}\frac{P(1-Q)}{P(Q)}}. \tag{4.13}$$

Vjerojatnost da se mjerenje nađe na koncentraciji iznad granične vrijednosti, dana je izrazom:

$$P(Q) = Q, \tag{4.14}$$

dok je vjerojatnost da se mjerenje pronađe na koncentraciji ispod granične vrijednosti dana izrazom:

$$P(1-Q) = 1 - Q. \tag{4.15}$$

Konačno, uvrštavanjem izraza 4.14 i 4.15 u izraz 4.13, dobiva se izraz:

$$P(Q|w_{\Delta\theta}) = \frac{m_{\Delta\theta}}{n_{\Delta\theta}}. \tag{4.16}$$

Usporedbom izraza 4.16 s izrazom 3.1, jasno je da se radi o istom izrazu, odnosno da se CPF može promatrati kao $P(Q|w_{\Delta\theta})$.

4.5 Intervali pouzdanosti za omjer binomnih varijabli

Neka su p_1 i p_2 parametri binomnih varijabli te neka su dani s:

$$p_1 = \frac{n_{\Delta\theta} - m_{\Delta\theta}}{(1-Q)N}, \tag{4.17}$$

$$p_2 = \frac{m_{\Delta\theta}}{QN}. \tag{4.18}$$

Neka je ϕ omjer parametara p_1 i p_2 dviju nezavisnih binomnih varijabli te neka je taj omjer dan izrazom:

$$\phi = \frac{p_1}{p_2}. \tag{4.19}$$

Omjer binomnih varijabli je već obrađen u brojnoj literaturi te je dobro istražen.
se tada ocjenjuje estimatorom:

$$\hat{\phi} = \frac{\hat{p_1}}{\hat{p_2}}, \tag{4.20}$$

odnosno:

$$\hat{\phi} = \frac{\dfrac{n_{\Delta\theta} - m_{\Delta\theta}}{(1-Q)N}}{\dfrac{m_{\Delta\theta}}{QN}}. \tag{4.21}$$

Logaritam omjera binomnih varijabli aproksimativno se raspodjeljuje po normalnoj raspodjeli te se njegovom eksponencijacijom mogu dobiti intervali pouzdanosti estimatora $\hat{\phi}$ [18][19]. Interval pouzdanosti logaritma omjera binomnih varijabli je tada dan izrazom:

$$[\log \hat{\phi}_L, \log \hat{\phi}_U] = \log \hat{\phi} \pm z_{\alpha/2} \sqrt{\frac{1}{n_{\Delta\theta} - m_{\Delta\theta}} + \frac{1}{m_{\Delta\theta}} - \frac{1}{(1-Q)N} - \frac{1}{QN}}, \tag{4.22}$$

dok je interval pouzdanosti estimatora $\hat{\phi}$ dan izrazom:

$$[\hat{\phi}_L, \hat{\phi}_U] = \hat{\phi} e^{\pm z_{\alpha/2} \sqrt{\frac{1}{n_{\Delta\theta} - m_{\Delta\theta}} + \frac{1}{m_{\Delta\theta}} - \frac{1}{(1-Q)N} - \frac{1}{QN}}}. \tag{4.23}$$

Uvrštavanjem izraza 4.21 u izraz 4.13, dobiva se:

$$CPF = P(Q|w_{\Delta\theta}) = \frac{1}{1 + \phi \dfrac{1-Q}{Q}}, \tag{4.24}$$

te je za taj izraz moguće dobiti intervale pouzdanosti, uvrštavanjem izraza 4.23 u izraz 4.24.

4.6 Program za dobivanje intervala omjera nezavisnih binomnih varijabli

Postupak programiranja sličan je onome korištenom u poglavlju o *bootstrap* programu. Podatci se učitavaju preko *excel-a* u CSV formatu te je potrebno pohraniti u *excel* datoteku tako da se u prvom stupcu nalaze podatci o brzini vjetra, u drugom stupcu podatci o smjeru vjetra, a u trećem stupcu podatci o koncentracijama onečišćenja česticama PM_{10}. Program zatim uklanja sve podatke u kojima nedostaje mjerenje u barem jednom od stupaca podataka, te se također uklanjaju sva mjerenja s brzinom vjetra manjom od 1 m/s.

U programu je zadana podjela na 8 sektora te je odabrana razina poudaznosti od jednog σ, odnosno CL = 68, 27 %. Za svojsto na kojem se temelji CPF, koristi se koncentracija zagađenja

koja odgovara 75. perecentilu koncentracije svih mjerenja. Podjela sektora primjenjuje se na isti način kao što je opisano u postupku za *bootstrap*. Odnosno, širina sektora koji predstavlja smjer puhanja vjetra je dana izrazom 4.3.

Program zatim nasumično uzima uzorak značajno manjeg broja podataka iz originalnog skupa mjerenja, što simulira postupak mjerenja iz asimptotske populacije koja predstavlja pravu raspodjelu podataka. Zatim program sortira podatke simuliranog mjerenja po sektorima, odnosno po smjerovima vjetra te za svaki sektor računa vrijednost CPF-a prema izrazu 4.24, gdje je supstitucija za ϕ dana izrazom 4.21. Također se računaju i intervali pouzdanosti CPF-a tako da se umjesto izraza 4.21 koriste izrazi 4.23, gdje se umjesto ϕ unose izrazi $\hat{\phi}_L$ i $\hat{\phi}_U$ respektivno. Zatim se za tako izračunate vrijednosti CPF-a i njihove intervale pouzdanosti crta grafički prikaz.

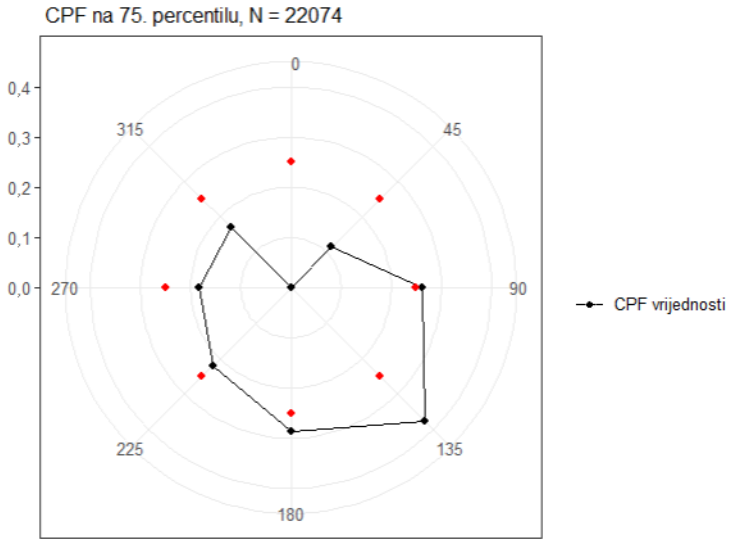
Za provjeru prekrivanja, ovaj se postupak ponavlja R puta. Odnosno, R puta se simulira proces mjerenja podataka te se računa proporcija puta koju intervali pouzdanosti sadrže pravu vrijednost CPF-a, odnosno vrijednost CPF-a petogodišnjeg skupa podataka. Konačno, računaju se intervali pouzdanosti za prekrivanje pomoću izraza 4.2.

5 Rezultati i diskusija

5.1 *Bootstrap*

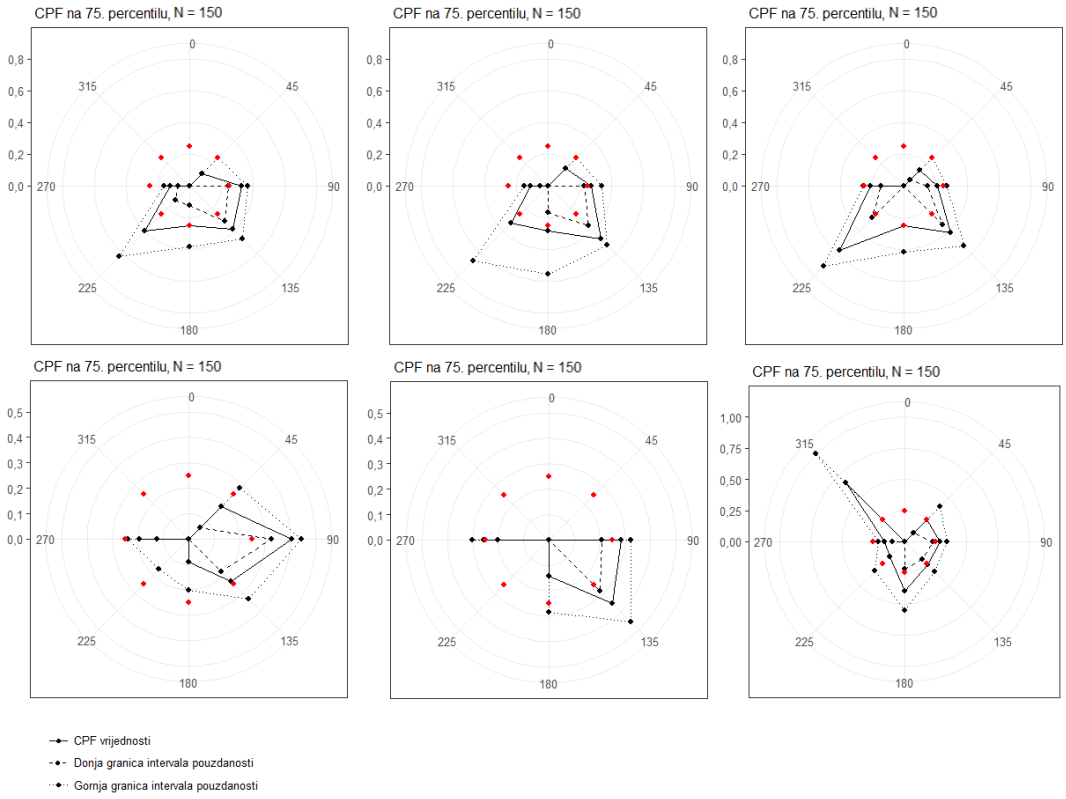
Za potrebe ovog istraživanja, analiziraju se validirani petogodišnji satni podatci o koncentraciji čestica PM_{10} prikupljenih na postaji ZAGREB-1. Skup podataka mjeren je u razdoblju 1. 1. 2017. do 1. 1. 2022. te sadrži 43824 podataka. Nakon uklanjanja svih stupaca podataka gdje nedostaje barem jedna izmjerena vrijednost te nakon uklanjanja svih podataka s brzinom vjetra manjom od 1 m/s, preostaje 22074 podataka. Sortiranjem tih podataka po smjeru vjetra i računanjem CPF-a na skupu od 22074 podataka, dobiven je grafički prikaz 5.1. Pošto je taj graf dobiven iz veoma velikog skupa podataka, on aproksimira pravu vrijednost CPF-a, odnosno CPF populacije iz koje se uzima uzorak. U ostatku ovog poglavlja, korišten je *bootstrap* u svrhu određivanja intervala pouzdanosti s razinom pouzdanosti od CL $= 68,27$ %. Intervali su generirani za nasumično uzorkovane skupove od 150, 300, 1000, 2500 mjerenja.

Grafički prikaz petogodišnjih satnih podataka CPF-a za 8 smjerova vjetra, prikazan je na slici 5.1. Crvene točke na grafu nacrtane su na poziciji gdje je CPF vjerojatnost, za dani sektor smjera vjetra, jednaka $CPF = 0,25$. Graf se tumači tako da ako CPF, za sektor $\Delta\theta$ iznosi $CPF > 0,25$, onda vjetar iz tog smjera povećava koncentracije lebdećih lestica, a ako je $CPF < 0,25$, onda vjetar iz tog smjera smanjuje koncentracije lebdećih čestica.



Slika 5.1: Grafički prikaz CPF-a petogodišnjih satnih podataka čestica PM_{10} iznad 75. percentila

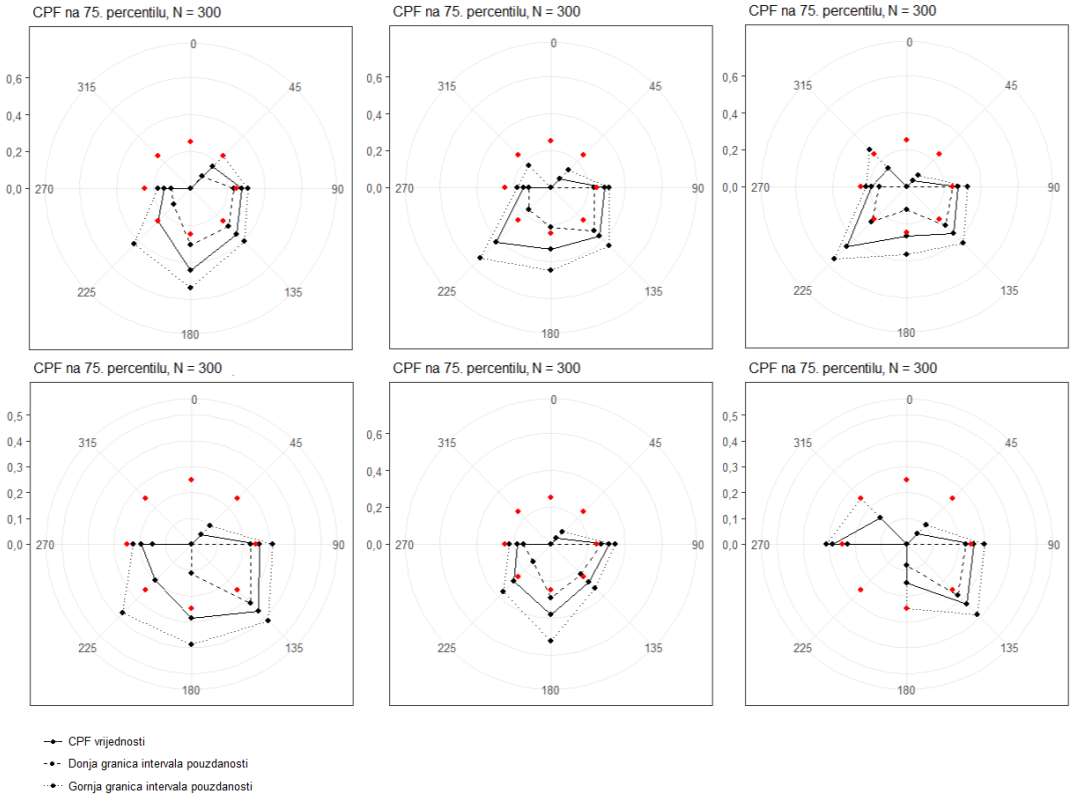
Grafovi na slikama 5.2, 5.1, 5.4 i 5.5 dobiveni su uzastopnim pokretanjem programa priloženog u dodatcima. Na svakoj od slika nalazi se 6 grafova CPF-a. Priložene tablice odgovaraju prvim grafovima CPF-a na navedenim slikama. U priloženim tablicama izračunate su vrijednosti CPF-a po smjeru vjetra, broj podataka u sektoru, broj podataka u sektoru iznad 75. percentila koncentracije svih podataka, intervali pouzdanosti za CPF, prekrivanje i pouzdanost prekrivanja.



Slika 5.2: 6 uzastopno dobivenih grafova CPF-a generiranih putem *bootstrap* postupka za $R = 1000$ ponavljanja i $N = 150$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

Tablica 1: Tablica vrijednosti CPF-a za *bootstrap* postupak s nasumičnim uzorkovanjem od 150 članova za $R = 1000$.

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	13	2	0,154	[0,000, 0,250]	0,640	[0,625, 0,655]
90	48	13	0,271	[0,225, 0,340]	0,723	[0,709, 0,737]
135	32	15	0,469	[0,357, 0,528]	0,726	[0,712, 0,740]
180	7	2	0,286	[0,167, 0,556]	0,732	[0,718, 0,746]
225	3	1	0,333	[0,000, 0,667]	0,655	[0,640, 0,670]
270	46	5	0,109	[0,050, 0,149]	0,714	[0,700, 0,728]
315	1	0	0,000	[0,000, 0,000]	0,329	[0,314, 0,344]
360	0	0	0,000	[0,000, 0,000]	0	[0,000, 0,000]

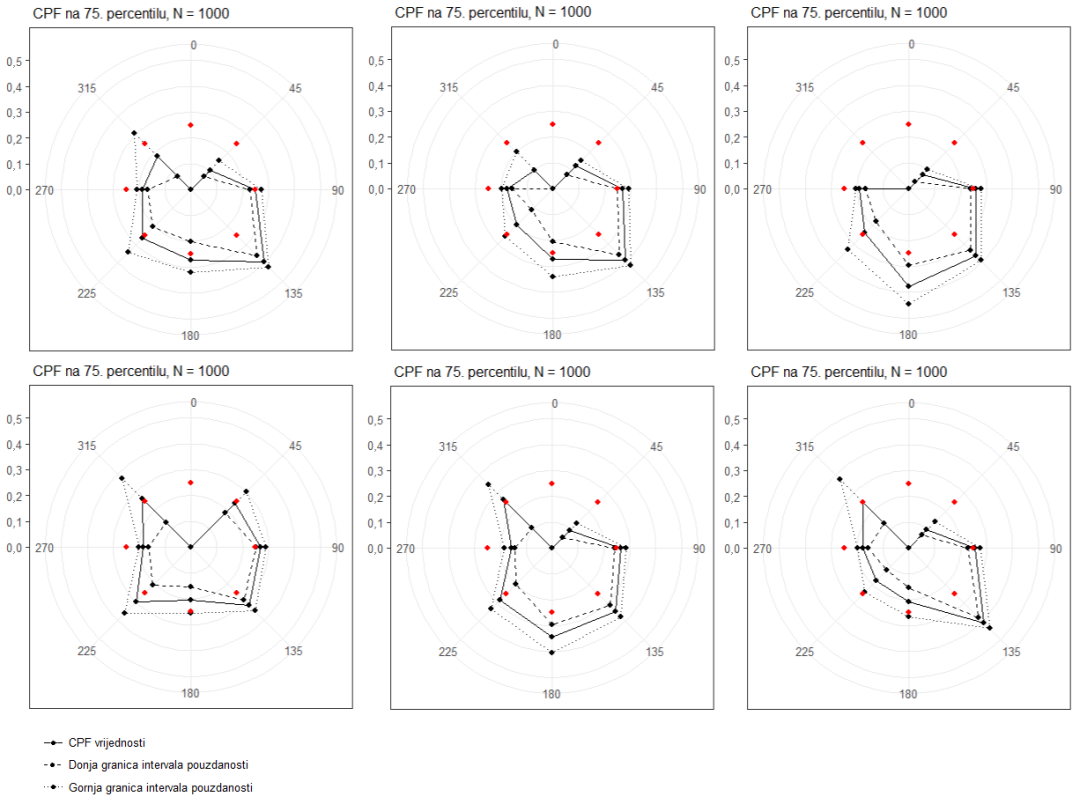


Slika 5.3: 6 uzastopno dobivenih grafova CPF-a generiranih putem *bootstrap* postupka za $R = 1000$ ponavljanja i $N = 300$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

Uspoređivanjem višestrukih grafičkih prikaza CPF-a dobivenih za $N = 150$ i $N = 300$ podataka, primjetno je da je višestrukim postupkom nasumičnog uzorkovanja iz originalnog skupa

Tablica 2: Tablica vrijednosti CPF-a za *bootstrap* postupak s nasumičnim uzorkovanjem od 300 članova za $R = 1000$.

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	24	4	0,167	[0,091, 0,250]	0,708	[0,694, 0,722]
90	91	25	0,275	[0,234, 0,311]	0,723	[0,709, 0,737]
135	65	23	0,354	[0,293, 0,408]	0,701	[0,687, 0,715]
180	18	8	0,444	[0,308, 0,538]	0,696	[0,681, 0,711]
225	8	2	0,250	[0,125, 0,429]	0,714	[0,700, 0,728]
270	92	13	0,141	[0,107, 0,176]	0,736	[0,722, 0,750]
315	2	0	0,000	[0,000, 0,000]	0,499	[0,483, 0,515]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]



Slika 5.4: 6 uzastopno dobivenih grafova CPF-a generiranih putem *bootstrap* postupka za $R = 1000$ ponavljanja i $N = 1000$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

podataka, moguće dobiti značajno različite vrijednosti CPF-a po smjerovima vjetra.

Tablica 3: Tablica vrijednosti CPF-a za *bootstrap* postupak s nasumičnim uzorkovanjem od 1000 članova za $R = 1000$.

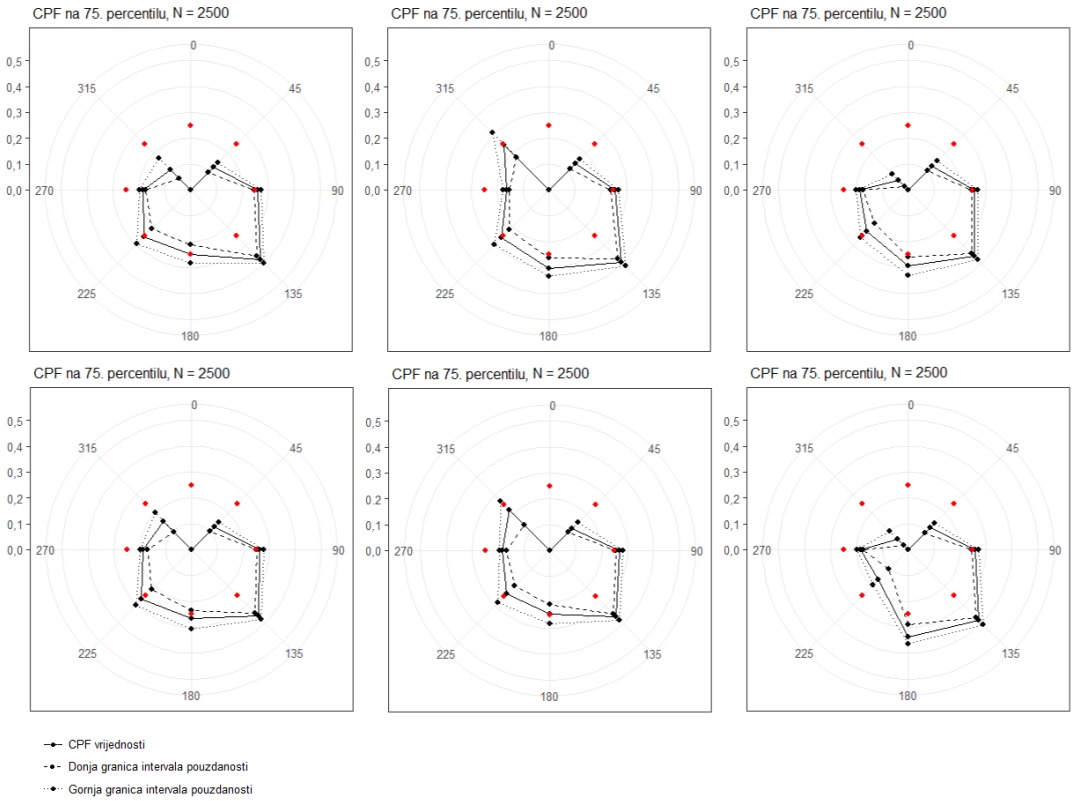
Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	66	7	0,106	[0,073, 0,156]	0,705	[0,691, 0,719]
90	310	77	0,248	[0,227, 0,271]	0,689	[0,674, 0,704]
135	196	78	0,398	[0,362, 0,425]	0,722	[0,708, 0,736]
180	55	15	0,273	[0,200, 0,321]	0,699	[0,684, 0,714]
225	45	12	0,267	[0,205, 0,340]	0,714	[0,700, 0,728]
270	317	59	0,186	[0,168, 0,206]	0,705	[0,691, 0,719]
315	11	2	0,182	[0,071, 0,308]	0,695	[0,680, 0,710]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]

U tablicama CPF vrijednosti za $N = 150$ i $N = 300$ podataka, za smjerove vjetra koji pušu iz smjera 315° i 360°, dobivene su CPF vrijednosti 0 s intervalima pouzdanosti [0,000 , 0,000]. Takvi intervali pouzdanosti nemaju smisla, ali su tako izračunati unutar programa jer su za svaku iteraciju *bootstrap*-a CPF vrijednosti iznosile 0. Takav izračun se javlja kada se u sektorima nalazi jako mali ili nikakav broj podataka. Iako takvi rezultati nemaju matematičkog smisla, uključeni su u tablicu radi potpunosti.

Ovisno o sreći izvlačenja, kada bi se interpretirali grafički prikazi poput onih na slikama 5.2 i 5.1, bez upotrebe intervala pouzdanosti, moglo bi se doći do pogrešnih zaključaka o smjeru značajnog doprinosa zagađenja. Međutim, uvažavanjem intervala pouzdanosti, moguće je umanjiti taj rizik.

Dobar primjer grafa koji bi naveo na pogrešan zaključak, bio bi 6. graf na slici 5.2. Za podatke vjetra koji puše iz smjera $\theta = 315°$, vrijednost CPF-a iznosi približno CPF = 0,75, ali ako se promotre intervali pouzdanosti i graf pravih vrijednosti 5.1, vidljivo je da to nije točno. U tom slučaju nemoguće je odrediti gdje se nalazi prava vrijednost CPF-a jer su intervali pouzdanosti dani na granicama od 0 do 1. Općenito, u slučajevima s niskim brojem podataka unutar sektora, jasno je da su intervali nepouzdani i često preširoki da se iz njih izvuku valjani zaključci.

Pomoću vizualne usporedbe grafova na slikama 5.4 i 5.5, za $N = 1000$ i $N = 2500$, vidljivo je da se, prilikom uzastopnog uzorkovanja, za sve sektore dobivaju relativno slične vrijednosti CPF-a, odnosno da su da je primjetna sve veća konzistentnost uzastopnog uzorkovanja. Unutar



Slika 5.5: 6 uzastopno dobivenih grafova CPF-a generiranih putem *bootstrap* postupka za $R = 1000$ ponavljanja i $N = 2500$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM10 čestica.

Tablica 4: Tablica vrijednosti CPF-a za *bootstrap* postupak s nasumičnim uzorkovanjem od 2500 članova za $R = 1000$.

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	168	21	0,125	[0,098, 0,149]	0,729	[0,715, 0,743]
90	748	194	0,259	[0,245, 0,272]	0,722	[0,708, 0,736]
135	533	202	0,379	[0,363, 0,400]	0,735	[0,721, 0,749]
180	140	35	0,250	[0,211, 0,284]	0,733	[0,719, 0,747]
225	105	27	0,257	[0,211, 0,294]	0,723	[0,709, 0,737]
270	770	142	0,184	[0,172, 0,196]	0,733	[0,719, 0,747]
315	36	4	0,111	[0,061, 0,172]	0,720	[0,706, 0,734]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]

sektora s manjim brojem podataka, je ta konzistentnost manja, ali također raste s brojem podataka.

Na slikama 5.4 i 5.5, primjetno je da su intervali na smjerovima s $\theta = 90°$, $\theta = 135°$ i $\theta = 270°$ vrlo uski s približno odgovarajućim prekrivanjem te da je na tim smjerovima pristuna najveća konzistentnost u postupku dobivanja CPF vrijednosti. Vrijednosti CPF-a na tim sektorima dovoljno dobro aproksimiraju CPF u asimptotskom slučaju. Osim toga, za $N = 1000$ i $N = 2500$, prekrivanje je za sve vrijednosti CPF-a prihvatljivo jer dobro aproksimira razinu pouzdanosti.

Pošto su su intervali za $N = 2500$, za smjerove $\theta = 90°$, $\theta = 135°$ i $\theta = 270°$ vrlo uski, postupak konstrukcije ispravan i vrijednosti CPF pouzdano aproksimiraju asimptotske vrijednosti, može se zaključiti da ukupni broj mjerenja po sektoru u većini slučajeva ne treba biti veći od $n_{\Delta\theta} \approx 533$ kako bi ih se moglo interpretirati bez uporabe intervala pouzdanosti. Intervali pouzdanosti predstavljaju potencijalni skup vrijednosti na kojima se prava vrijednost CPF-a nalazi, s razinom pouzdanosti CL $= 68,27$ %. Ako je vrijednost donje granice intervala pouzdanosti $> 0,25$, tada je moguće tvrditi da iz tog smjera dolazi veći doprinos zagađenju s CL $= 68,27$ %. Za određivanje donje granice broja mjerenja, odnosno najmanjeg broja mjerenja, nema jasnog pravila, ali je iz tablica vidljivo da ako je $n_{\Delta\theta} \approx 200$, tada su intervali relativno uski s približno odgovarajućim prekrivanjem. Dakle za radove s brojem podataka $n_{\Delta\theta} \approx 200$, manjak intervala pouzdanosti nije velika greška, ali bez navođenja broja podataka za svaki sektor, nemoguće je prosuditi jesu li rezultati zasigurno valjani. Intervali pouzdanosti bi generalno trebali biti određeni upravo kako bi se to moglo vidjeti, te bi se uz njih trebao navoditi i broj podataka u sektoru kako bi se olakšala njihova interpretacija.

Usporedbom tablica za $N = 150$, $N = 300$, $N = 1000$ i $N = 2500$, primjetno je da s porastom broja podtaka N također raste i vrijednost prekrivanja. S porastom broja podataka prema $N = 22074$, prekrivanje se približava broju 1. Naime, s većim brojem mjerenja, prekrivanje bi trebalo sve bolje odgovarati razini pouzdanosti. Bitno je naznačiti da se ovaj fenomen porasta prekrivanja javlja zato što s porastom N, veličina nasumično uzorkovanog uzorka više nije zanemariva u usporedbi s početnim brojem podataka. Iz tog razloga, dolazi do nezanemarivog preklapanja podataka u postupku nasumičnog uzorkovanja te se dobivaju intervali pouzdanosti koji sa sve većom učestalosti sadrže pravu vrijednost CPF-a. Naime, to ne znači da je priloženi postupak dobivanja intervala pouzdanosti loš, već je to ograničenje postupka dobivanja prekrivanja koji je u ovoj diplomskoj radnji korišten. Dakle intervali pouzdanosti dobiveni ovim postupkom su bolji nego što na prvi pogled čine, jer je dobivanje njihovog prekrivanja ograničeno s početnim brojem podataka iz kojeg se nasumično uzorkovalo.

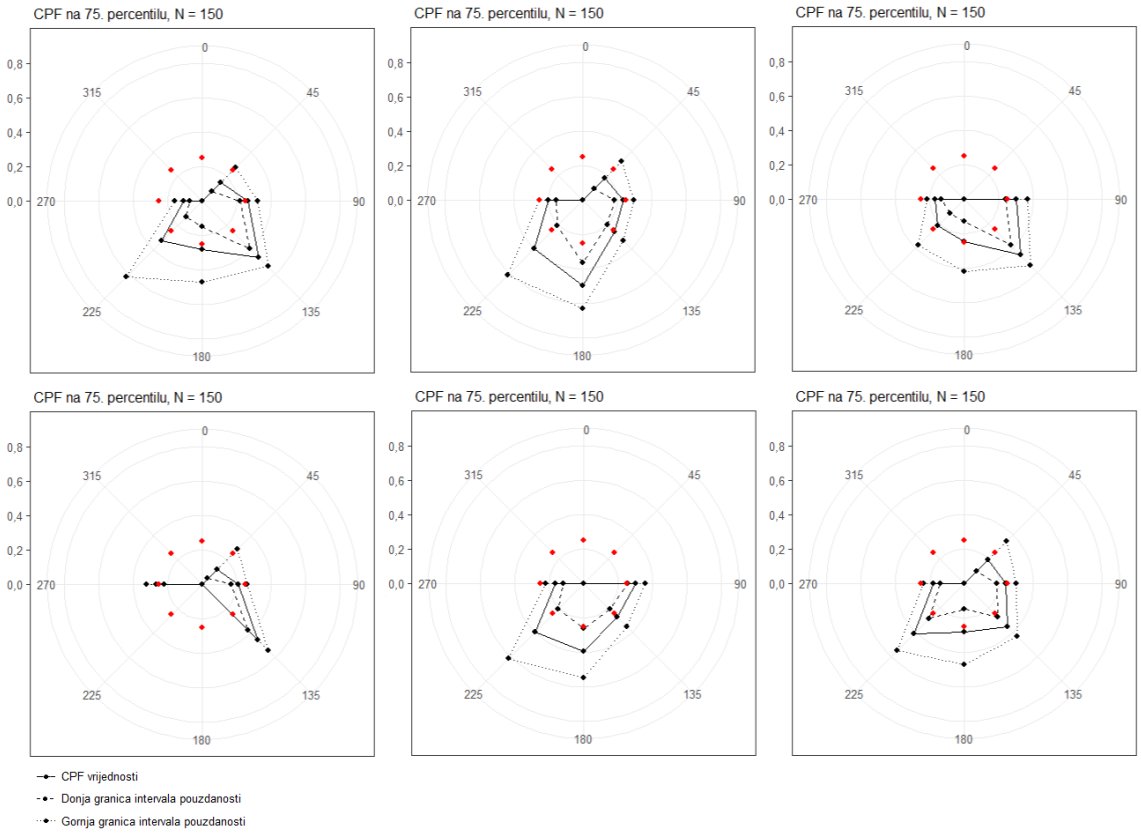
5.2 Omjer binomnih varijabli

Korišten je isti skup petogodišnjih satnih podataka kao i u prethodnom poglavlju te je za njega izračunat broj podataka iznad 75. percentila koncentracije. Ti su podatci zatim podijeljeni prema smjeru vjetra u 8 sektora. Sve vrijednosti CPF-a izračunate su s razinom pouzdanosti CL = 68, 27 %. CPF petogodišnjih satnih podataka čestica PM_{10}, izračunat je putem izraza 4.16 i 4.21. Dobiveni rezultati odgovaraju onima dobivenim u prethodnom poglavlju te grafički prikaz odgovara onom dobivenom na slici 5.1.

Ponovno, može se zaključiti da vjetrom koji puše iz smjera $\theta = 135°$ dolazi najveće povećanje koncentracije čestica PM_{10}. Skupovi od 6 uzastopno dobivenih grafova za nasumične uzorke veličine $N = 150$ i $N = 300$ iz petogodišnjih satnih podataka, koji predstavljaju podatke dobivene procesom mjerenja, nalaze se na slikama 5.6 i 5.7. Grafovi na tim slikama generalno daju preširoke intervale pouzdanosti za sektore s malim brojem podataka, što upućuje na to da ukoliko intervali pouzdanosti nisu navedeni, moguće je doći do pogrešnih zaključaka o lokaciji izvora onečišćenja.

Tablica 5: Tablica vrijednosti CPF-a za postupak putem omjera binomnih varijabli s nasumično generiranim skupom od 150 članova i intervalima pouzdanostima dobivenim putem izraza 4.2

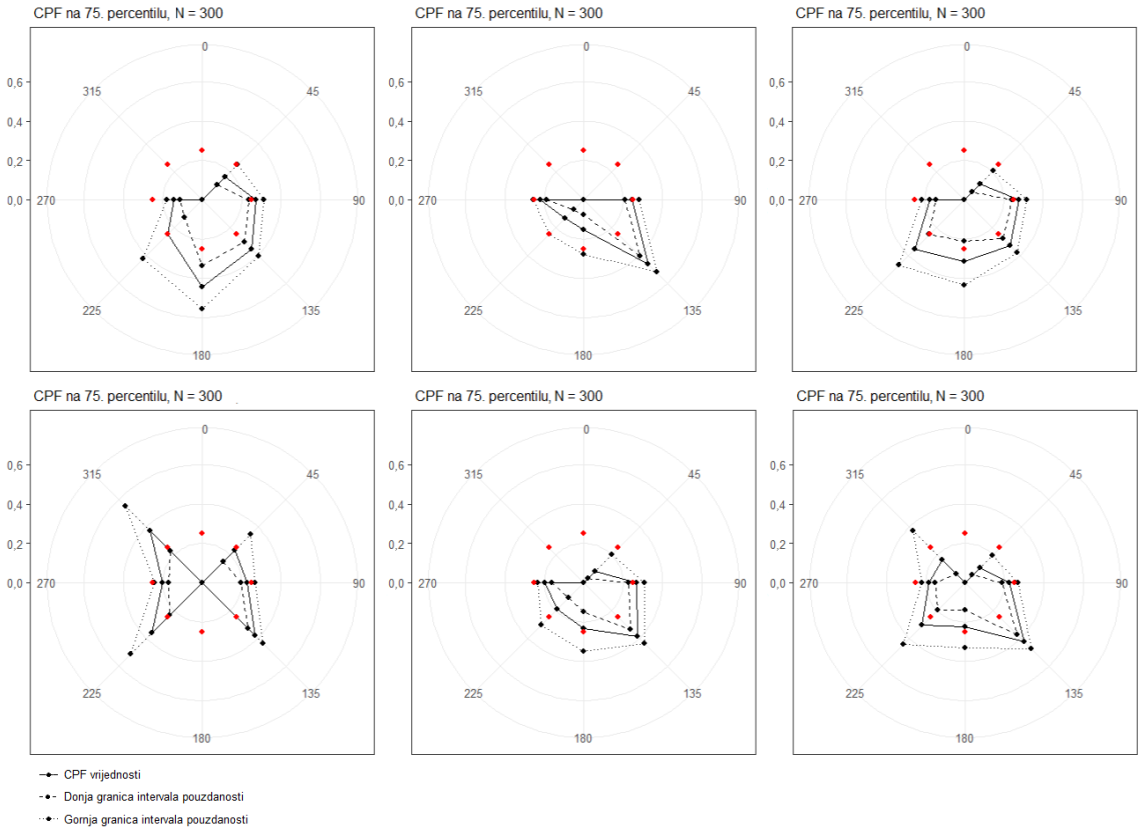
Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	13	2	0,152	[0,078, 0,273]	0,577	[0,561, 0,593]
90	48	13	0,267	[0,219, 0,322]	0,684	[0,669, 0,699]
135	32	15	0,464	[0,391, 0,539]	0,667	[0,652, 0,682]
180	7	2	0,282	[0,148, 0,470]	0,671	[0,656, 0,686]
225	3	1	0,329	[0,128, 0,622]	0,610	[0,595, 0,625]
270	46	5	0,107	[0,072, 0,156]	0,670	[0,655, 0,685]
315	1	0	0,000	[0,000, 0,000]	0,219	[0,206, 0,232]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]



Slika 5.6: 6 uzastopno dobivenih grafova CPF-a generiranih putem omjera binomnih varijabli za $N = 150$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije čestica PM_{10}.

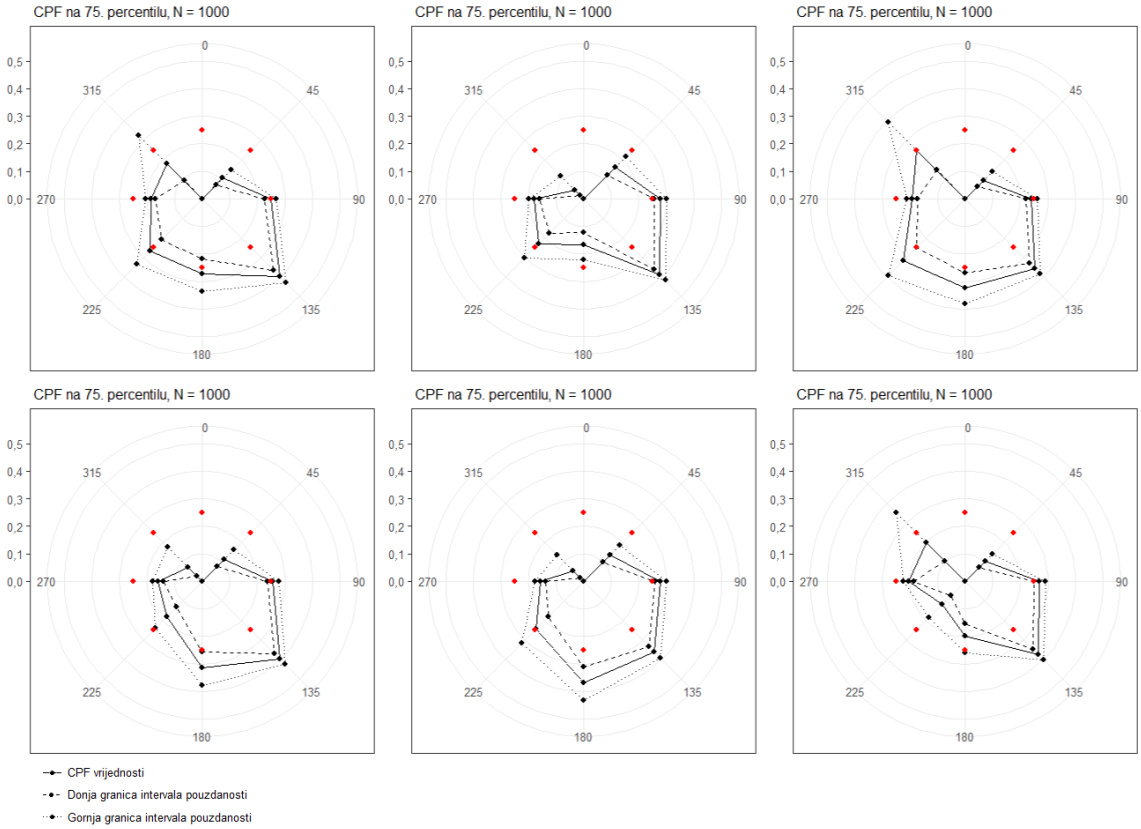
Tablica 6: Tablica vrijednosti CPF-a za postupak putem omjera binomnih varijabli s nasumično generiranim skupom od 300 članova i intervalima pouzdanostima dobivenim putem izraza 4.2

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	24	4	0,167	[0,105, 0,254]	0,720	[0,706, 0,734]
90	91	25	0,275	[0,238, 0,315]	0,707	[0,693, 0,721]
135	65	23	0,354	[0,305, 0,406]	0,685	[0,670, 0,700]
180	18	8	0,444	[0,337, 0,558]	0,707	[0,693, 0,721]
225	8	2	0,250	[0,130, 0,427]	0,708	[0,694, 0,722]
270	92	13	0,141	[0,112, 0,177]	0,711	[0,697, 0,725]
315	2	0	0,000	[0,000, 0,000]	0,415	[0,399, 0,431]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]



Slika 5.7: 6 uzastopno dobivenih grafova CPF-a generiranih putem omjera binomnih varijabli za $N = 300$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

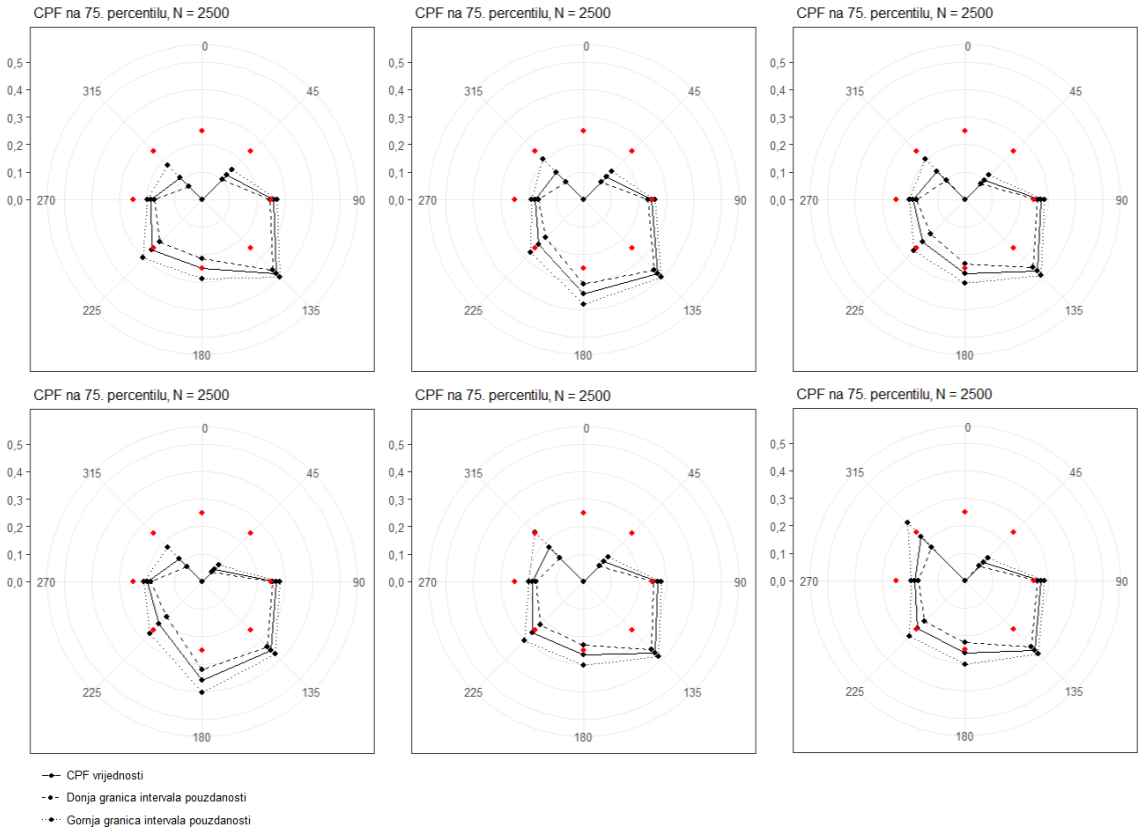
Na slikama 5.8 i 5.9 nalaze se nizovi od 6 uzastopno dobivenih grafova CPF-a za nasumične uzorke za koje je $N = 1000$ i $N = 2500$. Kao i u prethodnom poglavlju, prekrivanje približno odgovara razini pouzdanosti $\mathrm{CL} = 68,27\ \%$ što potvrđuje ispravnost postupka konstrukcije intervala pouzdanosti za sve smjerove vjetra. Osim same vrijednosti prekrivanja, važna je i širina intervala pouzdanosti. Širinu intervala moguće je smanjiti mijenjanjem razine pouzdanosti i povećanjem broja podataka. Naravno, zbog ograničenja korištenog postupka, za $N = 2500$, intervali pouzdanosti se ne polako ne suzuju i prekrivanje raste, ali to je samo ograničenje postupka testiranja i ne odnosi se na kvalitetu priloženog postupka konstrukcije intervala pouzdanosti.



Slika 5.8: 6 uzastopno dobivenih grafova CPF-a generiranih putem omjera binomnih varijabli za $N = 1000$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

Tablica 7: Tablica vrijednosti CPF-a za postupak putem omjera binomnih varijabli s nasumično generiranim skupom od 1000 članova i intervalima pouzdanostima dobivenim putem izraza 4.2

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	66	7	0,106	[0,074, 0,149]	0,715	[0,701, 0,729]
90	310	77	0,248	[0,229, 0,269]	0,694	[0,679, 0,709]
135	196	78	0,398	[0,368, 0,429]	0,663	[0,648, 0,678]
180	55	15	0,273	[0,218, 0,335]	0,707	[0,693, 0,721]
225	45	12	0,267	[0,207, 0,336]	0,699	[0,684, 0,714]
270	317	59	0,186	[0,168, 0,206]	0,694	[0,679, 0,709]
315	11	2	0,182	[0,093, 0,326]	0,705	[0,691, 0,719]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]



Slika 5.9: 6 uzastopno dobivenih grafova CPF-a generiranih putem omjera binomnih varijabli za $N = 2500$. Grafovi prikazuju vrijednosti CPF-a iznad 75. percentila koncentracije PM_{10} čestica.

Tablica 8: Tablica vrijednosti CPF-a za postupak putem omjera binomnih varijabli s nasumično generiranim skupom od 2500 članova i intervalima pouzdanostima dobivenim putem izraza 4.2

Smjer /°	$n_{\Delta\theta}$	$m_{\Delta\theta}$	CPF	Interval pouzdanosti	Prekrivanje	Interval prekrivanja
45	168	21	0,125	[0,102, 0,152]	0,718	[0,704, 0,732]
90	748	194	0,259	[0,246, 0,273]	0,692	[0,677, 0,707]
135	533	202	0,379	[0,361, 0,397]	0,704	[0,690, 0,718]
180	140	35	0,250	[0,216, 0,287]	0,694	[0,679, 0,709]
225	105	27	0,257	[0,218, 0,301]	0,723	[0,709, 0,737]
270	770	142	0,184	[0,173, 0,197]	0,713	[0,699, 0,727]
315	36	4	0,111	[0,069, 0,175]	0,723	[0,709, 0,737]
360	0	0	0,000	[0,000, 0,000]	0,000	[0,000, 0,000]

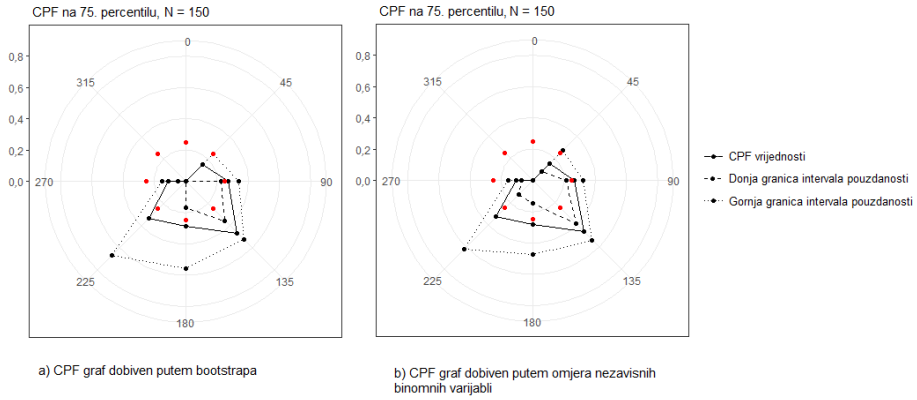
5.3 Usporedba metoda

U sklopu ovog rada, zbog nedostatka radova o CPF vjerojatnostima koji provjeravaju svoje rezultate ili koriste premali broj podataka za adekvatan zaključak, predstavljene su dvije različite metode sastavljanja intervala pouzdanosti CPF vjerojatnosti. Sljedeći je korak izvršavanje usporedbe predstavljenih metoda.

U asimptotskom režimu oba postupka daju iste vrijednosti CPF-a. Sada još preostaje usporediti vrijednosti CPF-a, intervale pouzdanosti te prekrivanje za manji broj podataka. Slika 5.10 i tablica 9 daju usporedbu CPF-ova dobivenih različitim metodama. CPF vrijednosti su za obje metode dobivene namještanjem sjemena u R programu s naredbom *set.seed(19)*, kako bi se osiguralo identično nasumično uzorkovanje pri postupku određivanja CPF-a. Nakon nasumičnog uzorkovanja su zatim primjenjeni različiti pripadni postupci za generaciju intervala pouzdanosti.

U tablicama 9 i 10, izostavljene su informacije o broju mjerenja i pouzdanosti prekrivanja zbog čitkosti tablice. Svrha ovih tablica je usporedba vrijednosti CPF-a, intervala pouzdanosti i prekrivanja. Izostavljene vrijednosti se mogu pronaći u tablicama 1, 5, 3 i 10.

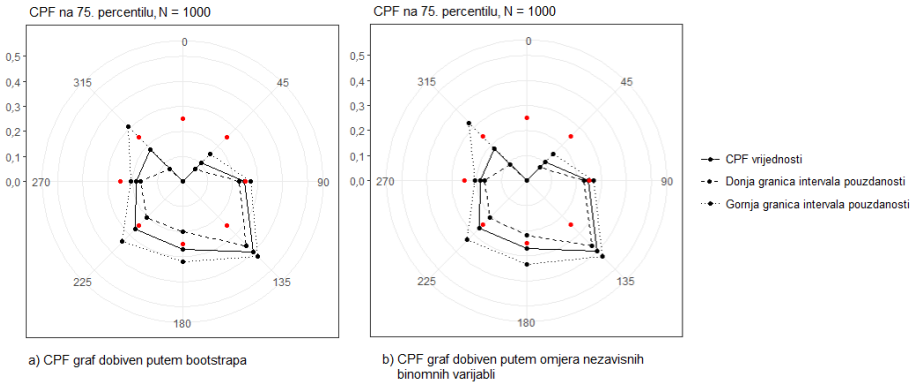
Usporedbom podataka u tablici 9, vrijednosti CPF-a su vrlo slične za obje metode kada je $N = 150$. Intervali pouzdanosti se bitno razlikuju za sektore s malim brojem podataka. Prekrivanje dobiveno *bootstrap* metodom je generalno konzervativno, dok je prekrivanje dobiveno za omjer binomnih varijabli liberalno, odnosno manje nominalne vrijednosti.



Slika 5.10: Grafovi CPF-a s $N = 150$ podataka za dvaju različitih postupaka

Tablica 9: Tablica vrijednosti CPF-a za dobivenih putem a) *bootstrap-a* i b) omjera nezavisnih binomnih varijabli za $N = 150$

	a)			b)		
Smjer /°	CPF	Interval pouzdanosti	Prekrivanje	CPF	Interval pouzdanosti	Prekrivanje
45	0,154	[0,000, 0,250]	0,640	0,152	[0,078, 0,273]	0,577
90	0,271	[0,225, 0,340]	0,723	0,267	[0,219, 0,322]	0,684
135	0,469	[0,357 0,528]	0,726	0,464	[0,391, 0,539]	0,667
180	0,286	[0,167, 0,556]	0,732	0,282	[0,148, 0,470]	0,671
225	0,333	[0,000, 0,667]	0,655	0,329	[0,128, 0,622]	0,610
270	0,109	[0,050, 0,149	0,714	0,107	[0,072, 0,156]	0,670
315	0,000	[0,000, 0,000]	0,329	0,000	[0,000, 0,000]	0,219
360	0,000	[0,000, 0,000]	0	0,000	[0,000, 0,000]	0,000



Slika 5.11: Grafovi CPF-a s $N = 1000$ podataka za dvaju različitih postupaka

Tablica 10: Tablica vrijednosti CPF-a za dobivenih putem a) *bootstrap-a* i b) omjera nezavisnih binomnih varijabli za $N = 1000$

	a)			b)		
Smjer /°	CPF	Interval pouzdanosti	Prekrivanje	CPF	Interval pouzdanosti	Prekrivanje
45	0,106	[0,073, 0,156]	0,705	0,106	[0,074, 0,149]	0,715
90	0,248	[0,227, 0,271]	0,689	0,248	[0,229, 0,269]	0,694
135	0,398	[0,362, 0,425]	0,722	0,398	[0,368, 0,429]	0,663
180	0,273	[0,200, 0,321]	0,699	0,273	[0,218, 0,335]	0,707
225	0,267	[0,205, 0,340]	0,714	0,267	[0,207, 0,336]	0,699
270	0,186	[0,168, 0,206]	0,705	0,186	[0,168, 0,206]	0,694
315	0,182	[0,071, 0,308]	0,695	0,182	[0,093, 0,326]	0,705
360	0,000	[0,000, 0,000]	0,000	0,000	[0,000, 0,000]	0,000

Usporedbom intervala pouzdanosti obiju metoda za $N = 1000$, vidljivo je da je prekrivanje ponekad konzervativnije, a ponekad liberalnije bez nekog pravila. Odnosno, s porastom broja podataka N, prekrivanja su sve sličnija jedna drugom te nije moguće utvrditi koja metoda daje bolje prekrivanje. Kao što je i očekivano, asimptotski rezultat sve manje ovisi o izabranoj metodi. To upućuje na zaključak da je su obje metode validne opcije za proučavanje CPF-a, osobito ako je korišten veći broj mjerenja. Također, postupak dobiven putem omjera binomnih varijabli bi u budućim istraživanjima mogao biti poželjniji zbog mnogo bržeg komputacijskog vremena.

Bitno je napomenuti, da je s postupkom u poglavljima 4.4 i 4.5, konstrukcija intervala pouzdanosti svedena na problem određivanja intervala pouzdanosti za omjer nezavisnih binomnih varijabli, što je dobro istražen problem u stručnoj literaturi [19]. Iako je u stručnoj literaturi ponuđen veći broj potencijalnih opcija, za potrebe diplomske radnje, izabrana je metoda s najlakšom implementacijom. Ostale preporučene metode su vjerojatno točnije, ali je njihova implentacija u program teža. Sljedeći korak koj bi bilo dobro napraviti je pokušaj implementacije ostalih ponuđenih metoda kako bi se pronašle još bolje metode konstrukcije intervala pouzdanosti putem omjera binomnih varijabli.

6 Zaključak

Prikupljeni su petogodišnji satni podatci o zagađenju zraka česticama PM_{10} te su za njih analizirane vrijednosti CPF-a, intervali pouzdanosti, prekrivanje i pouzdanost prekrivanja za razinu pouzdanosti CL $= 68,27\%$. Analizom podataka za $N = 150$ i $N = 300$ potvrđena je pretpostavka da se u mnogim radovima koristi nedovoljan broj podataka, te ako neodređenost CPF-a nije procijenjena, tada CPF metoda ne može biti korištena za donošenje zaključka o smjeru širenja onečišćenja.

Intervali pouzdanosti dobiveni su za dvije različite metode te je za njih također dobiveno prekrivanje. Prekrivanje dobiveno frekvencionističkim postupkom približno odgovara razini pouzdanosti, što pokazuje da je sam postupak konstrukcije intervala ispravan za obje metode. Također su analizirani postupci konstrukcije intervala za $N = 1000$ i $N = 2500$, te je ustanovljeno da su za te brojeve podataka generalno dobivene vrijednosti koje učestalo ispravno aproksimiraju asimptotske vrijednosti. Naime, nije dovoljno promatrati samo općeniti broj elemenata nego i broj elemenata unutar danog sektora smjera vjetra. Osim toga, uvijek bi se trebali razmatrati intervali pouzdanosti. Također, broj podataka može se drastično smanjiti nakon uklanjanja loše izmjerenih podataka i uklanjanja vrijednosti s premalom brzinom vjetra, pa bi iz tog razloga, prije pokretanja eksperimenta, trebalo odrediti broj podataka koji će biti promatrani, tako da taj broj podataka odgovara preporuci čak i nakon čišćenja. Unatoč tome, tijekom istraživanja, nije moguće unaprijed znati raspodjelu, odnosno broj podataka po sektoru, ali upravo to i jest razlog zbog kojeg bi se trebao provoditi postupak konstrukcije i analize intervala pouzdanosti. Potrebno je promatrati broj podataka zajedno s intervalima pouzdanosti unutar danog sektora kako bi bilo moguće precizno i s određenom razinom pouzdanosti tvrditi da je doprinos nekog smjera vjetra značajan ili neznačajan.

Svi rezultati dobiveni su za razinu pouzdanosti CL $= 68,27\%$ te bi preporučen broj podataka po sektoru smjera vjetra bio veći za više razine pouzdanosti. Konačno, uspoređena je brzina provođenja računalnog programa za izračun vrijednosti CPF-a i *bootstrap-a* te bi za veoma veliki broj podataka bila prikladnija metoda dobivanja intervala pouzdanosti putem omjera nezavisnih binomnih varijabli dok je prednost *bootstrap* metode to da je moguće optimizirati proces dobivanja intervala pouzdanosti ovisno o korištenoj metodi *bootstrap-a*.

Literatura

[1] Ministarstvo gospodarstva i održivog razvoja (Republika Hrvatska) - „Kvaliteta zraka", 2021. Dostupno na: https://mingor.gov.hr/o-ministarstvu-1065/djelokrug/uprava-za-klimatske-aktivnosti-1879/zrak/kvaliteta-zraka/1310. [Pristupljeno 16. 10. 2022.]

[2] Ministarstvo gospodarstva i održivog razvoja (Republika Hrvatska) - Zavod za zaštitu okoliša i prirode - „Informacijski sustav zaštite zraka", 2018. Dostupno na: https://www.haop.hr/hr/baze-i-portali/kvaliteta-zraka-u-republici-hrvatskoj. [Pristupljeno 16. 10. 2022.]

[3] Ministarstvo gospodarstva i održivog razvoja (Republika Hrvatska) - „Državna mreža za trajno praćenje kvalitete zraka", 2008. Dostupno na: http://iszz.azo.hr/iskzl/index.html. [Pristupljeno 16. 10. 2022.]

[4] Ministarstvo gospodarstva i održivog razvoja (Republika Hrvatska) - „Detaljni podatci o postaji ZAGREB-1", 2008. Dostupno na: http://iszz.azo.hr/iskzl/postajad.html?pid=155&mt=1#. [Pristupljeno 16. 10. 2022.]

[5] Y. Su, U. Sofowote, A. Munoz, M. Noble, C. Charron, A. Todd, V. Celo, E. Dabek-Zlotorzynska, A. Kryukova, T. Switzer, „Baseline Air Monitoring of Fine Particulate Matter and Trace Elements in Ontario's Far North, Canada", Kanada, „Applied Sciences", vol. 11, br. 13, str. 6140, lipanj 2021. Dostupno na: https://doi.org/10.3390/app11136140. [Pristupljeno 18. 11. 2022.]

[6] , M. Manousakas, E. Diapouli, H. Papaefthymiou, A. Migliori, A.G. Karydas, R. Padilla-Alvarez, M. Bogovac, R.B. Kaiser, M. Jaksic, I. Bogdanovic-Radovic, K. Eleftheriadis, „Source apportionment by PMF on elemental concentrations obtained by PIXE analysis of PM10 samples collected at the vicinity of lignite power plants and mines in Megalopolis, Greece", Grčka, Austrija, Hrvatska, „Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms", vol. 349, br. 15, str 114-124, travanj 2015. Dostupno na: https://doi.org/10.1016/j.nimb.2015.02.037. [Pristupljeno 18. 11. 2022.]

[7] C. H. Chen, Y. C. Chuang, C. C. Hsieh, C. S. Lee, „VOC characteristics and source apportionment at a PAMS site near an industrial complex in central Taiwan", Taiwan, „Atmospheric Pollution Research", vol. 10, br. 4, str 1060-1074, lipanj 2019. Dostupno na: https://doi.org/10.1016/j.apr.2019.01.014. [Pristupljeno 18. 11. 2022.]

[8] G. P. Humairoh, A. D. Syafei, M. Santoso, R. Boedisantoso, A. F. Assomadi, J. Hermana, „Identification of Trace Element in Ambient Air Case Study: Industrial Estate in Waru, Sidoarjo, East Java", Taiwan Association for Aerosol Research, Sukolilo, Surabaya i Bandung, Indonezija, „Aerosol and Air Quality Research", vol. 20, br. 9, str. 1910-1921, rujan 2020. Dostupno na: https://doi.org/10.4209/aaqr.2019.11.0590. [Pristupljeno 13. 11. 2022.]

[9] „Statistika" u *Hrvatska enciklopedija*, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. Dostupno na: https://www.enciklopedija.hr/natuknica.aspx?ID=57896. [Pristupljeno 29. 1. 2023.]

[10] „Interval pouzdanosti" u *Struna, Hrvatsko strukovno nazivlje*. Institut za hrvatski jezik i jezikoslovlje, 2011. Dostupno na: http://struna.ihjj.hr/naziv/interval-pouzdanosti/35536/. [Pristupljeno 6. 2. 2023.]

[11] J. L. Devore, K. N. Berk, „Modern Mathematical Statistics with Applications", drugo izdanje, Springer, New York, NY, 2021. Dostupno na: https://doi.org/10.1007/978-1-4614-0391-3.

[12] P. Cook, „Coverage versus Confidence A Tutorial", Tulsa, Oklahoma „The Mathematica Journal", vol. 23, ožujak 2021. Dostupno na: https://doi.org/10.3888/tmj.23-1.

[13] B. Efron, R. J. Tibshirani, „An Introduction to the Bootstrap", prvo izdanje, Chapman and Hall/CRC, 1994. Dostupno na: https://doi.org/10.1201/9780429246593.

[14] K. M. Ramachandran i C. P. Tsokos „Mathematical Statistics with Applications in R", treće izdanje, 2021. Academic Press. Dostupno na: https://doi.org/10.1016/B978-0-12-817815-7.00013-0.

[15] D. C. Carslaw, K. Ropkins, „openair — an R package for air qualitydata analysis." *Environmental Modelling & Software*. vol. 27-28, str. 52-61, 2012. Dostupno na: https://doi.org/10.1016/j.envsoft.2011.09.008. [Pristupljeno 21. 11. 2022.]

[16] D. C. Carslaw, „The openair manual — open-source tools for analysing air pollution data. Manual for version 2.6-6." .University of York. 2019. Dostupno na: https://davidcarslaw.com/project/openair/. [Pristupljeno 21. 11. 2022.]

[17] J. Joyce, „Bayes' Theorem",The Stanford Encyclopedia of Philosophy (Spring 2019 Edition). Dostupno na: https://plato.stanford.edu/archives/spr2019/entries/bayes-theorem/. [Pristupljeno 15. 11. 2022.]

[18] D. Katz, J. Baptista, S. P. Azen and M. C. Pike, „Obtaining Confidence Intervals for the Risk Ratio in Cohort Studies", *Biometrics*, vol. 34, br. 3, str. 469-474, rujan 1978. International Biometric Society. Dostupno na: http://www.jstor.org/stable/2530610. [Pristupljeno 15. 11. 2022.]

[19] M. W. Fagerland, S. Lydersen, P. Laake, „Recommended confidence intervals for two independent binomial proportions", Statistical methods in medical research, vol. 24, br. 2, str. 224-254, lipanj 2011. Dostupno na: https://doi.org/10.1177/09622802114154. [Pristupljeno 15. 11. 2022.]

Popis slika

Popis tablica

Dodatci

R program korišten za dobivanje grafova, intervala pouzdanosti i prekrivanja putem *bootstrap-a*:

```
################################################################################

# Program za provjeru intervala pouzdanosti putem bootstrapa

################################################################################

# ucitavanje potrebnih paketa
library(dplyr)
library(ggplot2)

dat <- read.csv(
  file.choose(),
  header = TRUE,
)

################################################################################
# Program je sastavljen od 3 glavna dijela:
#
# 1.) izracun CPF-a za sve podatke
# 2.) izracun CPF-a za skup podataka koji simulira mjerenje, ti podatci su
#     nasumicno uzorkovani iz ucitanog skupa podataka (nakon ciscenja), izracun
#     intervala pouzdanosti i crtanje grafa
# 3.) izracun prekrivanja
#
################################################################################
# uklanja sve redove gdje nedostaje barem 1 od podataka
df <- na.omit(data.frame(dat))

# uklanja sve podatke gdje je koncentracija manja od 1
df <- df[df[1] > 1, ]

# broj segmenata koji se generalno koristi je 4, 8, 12, 16, 24, 32 ili 36
segm = 8

#sirina segmenta u stupnjevima
theta = 360 / segm

# kvantil iznad kojeg se traze podatci o koncentraciji, promijenom varijable
# "kvantil", najcesce se koristi kvantil od 75% tj 0.75, odnosno, promatra se
# gornjih 25% vrijednost koncentracija
kvantil = 0.75
kvant = unname(quantile(df[3], probs = kvantil, na.rm = TRUE))

# izbor sigme, postavljena vrijednost je sigma = 1
s = 1

if (s == 1) {
alph = 0.3173 }
if(s == 2){
alph = 0.0455}
```

```
if(s == 3){
alph = 0.0027}


# alokacija memorije za sve vektore u petlji
n_21 = NA                  # broj elemenata u segmentu
n_elem_og = NA                  # broj elemenata vecih od kvant u segmentu
elem = NA
cpf_og = NA

# petlja sortira podatke prema smjeru vjetra i racuna CPF vrijednosti
for (i in 1:segm) {
  if (i == segm )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i - 2 ) )
                     & df[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

    n_elem_og[segm - 1] = length ( elem[,3] )
    n_21[segm - 1] = length ( subset(elem, elem[3] > kvant)[,3] )

    cpf_og[segm - 1] = n_21[segm - 1] / n_elem_og[segm - 1]
  }
  else if ( i < ( segm - 1 )  )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i - 1 ) )
                     & df[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

    n_elem_og[i] = length ( elem[,3] )
    n_21[i] = length ( subset(elem, elem[3] > kvant)[,3] )

    cpf_og[i] = n_21[i] / n_elem_og[i]

  }

  else if ( i == ( segm - 1 ) )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i ) )
                     | df[2] < ( theta / 2 ) )

    n_elem_og[segm] = length ( elem[,3] )
    n_21[segm] = length ( subset(elem, elem[3] > kvant)[,3] )

    cpf_og[segm] = n_21[segm] / n_elem_og[segm]

  }
}

# pretvara sve NaN i NA vrijednosti u 0
cpf_og[is.nan(cpf_og)] <- 0
cpf_og[is.na(cpf_og)] = 0

# kod uzet iz paketa ggiraphExtra, daje kružni graf
coord_radar <- function (theta = "x", start = -pi/4, direction = 1) {
  theta <- match.arg(theta, c("x", "y"))
  r <- if (theta == "x") "y" else "x"
  ggproto("CordRadar", CoordPolar, theta = theta, r = r, start = start,
          direction = sign(direction),
```

```
          is_linear = function(coord) TRUE)
}


# vektor sa smjerovima koje oznacava svaki segment
smjer = c(1:segm) * theta

# cpf_mod je data frame koji CPF vrijednostima pridruzuje smjer vjetra
cpf_mod_og <- data.frame(cpf_og, smjer)

# duplicira zadnji red iz cpf_mod data framea i pohranjujemo ga kao novi data
# frame. Oznacava stupac smjera s brojem 0 umjesto prethodnog broja te ga ljepi
# na prvo mjesto u data frameu. To je bitno zbog povezivanja tocaka/linija u
# konacnom grafickom prikazu
cpf_x <- cpf_mod_og[segm,]
cpf_x$smjer <- 0
cpf_new <- rbind(cpf_x, cpf_mod_og)

cpf_mod_og = cpf_new %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new)

# vektor koji odreduje vrijednosti na osima grafa
lim = c(1:(segm) -1) * theta

# graf CPF-a originalnih podataka
# coord_radar sa geom_line linijama i geom_point tockama
ggplot() +
  coord_radar(start = 0) +
  scale_x_discrete(expand = c(0, 0),
                   limits = lim) +
  #ylim(0, 0.4) +
  geom_line(data = cpf_mod_og, aes(x = smjer, y = cpf_og, group = 1,
                                   linetype = "CPF vrijednosti"), size=0.5) +
  geom_point(data = cpf_mod_og, aes(x = smjer, y = cpf_og, group = 1), size=
1.5) +
  scale_linetype_manual(" ", values = c("CPF vrijednosti" = 1)) +
  labs(x = " ", y = " CPF vjerojatnost ", subtitle =
         paste0(" CPF na ", kvantil*100,". percentilu, CPF = ",
                gsub("\\.", ",", kvant), ". N = ",  length(df[,3]) ),
       title=" ", x="smjer",caption=" ") +
  geom_point(aes(x = smjer, y = 0.25), color = "Red") +
  theme_bw()

################################################################################
################################################################################
#
#2.)
#
################################################################################
################################################################################
set.seed(19)

n_21 = NA
n_elem_smp_2 = NA
n_elem_smp = NA
```

```
elem = NA
cpf_smp = NA

# ova petlja racuna vrijednost CPF-a za uzorak koj ije uzorkovan iz orginalnih
# podataka

# broj elemenata u uzorkovan iz skupa koji predstavlja populaciju
n_elem0 = 300

# nasumicno uzorkuje skup od n_elem0 podataka iz populacije
rsmp0 <- df[sample(nrow(df),n_elem0, replace = F), ]
kvant_smp0 = unname(quantile(rsmp0[3], probs = kvantil, na.rm = TRUE))

#sortiranje po sektorima, izracun CPF-a i intervala pouzdanosti
for (i in 1:segm) {
  if (i == segm )
  {
    elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 2 ) )
                   & rsmp0[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

    n_elem_smp[segm - 1] = length ( elem[,3] )
    n_elem_smp_2[segm - 1] = length ( subset(elem, elem[3] > kvant_smp0)[,3] )

    cpf_smp[segm - 1] = n_elem_smp_2[segm - 1] / n_elem_smp[segm - 1]
  }
  else if ( i < ( segm - 1 )  )
  {
    elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 1 ) )
                   & rsmp0[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

    n_elem_smp[i] = length ( elem[,3] )
    n_elem_smp_2[i] = length ( subset(elem, elem[3] > kvant_smp0)[,3] )

    cpf_smp[i] = n_elem_smp_2[i] / n_elem_smp[i]

  }

  else if ( i == ( segm - 1 ) )
  {
    elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i ) )
                   | rsmp0[2] < ( theta / 2 ) )

    n_elem_smp[segm] = length ( elem[,3] )
    n_elem_smp_2[segm] = length ( subset(elem, elem[3] > kvant_smp0)[,3] )

    cpf_smp[segm] = n_elem_smp_2[segm] / n_elem_smp[segm]

  }
}

# pretvara sve NaN i NA vrijednosti u 0 u danim vektorima
cpf_smp[is.nan(cpf_smp)] <- 0
cpf_smp[is.na(cpf_smp)] = 0

# vektor sa smjerovima koje oznacava svaki segment
smjer = c(1:segm) * theta
```

```
# cpf_mod je data frame gdje pridruzujemo elemente iz vektora smjer uz
# odgovarajuce vrijednost cpf-a
cpf_mod_smp <- data.frame(cpf_smp, smjer)

# duplicira zadnji red iz cpf_mod data framea i pohranjuje ga kao  novi data
# frame, oznacava stupac smjera s brojem 0 umjesto prethodnog broja te ga
# priljepljujemo na prvo mjesto u data frameu.
cpf_x <- cpf_mod_smp[segm,]
cpf_x$smjer <- 0
cpf_new <- rbind(cpf_x, cpf_mod_smp)

# ovaj dio povezuje te vrijednosti u data frameu te omogucava prikaz s povezanim
# tockama
cpf_mod_smp = cpf_new %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new)

# vektor potreban za korektno oznacavanje grafa
lim = c(1:(segm) -1) * theta

################################################################################
# kod za dobivanje intervala pouzdanosti putem bootstrappa
################################################################################

#alokacija memorije za bootstrap cpf petlju
cpf_b <- c(1:segm)*0

n_11 = 0
# cpf_rs ce biti vektor za pohranjivanje svih podataka dobivenih petljom
cpf_rs = NA

# broj ponavljanja petlje, odnosno broja bootstrapa, ako se radi sa CL = 95,
# onda reps treba biti barem 20
reps = 1000

for (k in 1:reps){
  #nasumicno uzorkuje "elem" elemenata sa zamjenom iz df
  rsmpB <- rsmp0[sample(nrow(rsmp0), n_elem0, replace = T), ]

  #kvant za k-ti nasumicni uzorak dobiven bootstrappom
  kvant_r = unname(quantile(rsmpB[3], probs = kvantil, na.rm = TRUE))

  # petlja za sortiranje po segmentima i izracun vrijednosti CPF-a
  for (i in 1:segm) {
    if (i == segm )
    {
      elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i - 2 ) )
                     & rsmpB[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

      n_11[segm - 1] = length ( elem[,3] )
      n_21[segm - 1] = length ( subset(elem, elem[3] > kvant_r)[,3] )

      cpf_b[segm - 1] = n_21[segm - 1] / n_11[segm - 1]
    }
    else if ( i < ( segm - 1 )  )
    {
```

```
      elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i - 1 ) )
                      & rsmpB[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

      n_11[i] = length ( elem[,3] )
      n_21[i] = length ( subset(elem, elem[3] > kvant_r)[,3] )

      cpf_b[i] = n_21[i] / n_11[i]

    }

    else if ( i == ( segm - 1 ) )
    {
      elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i ) )
                      | rsmpB[2] < ( theta / 2 ) )

      n_11[segm] = length ( elem[,3] )
      n_21[segm] = length ( subset(elem, elem[3] > kvant_r)[,3] )

      cpf_b[segm] = n_21[segm] / n_11[segm]

    }
  }

  cpf_rs=c(cpf_rs,cpf_b)
}
cpf_rs[is.nan(cpf_rs)] <- 0
cpf_rs[is.na(cpf_rs)] = 0

# vektor sa svim cpf vrijednostima za sve iteracije petlje
cpf_rs = cpf_rs[2:(reps*segm+1)]

# cpf_smjerovi je data frame sa sortiranim nizom dobivenih podataka iz petlje
cpf_smjerovi = data.frame(matrix(NA, nrow = reps, ncol = segm))

for (i in 1:segm)
{
  cpf_smjerovi[,i] = sort(cpf_rs[seq( i ,length(cpf_rs), segm)])
}

#13.3.1. postupak, knjiga: Ramachandran, Tsokos
# vektori s donjim i gornjim vrijednostima intervala pouzdanosti za svaki smjer
# vjetra
# za CL = 68, koriste se vrijednosti 0.16 i 0.84 za donji i gornji interval
# za CL = 95, koriste se vrijednostiili 0.025 i 0.975

#donji i gornji interval pouzdanosti CPF-a
bl_cpf = cpf_smjerovi[round(alph/2 * (reps+1)),]
bu_cpf = cpf_smjerovi[round((1- alph/2) * (reps+1)),]

# vektor smjera u stupnjevima, koristi se za oznacavanje binova
smjer = c(1:segm) * theta


cpf_mod_bl <- data.frame( bl =  t( bl_cpf )[1:segm], smjer)
cpf_mod_bl[is.na(cpf_mod_bl)] = 0

cpf_x_bl <- cpf_mod_bl[cpf_mod_bl$smjer==segm*theta, ]
```

```
cpf_x_bl$smjer <- 0
cpf_new_bl <- rbind(cpf_x_bl, cpf_mod_bl)

cpf_mod_bl = cpf_new_bl %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new_bl)


cpf_mod_bu <- data.frame( bu =  t( bu_cpf )[1:segm], smjer)
cpf_mod_bu[is.na(cpf_mod_bu)] = 0

cpf_x_bu <- cpf_mod_bu[cpf_mod_bu$smjer==segm*theta, ]
cpf_x_bu$smjer <- 0
cpf_new_bu <- rbind(cpf_x_bu, cpf_mod_bu)

cpf_mod_bu = cpf_new_bu %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new_bu)


#graf
ggplot() +

  # primjenjuje coord_radar funkciju --> daje oblik ruze vjetrova
  coord_radar(start = 0) +

  # ponekad pozeljno ukljuciti i namjestiti gornju granicu za slucaj s puno
  # segmenata i velikim gornjim intervalom pouzdanosti zbog preglednosti
  #ylim(0, 0.7) +

  #Imenuje smjerove vjetra na ruzi vjetrova
  scale_x_discrete(expand = c(0, 0),
                   limits = lim) +

  # linija koja povezuje podatke
  geom_line(data = cpf_mod_smp, aes(x = smjer, y = cpf_smp, group = 1,
                                    linetype = "CPF vrijednosti"), size=0.5) +

  # tocke na grafu
  # mozda ih je pozeljno iskljuciti ili smanjiti im velicinu za slucaj s
  # velikim brojem segmenata
  geom_point(data = cpf_mod_smp, aes(x = smjer, y = cpf_smp, group = 1), size=
1.5) +

  geom_line(data = cpf_mod_bl, aes(x = smjer, y = bl, group = 1,
                                     linetype = "Donji interval pouzdanosti"),
size=0.5) +
  geom_point(data = cpf_mod_bl, aes(x = smjer, y = bl, group = 1), size= 1.5) +

  geom_line(data = cpf_mod_bu, aes(x = smjer, y = bu, group = 1,
                                     linetype = "Gornji interval pouzdanosti"),
size=0.5) +
  geom_point(data = cpf_mod_bu, aes(x = smjer, y = bu, group = 1), size= 1.5)  +

  # namjesta podatke za korektno imenovanje legende
```

```
  scale_linetype_manual(" ", values = c("CPF vrijednosti" = 1,
                                        "Donji interval pouzdanosti" = 2,
                                        "Gornji interval pouzdanosti" = 3)) +

  # namjesta osi i naslov grafa izbacuje 75 i pretvara da segm dio bude
varijabla
  labs(x = " ", y = " CPF vjerojatnost ",
       subtitle = paste0(" CPF na ", kvantil*100,". percentilu, CPF = ",

                         #pretvara decimalnu tocku u zarez
                         gsub("\\.", ",", kvant_smp0), ". N = ", n_elem0),
       title=" ", x="smjer",caption=" ") +

  # crvene tocke
  geom_point(aes(x = smjer, y = 0.25), color = "Red") +

  # bijela pozadina grafa
  theme_bw()

################################################################################
################################################################################
#
# 3.) Prekrivanje
#
################################################################################
################################################################################
set.seed(19)


coverage = c(1:segm)*0
cpf_p = 0

for (l in 1:cov_reps ){

cpf_rs = NA

rsmp0 <- df[sample(nrow(df),n_elem0, replace = F), ]

# petlja za racunanje prekrivanje uz pomoc bootstrap-a
for (k in 1:reps){
  rsmpB <- rsmp0[sample(nrow(rsmp0), n_elem0, replace = T), ]

  kvant_r = unname(quantile(rsmpB[3], probs = kvantil, na.rm = TRUE))

  for (i in 1:segm) {
    if (i == segm )
    {

      n_elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i - 2 ) )
                       & rsmpB[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

      n_11[segm - 1] = length ( n_elem[,3] )
      n_21[segm - 1] = length ( subset(n_elem, n_elem[3] > kvant_r)[,3] )

      cpf_p[segm - 1] = n_21[segm - 1] / n_11[segm - 1]
      cpf_p[segm - 1][is.na(cpf_p[segm - 1])] = 0
```

```
    }
    else if ( i < ( segm - 1 )  )
    {
      n_elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i - 1 ) )
                       & rsmpB[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

      n_11[i] = length ( n_elem[,3] )
      n_21[i] = length ( subset(n_elem, n_elem[3] > kvant_r)[,3] )

      cpf_p[i] = n_21[i] / n_11[i]
      cpf_p[i][is.na(cpf_p[i])] = 0
    }

    else if ( i == ( segm - 1 ) )
    {
      n_elem = subset( rsmpB, rsmpB[2] >= ( theta / 2 + theta * ( i ) )
                       | rsmpB[2] < ( theta / 2 ) )

      n_11[segm] = length ( n_elem[,3] )
      n_21[segm] = length ( subset(n_elem, n_elem[3] > kvant_r)[,3] )

      cpf_p[segm] = n_21[segm] / n_11[segm]
      cpf_p[segm][is.na(cpf_p[segm])] = 0

    }
  }

  cpf_rs=c(cpf_rs,cpf_p)
}

cpf_rs[is.nan(cpf_rs)] <- 0
cpf_rs[is.na(cpf_rs)] = 0

cpf_rs = cpf_rs[2:(reps*segm+1)]


cpf_smjerovi = data.frame(matrix(NA, nrow = reps, ncol = segm))


for (i in 1:segm)
{
  cpf_smjerovi[,i] = sort(cpf_rs[seq( i ,length(cpf_rs), segm)])
}


bl_cpf = cpf_smjerovi[round(alph/2 * (reps+1)),]
tl_cpf = cpf_smjerovi[round((1 - alph/2) * (reps+1)),]


for (i in 1:segm)
{
  if (cpf_og[i] > bl_cpf[i] & cpf_og[i] < tl_cpf[i])
  {
    coverage[i] = coverage[i] + 1
  }
  else
  {
```

```
    coverage[i] = coverage[i]
  }
}

}

coverage = coverage / cov_reps
coverage

#https://content.wolfram.com/uploads/sites/19/2021/02/Cook.pdf
sig_c = s*sqrt(coverage*(1-coverage)/cov_reps) # CL= 68%
sig_c
##intervali pouzdanosti
cov_int_l = coverage - sig_c
cov_int_u = coverage + sig_c


cov_mod <- data.frame(smjer, n_elem_smp_2, cpf_smp ,coverage, sig_c, cov_int_l,
cov_int_u)
cov_mod

cpf_mod_bl
cpf_mod_bu
################################################################################
```

R program korišten za dobivanje grafova, intervala pouzdanosti i prekrivanja putem omjera proporcija i Katzovog logaritma:

```
##############################################################################

# Program za provjeru intervala pouzdanosti putem omjera
# proporcija i Karzovog logaritma.

##############################################################################

# učitavanje potrebnih paketa
library(dplyr)
library(ggplot2)

# treba ucitati podatke u obliku tablice tako da se u prvom stupcu nalaze
# podatci o brzini vjetra, u drugom stupcu smjer, a u trecem koncentracije
dat <- read.csv(
  file.choose(),
  header = TRUE,
)

##############################################################################
# Program je sastavljen od 3 glavna dijela:
#
# 1.) izračun CPF-a za sve podatke
# 2.) izračun CPF-a za skup podataka koji simulira mjerenje, ti podatci su
#     nasumično uzorkovani iz učitanog skupa podataka (nakon čišćenja), izračun
#     intervala pouzdanosti i crtanje grafa
# 3.) izračun prekrivanja
#
##############################################################################

# uklanja sve redove gdje nedostaje barem 1 od podataka
df <- na.omit(data.frame(dat))

# uklanja sve podatke gdje je koncentracija manja od 1
df <- df[df[1] > 1, ]

# broj segmenata koji se generalno koristi je 4, 8, 12, 16, 24, 32 ili 36
segm = 8

#sirina segmenta u stupnjevima
theta = 360 / segm

# kvantil iznad kojeg se traze podatci o koncentraciji, promijenom varijable
# "kvantil", najcesce se koristi kvantil od 75% tj 0.75, odnosno, promatra se
```

```
# gornjih 25% vrijednost koncentracija
kvantil = 0.75
kvant = unname(quantile(df[3], probs = kvantil, na.rm = TRUE))

# izbor sigme, postavljena vrijednost je sigma = 1
s = 1

if (s == 1) {z = 1}
if (s == 2) {z = 2}
if (s == 3) {z = 3}

# alokacija memorije za cpf vektor
cpf_og = c(1:segm)

# duljina svih mjerenja koncentracija manjih i vecih od zadanog kvantila, te
# alokacija memorije za sve vektore u petlji
n_1p_og = length (subset(df, df[3] <= kvant)[,3] )
n_2p_og = length (subset(df, df[3] > kvant)[,3] )
n_21_og = NA
n_11_og = NA

elem = NA
phi = NA
cpf_og = NA

# petlja sortira podatke prema smjeru vjetra i računa CPF vrijednosti
for (i in 1:segm) {
  if (i == segm )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i - 2 ) )
                     & df[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

    n_11_og[segm - 1] = length ( subset(elem, elem[3] <= kvant)[,3] )
    n_21_og[segm - 1] = length ( subset(elem, elem[3] > kvant)[,3] )
    phi[segm - 1] = (( n_11_og[segm - 1])/(n_1p_og)) / ((n_21_og[segm -
1])/(n_2p_og))

    cpf_og[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm - 1])
  }
  else if ( i < ( segm - 1 )  )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i - 1 ) )
                     & df[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

    n_11_og[i] = length ( subset(elem, elem[3] <= kvant)[,3] )
    n_21_og[i] = length ( subset(elem, elem[3] > kvant)[,3] )
    phi[i] = (( n_11_og[i])/(n_1p_og)) / ((n_21_og[i])/(n_2p_og))

    cpf_og[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i])
  }

  else if ( i == ( segm - 1 ) )
  {
    elem = subset( df, df[2] >= ( theta / 2 + theta * ( i ) )
                     | df[2] < ( theta / 2 ) )

    n_11_og[segm] = length ( subset(elem, elem[3] <= kvant)[,3] )
```

```
    n_21_og[segm] = length ( subset(elem, elem[3] > kvant)[,3] )
    phi[segm] = (( n_11_og[segm])/(n_1p_og)) / ((n_21_og[segm])/(n_2p_og))

    cpf_og[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm])
  }
}

# pretvara sve NaN i NA vrijednosti u 0
cpf_og[is.nan(cpf_og)] <- 0
cpf_og[is.na(cpf_og)] = 0

# kod uzet iz paketa ggiraphExtra, daje kružni graf
coord_radar <- function (theta = "x", start = -pi/4, direction = 1) {
  theta <- match.arg(theta, c("x", "y"))
  r <- if (theta == "x") "y" else "x"
  ggproto("CordRadar", CoordPolar, theta = theta, r = r, start = start,
          direction = sign(direction),
          is_linear = function(coord) TRUE)
}

# vektor sa smjerovima koje oznacava svaki segment
smjer = c(1:segm) * theta

# cpf_mod je data frame koji CPF vrijednostima pridruzuje smjer vjetra
cpf_mod_og <- data.frame(cpf_og, smjer)

# duplicira zadnji red iz cpf_mod data framea i pohranjujemo ga kao novi data
# frame. Oznacava stupac smjera s brojem 0 umjesto prethodnog broja te ga ljepi
# na prvo mjesto u data frameu. To je bitno zbog povezivanja tocaka/linija u
# konacnom grafickom prikazu
cpf_x <- cpf_mod_og[segm,]
cpf_x$smjer <- 0
cpf_new <- rbind(cpf_x, cpf_mod_og)

cpf_mod_og = cpf_new %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new)

# vektor koji odreduje vrijednosti na osima grafa
lim = c(1:(segm) -1) * theta

#graf CPF-a originalnih podataka
ggplot() +
  coord_radar(start = 0) +
  scale_x_discrete(expand = c(0, 0),
                   limits = lim) +
  ylim(0, 0.4) +
  geom_line(data = cpf_mod_og, aes(x = smjer, y = cpf_og, group = 1,
                                   linetype = "CPF vrijednosti"), size=0.5) +
  geom_point(data = cpf_mod_og, aes(x = smjer, y = cpf_og, group = 1), size=
1.5) +
  scale_linetype_manual(" ", values = c("CPF vrijednosti" = 1)) +
  labs(x = " ", y = " CPF vjerojatnost ", subtitle =
         paste0(" CPF na ", kvantil*100,". percentilu, CPF = ",
                gsub("\\.", ",", kvant), ". N = ",  length(df[,3]) ),
       title=" ", x="smjer",caption=" ") +
```

```
  geom_point(aes(x = smjer, y = 0.25), color = "Red") +
  theme_bw()

################################################################################
################################################################################
#
#2.)
#
################################################################################
################################################################################
set.seed(19)

# alokacija memorije za CPF vektor nasumicnog uzorka
cpf_smp = c(1:segm)

# broj elemenata u uzorkovan iz skupa koji predstavlja populaciju
n_elem0 = 300

# nasumicno uzorkuje skup od n_elem0 podataka iz populacije
rsmp0 <- df[sample(nrow(df), n_elem0, replace = F), ]
kvant_rsmp0 = unname(quantile(rsmp0[3], probs = kvantil, na.rm = TRUE))

# duljina svih mjerenja koncentracija manjih i vecih od zadanog kvantila, te
# alokacija memorije za sve vektore u petlji
n_1p_smp = length (subset(rsmp0, rsmp0[3] <= kvant_rsmp0)[,3] )
n_2p_smp = length (subset(rsmp0, rsmp0[3] > kvant_rsmp0)[,3] )
n_21_smp = 0 * c(1:segm)
n_11_smp = 0 * c(1:segm)

phi = NA
cpf_smp = NA
cpf_b_smp = NA
cpf_p_smp = NA

#sortiranje po sektorima, izračun CPF-a i intervala pouzdanosti
for (i in 1:segm) {
  if (i == segm )
  {
    n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 2 ) )
                     & rsmp0[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

    n_11_smp[segm - 1] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
    n_21_smp[segm - 1] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
    phi[segm - 1] = ((n_11_smp[segm - 1])/(n_1p_smp))/((n_21_smp[segm -
1])/(n_2p_smp))

    cpf_smp[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm - 1])

    # donji interval pouzdanosti
    cpf_b_smp[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm -
1]*exp( z *
                                          sqrt( 1 / ( n_11_smp[segm - 1]) + 1 /
(n_21_smp[segm - 1]) -
                                                  1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
    # gornji interval pouzdanosti
```

```
    cpf_p_smp[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm -
1]*exp( -z *
                                                sqrt( 1 / ( n_11_smp[segm - 1]) + 1 /
(n_21_smp[segm - 1]) -
                                                        1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
  }
  else if ( i < ( segm - 1 )  )
  {
    n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 1 ) )
                    & rsmp0[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

    n_11_smp[i] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
    n_21_smp[i] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
    phi[i] = (( n_11_smp[i])/(n_1p_smp)) / ((n_21_smp[i])/(n_2p_smp))

    cpf_smp[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i])

    cpf_b_smp[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i]*exp( z *
                                                sqrt( 1 / ( n_11_smp[i]) + 1 /
(n_21_smp[i]) -
                                                        1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
    cpf_p_smp[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i]*exp( -z *
                                                sqrt( 1 / ( n_11_smp[i]) + 1 /
(n_21_smp[i]) -
                                                        1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
  }

    else if ( i == ( segm - 1 ) )
    {
      n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i ) )
                      | rsmp0[2] < ( theta / 2 ) )

      n_11_smp[segm] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
      n_21_smp[segm] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
      phi[segm] = (( n_11_smp[segm])/(n_1p_smp)) / ((n_21_smp[segm])/(n_2p_smp))

      cpf_smp[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm])

      cpf_b_smp[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm]*exp( z
*
                                                sqrt( 1 / ( n_11_smp[segm]) + 1 /
(n_21_smp[segm]) -
                                                        1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
      cpf_p_smp[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm]*exp( -
z *
                                                sqrt( 1 / ( n_11_smp[segm]) + 1 /
(n_21_smp[segm]) -
                                                        1 / (n_1p_smp) - 1 /
(n_2p_smp) ) ) )
  }
}

# pretvara sve NaN i NA vrijednosti u 0 u danim vektorima
```

```
cpf_smp[is.nan(cpf_smp)] <- 0
cpf_b_smp[is.nan(cpf_b_smp)] <- 0
cpf_p_smp[is.nan(cpf_p_smp)] <- 0

cpf_smp[is.na(cpf_smp)] <- 0
cpf_b_smp[is.na(cpf_b_smp)] <- 0
cpf_p_smp[is.na(cpf_p_smp)] <- 0

# uvodi coord_radar funkciju koja je kljucna za crtanje poalrnog grafa
#coord_radar <- function (theta = "x", start = 0, direction = 1) {
#  theta <- match.arg(theta, c("x", "y"))
#  r <- if (theta == "x") "y" else "x"
#  ggproto("CordRadar", CoordPolar, theta = theta, r = r, start = start,
#          direction = sign(direction),
#          is_linear = function(coord) TRUE)
#}

# vektor smjera u stupnjevima, koristi se za oznacavanje binova
smjer = c(1:segm) * theta

cpf_mod_smp <- data.frame(cpf_smp, smjer)

# duplicira zadnji red iz cpf_mod data framea i pohranjuje ga kao  novi data
# frame, oznacava stupac smjera s brojem 0 umjesto prethodnog broja te ga
# priljepljujemo na prvo mjesto u data frameu.
cpf_x <- cpf_mod_smp[segm,]
cpf_x$smjer <- 0
cpf_new <- rbind(cpf_x, cpf_mod_smp)

# ovaj dio povezuje te vrijednosti u data frameu te omogucava prikaz s povezanim
# tockama
cpf_mod_smp = cpf_new %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new)

# vektor potreban za korektno oznacavanje grafa
lim = c(1:(segm) -1) * theta

cpf_b_mod <- data.frame(cpf_b_smp, smjer)
cpf_b_mod[is.na(cpf_b_mod)] = 0

cpf_x_bl <- cpf_b_mod[cpf_b_mod$smjer==segm*theta, ]
cpf_x_bl$smjer <- 0
cpf_new_bl <- rbind(cpf_x_bl, cpf_b_mod)

cpf_b_mod = cpf_new_bl %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new_bl)


cpf_p_mod <- data.frame(cpf_p_smp, smjer)
cpf_p_mod[is.na(cpf_p_mod)] = 0

cpf_x_pl <- cpf_p_mod[cpf_p_mod$smjer==segm*theta, ]
cpf_x_pl$smjer <- 0
```

```
cpf_new_pl <- rbind(cpf_x_pl, cpf_p_mod)

cpf_p_mod = cpf_new_pl %>%
  filter(smjer == "N") %>%
  mutate(smjer = "idk") %>%
  rbind(cpf_new_pl)

#graf
ggplot() +

  # primjenjuje coord_radar funkciju --> daje oblik ruze vjetrova
  coord_radar(start = 0) +

  # ponekad pozeljno ukljuciti i namjestiti gornju granicu za slucaj s puno
  # segmenata i velikim gornjim intervalom pouzdanosti zbog preglednosti
  #ylim(0, 0.5) +

  # imenuje smjerove vjetra na ruzi vjetrova
  scale_x_discrete(expand = c(0, 0),
                   limits = lim ) +

  # linija koja povezuje podatke
  geom_line(data = cpf_mod_smp, aes(x = smjer, y = cpf_smp, group = 1,
                                    linetype = "CPF vrijednosti"), size=0.5) +

  # tocke na grafu
  # mozda ih je pozeljno iskljuciti ili smanjiti im velicinu za slucaj s
  # velikim brojem segmenata
  geom_point(data = cpf_mod_smp, aes(x = smjer, y = cpf_smp, group = 1), size=
1.5) +

  geom_line(data = cpf_b_mod, aes(x = smjer, y = cpf_b_smp, group = 1,
                                    linetype = "Donji interval pouzdanosti"),
size=0.5) +
  geom_point(data = cpf_b_mod, aes(x = smjer, y = cpf_b_smp, group = 1), size=
1.5) +

  geom_line(data = cpf_p_mod, aes(x = smjer, y = cpf_p_smp, group = 1,
                                    linetype = "Gornji interval pouzdanosti"),
size=0.5) +
  geom_point(data = cpf_p_mod, aes(x = smjer, y = cpf_p_smp, group = 1), size=
1.5)  +

  # namjesta podatke za ispravno imenovanje legende
  scale_linetype_manual(" ", values = c("CPF vrijednosti" = 1,
                                        "Donji interval pouzdanosti" = 2,
                                        "Gornji interval pouzdanosti" = 3)) +

  # namjesta osi i naslov grafa izbacuje 75 i pretvara da segm dio bude
varijabla
  labs(x = " ", y = " CPF vjerojatnost ",
       subtitle = paste0(" CPF na ", kvantil*100,". percentilu, CPF = ",

                         #pretvara decimalnu tocku u zarez
                         gsub("\\.", ",", kvant_rsmp0), ". N = ", n_elem0 ),
       title=" ", x="smjer",caption=" ") +
```

```
  # crvene tocke
  geom_point(aes(x = smjer, y = 0.25), color = "Red") +

  # bijela pozadina grafa
  theme_bw()

################################################################################
################################################################################
#
# 3.) Prekrivanje
#
################################################################################
################################################################################
set.seed(19)

# alokacija memorije za sve vektore u petlji
cpf_c = c(1:segm)
n_21 = NA
n_11 = NA

phi = NA
cpf_b = NA
cpf_p = NA

#broj ponavljanja petlje za prekrivanje
cov_reps = 1000
coverage = 0 * c(1:segm)

# petlja koja odreduje prekrivanje
for (c in 1:cov_reps){

rsmp0 <- df[sample(nrow(df), n_elem0, replace = F), ]

kvant_rsmp0 = unname(quantile(rsmp0[3], probs = kvantil, na.rm = TRUE))

n_1p = length (subset(rsmp0, rsmp0[3] <= kvant_rsmp0)[,3] )
n_2p = length (subset(rsmp0, rsmp0[3] > kvant_rsmp0)[,3] )

for (i in 1:segm) {
  if (i == segm )
  {
    n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 2 ) )
                     & rsmp0[2] < ( theta * 1.5 + theta * ( i - 2 ) ) )

    n_11[segm - 1] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
    n_21[segm - 1] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
    phi[segm - 1] = (( n_11[segm - 1])/(n_1p)) / ((n_21[segm - 1])/(n_2p))

    cpf_c[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm - 1])

    # donji interval pouzdanosti
    cpf_b[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm - 1]*exp(
z *
                           sqrt( 1 / ( n_11[segm - 1]) + 1 / (n_21[segm - 1]) -
                                                  1 / (n_1p) - 1 / (n_2p) ) ) )
    # gornji interval pouzdanosti
```

```
    cpf_p[segm - 1] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm - 1]*exp(
-z *
                              sqrt( 1 / ( n_11[segm - 1]) + 1 / (n_21[segm - 1]) -
                                                    1 / (n_1p) - 1 / (n_2p) ) ) )
  }
  else if ( i < ( segm - 1 )  )
  {
    n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i - 1 ) )
                     & rsmp0[2] < ( theta * 1.5 + theta * ( i - 1 ) ) )

    n_11[i] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
    n_21[i] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
    phi[i] = (( n_11[i])/(n_1p)) / ((n_21[i])/(n_2p))

    cpf_c[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i])

    cpf_b[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i]*exp( z *
                                        sqrt( 1 / ( n_11[i]) + 1 / (n_21[i]) -
                                                    1 / (n_1p) - 1 / (n_2p) ) ) )
    cpf_p[i] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[i]*exp( -z *
                                        sqrt( 1 / ( n_11[i]) + 1 / (n_21[i]) -
                                                    1 / (n_1p) - 1 / (n_2p) ) ) )
  }

  else if ( i == ( segm - 1 ) )
  {
    n_elem = subset( rsmp0, rsmp0[2] >= ( theta / 2 + theta * ( i ) )
                     | rsmp0[2] < ( theta / 2 ) )

    n_11[segm] = length ( subset(n_elem, n_elem[3] <= kvant_rsmp0)[,3] )
    n_21[segm] = length ( subset(n_elem, n_elem[3] > kvant_rsmp0)[,3] )
    phi[segm] = (( n_11[segm])/(n_1p)) / ((n_21[segm])/(n_2p))

    cpf_c[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm])

    cpf_b[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm]*exp( z *
                                    sqrt( 1 / ( n_11[segm]) + 1 / (n_21[segm]) -
                                                1 / (n_1p) - 1 / (n_2p) ) ) )
    cpf_p[segm] = 1 / (1 + ( kvantil / ( 1 - kvantil) ) * phi[segm]*exp( -z *
                                    sqrt( 1 / ( n_11[segm]) + 1 / (n_21[segm]) -
                                                1 / (n_1p) - 1 / (n_2p) ) ) )
  }
}
cpf_c[is.nan(cpf_c)] <- 0
cpf_b[is.nan(cpf_b)] <- 0
cpf_p[is.nan(cpf_p)] <- 0

for (i in 1:segm)
{
  if (cpf_og[i] > cpf_b[i] & cpf_og[i] < cpf_p[i])
  {
    coverage[i] = coverage[i] + 1
  }
  else
  {
    coverage[i] = coverage[i]
  }
```

```
}

}

coverage = coverage / cov_reps

#https://content.wolfram.com/uploads/sites/19/2021/02/Cook.pdf
sig_c = s*sqrt(coverage*(1-coverage)/cov_reps) # CL= 68%
sig_c
#intervali pouzdanosti
cov_int_l = coverage - sig_c
cov_int_u = coverage + sig_c


cov_mod <- data.frame(smjer, n_11_smp, n_21_smp, cpf_smp, cpf_b_smp, cpf_p_smp,
                      coverage, cov_int_l, cov_int_u)
cov_mod
################################################################################
```