

Simulacija redova čekanja u R programskom jeziku: Simmer paket

Jakšeković, Luka

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:926201>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-28**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



SVEUČILIŠTE JURJA DOBRILE U PULI
FAKULTET INFORMATIKE

LUKA JAKŠEKVIĆ

SIMULACIJA REDOVA ČEKANJA U R PROGRAMSKOM JEZIKU: SIMMER PAKET

Završni rad

Pula, rujan 2024.

SVEUČILIŠTE JURJA DOBRILE U PULI
FAKULTET INFORMATIKE

LUKA JAKŠEKOVIĆ

**SIMULACIJA REDOVA ČEKANJA U R PROGRAMSKOM JEZIKU: SIMMER
PAKET**

Završni rad

JMBAG: 0303069776, redoviti student

Studijski smjer: Informatika

Kolegij: Modeliranje i simulacija

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Darko Etinger

Pula, rujan 2024.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Luka Jakšeković, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio završnog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, 23. rujna, 2024. godine



IZJAVA O KORIŠTENJU AUTORSKOGA DJELA

Ja, Luka Jakšeković, dajem odobrenje Sveučilištu Jurja Dobrile u Puli, nositelju prava korištenja, da moj završni rad pod nazivom „Simulacija redova čekanja u R programskom jeziku: simmer paket“ upotrijebi da tako navedeno autorsko djelo objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te preslika u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu sa Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

Potpis

U Puli, 23.09.2024

SAŽETAK

U današnjem sve složenijem poslovnom okruženju, redovi čekanja predstavljaju ključni element koji utječe na zadovoljstvo korisnika, učinkovitost poslovanja i iskorištenost resursa. Redovi čekanja se često pojavljuju u različitim sektorima, od maloprodaje i zdravstvenih ustanova do transporta i telekomunikacija. U ovom završnom radu, istražujemo kako se R programski jezik i paket "simmer" mogu koristiti za dublje razumijevanje, modeliranje i optimizaciju redova čekanja u različitim kontekstima. R je moćan jezik za analizu podataka i statistiku, dok je "simmer" paket specijaliziran za simulaciju događaja. Kombinacija ova dva alata omogućuje nam da stvorimo realistične modele redova čekanja i testiramo različite scenarije kako bismo bolje razumjeli njihovo ponašanje. U teorijskom okviru ovog istraživanja, razmatramo osnove teorije redova čekanja i koncepte simulacije. Otkrivamo kako se statistički modeli mogu primijeniti na stvarne situacije s redovima čekanja kako bismo ih bolje razumjeli i optimizirali. Također, istražujemo prednosti R programskog jezika i paketa "simmer" u ovom kontekstu, ističući njihovu fleksibilnost i mogućnost prilagodbe različitim scenarijima. U metodologiji istraživanja, detaljno definiramo korake za postavljanje simulacijskog okruženja u R programskom jeziku. Ovdje razrađujemo proces modeliranja redova čekanja, uključujući definiranje entiteta, resursa i događaja. Također, pristupamo parametrima simulacije kako bismo stvorili realistične scenarije za ispitivanje. Na taj način osiguravamo da naša simulacija odražava stvarne uvjete i omogućava nam testiranje različitih strategija upravljanja redovima čekanja. Ovaj rad ima za cilj pružiti dublje razumijevanje redova čekanja i kako ih možemo optimizirati pomoću R programskog jezika i paketa "simmer". Kroz praktične primjere i analize u stvarnim scenarijima, ilustrirat ćemo kako ovi alati mogu pridonijeti boljem upravljanju redovima čekanja i poboljšati iskustvo korisnika, što je ključno za uspješno poslovanje u današnjem konkurentnom okruženju.

Ključne riječi: redovi čekanja, R programski jezik, paket „simmer“, simulacijsko okruženje

ABSTRACT

In today's increasingly complex business environment, queues represent a key element that affects customer satisfaction, business efficiency and resource utilization. Queues often occur in a variety of sectors, from retail and healthcare to transportation and telecommunications. In this final paper, we explore how the R programming language and the "simmer" package can be used for a deeper understanding, modeling and optimization of queues in different contexts. R is a powerful language for data analysis and statistics, while "simmer" is a specialized package for event simulation. The combination of these two tools allows us to create realistic queue models and test different scenarios to better understand their behavior. In the theoretical framework of this research, we consider the basics of queuing theory and simulation concepts. We discover how statistical models can be applied to real-world queuing situations to better understand and optimize them. We also explore the advantages of the R programming language and the "simmer" package in this context, highlighting their flexibility and adaptability to different scenarios. In the research methodology, we define in detail the steps for setting up the simulation environment in the R programming language. Here we elaborate on the process of modeling queues, including defining entities, resources and events. We also access simulation parameters to create realistic test scenarios. In this way, we ensure that our simulation reflects real conditions and allows us to test different queue management strategies. This paper aims to provide a deeper understanding of queues and how we can optimize them using the R programming language and the "simmer" package. Through practical examples and analysis in real scenarios, we will illustrate how these tools can contribute to better queue management and improve the customer experience, which is crucial for successful business in today's competitive environment.

Keywords: queues, R programming language, "simmer" package, simulation environment

SADRŽAJ

1. UVOD	1
2. OSNOVNE TEORIJE	2
2.1. Teorija redova čekanja.....	2
2.2. Koncept simulacije.....	3
3. METODOLOGIJA.....	6
4. STUDIJA SLUČAJA	10
5. RAZRADA I ANALIZA SIMULACIJSKOG KODA.....	11
5.1. Korišteni paketi i definiranje parametara	11
5.2. Definiranje putanje pacijenta	13
5.3. Stvaranje simulacijskog okruženja.....	13
5.4. Pokretanje simulacije i dohvaćanje podataka.....	14
5.5. Analiza redova čekanja.....	15
5.6. Vizualizacija rezultata	16
5.7. Rezultati Simulacije	18
5.8. Novi parametri.....	19
6. ZAKLJUČAK	22
7. LITERATURA	23
8. POPIS SLIKA	24

1. UVOD

Simulacije su moćan alat za istraživanje i razumijevanje složenih sustava u različitim područjima. One omogućuju analizu ponašanja i performansi sustava u različitim scenarijima, što može biti izuzetno korisno za donošenje informiranih odluka. U ovom radu, fokusirat ćemo se na primjenu simulacija u optimizaciji procesa u zdravstvenim ustanovama, posebno u kontekstu upravljanja resursima i radnih tokova.

Korištenjem alata R-ovog paketa *simmer*, istražiti ćemo kako se različiti čimbenici, kao što su dostupnost osoblja i radno vrijeme, odražavaju na efikasnost i kvalitetu usluga. Za primjer, koristit ćemo kod koji simulira radno vrijeme i resurse u hipotetskoj zdravstvenoj ustanovi. Ova simulacija omogućava uvid u to kako različiti scenariji mogu utjecati na rad i učinkovitost, pružajući vrijedne informacije za optimizaciju i poboljšanje procesa.

Na temelju ovog pristupa, cilj je pružiti uvid u metode simulacije koje mogu pomoći u optimizaciji resursa i procesa, te kako njihova primjena može donijeti stvarne prednosti u upravljanju složenim sustavima. Ovaj rad će detaljno obraditi metode i rezultate simulacije, te raspraviti kako se oni mogu primijeniti u stvarnim situacijama.

2. OSNOVNE TEORIJE

2.1. Teorija redova čekanja

Teoriju čekanja i cijelo područje inženjeringa telefonskog prometa izumio je Agner Krarup Erlang, danski matematičar, statističar i inženjer. Početkom 20. stoljeća, Erlang je bio na čelu tehničkog laboratorija u Copenhagen Telephone Co. Njegove temeljite studije o vremenu čekanja u automatiziranim telefonskim sustavima i njegovi prijedlozi za unapređenje efikasnosti mreža široko su prihvaćeni od strane telekomunikacijskih kompanija. Teorija čekanja je grana matematike koja istražuje formiranje, funkcioniranje i uzroke problema u redovima. Ona analizira svaki aspekt čekanja u redovima, uključujući dolazak entiteta, proces pružanja usluge, broj poslužitelja, kapacitet sustava i broj korisnika, bilo da se radi o osobama, podacima, vozilima ili nečem drugom. Primjene teorije čekanja u stvarnom svijetu obuhvaćaju različite sektore poslovanja. Njene spoznaje mogu biti korisne za poboljšanje brzine pružanja usluga korisnicima, povećanje protoka prometa, optimizaciju procesa isporuke proizvoda iz skladišta ili dizajniranje efikasnijih podatkovnih mreža i pozivnih centara.¹

Littleov teorem opisuje vezu između brzine priljeva entiteta (kao što su kupci ili poslovi), vremena ciklusa i broja entiteta koji se trenutno nalaze u sustavu. Ovaj teorem tvrdi da se očekivani broj entiteta (označen s "N") u stabilnom stanju može izračunati pomoću sljedeće jednadžbe:

$$L = \lambda T$$

U ovoj jednadžbi, " λ " predstavlja prosječnu brzinu priljeva entiteta (kupaca ili poslova), dok "T" predstavlja prosječno vrijeme koje entitet provodi u sustavu.

Rješenja za stabilno stanje, prijelazno stanje i distribuciju vremena zauzetosti sustava proučavaju se za različite discipline i dobivaju se rješenja za stacionarno stanje u drugoj

¹ Queuing Theory Definition, Elements, and Example, <https://www.investopedia.com/terms/q/queuing-theory.asp>, (15.7.2023)

disciplini. Vjerojatnosti stacionarnog stanja sustava određuju se eksplicitno primjenom iterativnih metoda, a također se razmatraju korisne mjere učinkovitosti sustava.²

M/M/1 model je matematički okvir koji se primjenjuje za proučavanje i analizu redova čekanja ili sustava u kojem dolazi niz entiteta (kao što su kupci, poslovi itd.) koji traže uslugu od jednog jedinog poslužitelja. U ovom modelu koriste se sljedeći simboli:

"M" predstavlja Poissonov proces dolaska entiteta, što implicira da dolasci entiteta su nasumični i nepravilni.

"M" predstavlja eksponencijalno distribuirano vrijeme usluge, što znači da trajanje svake pojedinačne usluge također slijedi nasumičnu i eksponencijalnu raspodjelu.

"1" ukazuje na prisutnost samo jednog poslužitelja u sustavu koji pruža usluge entitetima.

Osnovni cilj M/M/1 modela jest analizirati i predviđati performanse sustava, kao što su prosječno vrijeme čekanja entiteta u redu, broj entiteta u redu i slične metrike. Ovaj model često se primjenjuje u situacijama poput telefonskih pozivnih centara, web stranica s korisničkom podrškom i drugih sustava s redovima čekanja.

Dakle, M/M/1 model služi za matematičku analizu redova čekanja u sustavima gdje postoji samo jedan poslužitelj, a dolasci entiteta i trajanje usluge imaju određene statističke osobine (Poissonov proces dolaska i eksponencijalno vrijeme usluge).

2.2. Koncept simulacije

Simulacija je izuzetno moćan alat za rješavanje problema i pronalaženje najboljih rješenja za izazove koji se javljaju u stvarnom svijetu. Ovaj alat omogućuje stvaranje modela stvarnog svijeta putem računalnih programa. Simulacijom je moguće analizirati potencijalne rizike, smanjiti troškove i čak predviđati događaje, čak i kada imate nepotpune informacije ili podatke. Korištenjem analitičke ili računalne simulacije, korisnicima se pruža mogućnost predviđanja performansi sustava i eksperimentiranja s različitim scenarijima kako bi se pronašlo optimalno rješenje.³

² Performance Analysis of M/M/1 Queuing Model, <https://turcomat.org/index.php/turkbilmat/article/view/11792>, (15.7.2023)

³ Queuing System Analysis A case study, https://bibliotecadigital.ipb.pt/bitstream/10198/22878/1/Ferreira_Janaina.pdf, (16.7.2023)

U simulaciji, stvaranje entiteta, njihov dolazak i odlazak su ključni elementi koji se koriste za oblikovanje ponašanja različitih entiteta (objekata, događaja, korisnika itd.) unutar simuliranog okruženja. Stvaranje entiteta (generiranje entiteta): Ovaj aspekt simulacije odnosi se na proces stvaranja novih entiteta unutar modela. Entiteti su predstavljeni kao objekti ili entiteti koji će sudjelovati u simuliranim događajima ili procesima. Na primjer, u simulaciji trgovine, stvaranje entiteta može predstavljati dolazak novih kupaca u trgovinu. Dolazak (arrival): Pojam dolaska odnosi se na trenutak kada novi entitet ulazi ili "dolazi" u simulirani sustav. Dolazak entiteta može biti slučajan i slijediti određeni raspored (na primjer, Poissonov proces dolaska), ili ga može kontrolirati prema specificiranim pravilima ili uvjetima definiranim u simulaciji. Kao primjer, u simulaciji prometnog križanja, dolazak bi se mogao odnositi na trenutak kada vozilo pristupa križanju. Odlazak (departure): Odlazak predstavlja trenutak kada entitet napušta simulirani sustav. Odlazak se obično događa nakon završetka određenog procesa ili usluge unutar simulacije. Na primjer, u simulaciji telefonskog pozivnog centra, odlazak bi mogao označavati trenutak kada korisnik završava svoj poziv i napušta pozivni centar. Ovisno o specifičnostima simulacije, stvaranje entiteta, dolazak i odlazak mogu se detaljnije prilagoditi kako bi se što preciznije modeliralo stvarno ponašanje sustava ili procesa koji se analizira. Ovi koncepti omogućuju simulacijskom modelu da simulira i analizira ponašanje entiteta kroz vremenski tijek, što pomaže u razumijevanju i optimizaciji stvarnih situacija i procesa.⁴

2.3. Prednosti Korištenja R Programskog Jezika i "Simmer" Paketa

R je izuzetno popularan programski jezik za statističko modeliranje i analizu podataka. Kao i svi drugi programski jezici, R ima svoje prednosti i nedostatke. Važno je napomenuti da je R dinamički jezik koji se neprestano razvija, što znači da će mnogi nedostaci s vremenom nestajati kroz buduća ažuriranja R-a. Ovo su neke od prednosti R programskog jezika:

- Otvoreni kod - Otvoreni jezik je onaj na kojem možemo raditi bez potrebe za licencama ili plaćanjem naknada. R je primjer jezika otvorenog koda, što znači da možemo doprinijeti njegovom razvoju optimiziranjem paketa, razvijanjem novih funkcionalnosti i rješavanjem problema bez ograničenja.

⁴ Queuing Theory, <https://www.wallstreetmojo.com/queuing-theory/>, (20.7.2023)

- Neovisan o platformi - R je jezik koji ne ovisi o operativnom sustavu, što znači da se njegov kod može izvoditi na različitim platformama. Omogućuje programerima da razvijaju softver koji će funkcionirati na različitim operativnim sustavima, a podržava rad na Windowsima, Linuxu i Macu.
- Strojno učenje - R pruža mogućnost izvođenja raznih operacija strojnog učenja kao što su klasifikacija i regresija, te nudi brojne pakete i značajke za razvoj umjetnih neuronskih mreža. Zbog tih mogućnosti, R je odabir mnogih vodećih stručnjaka za analizu podataka.
- Analiza neurednih podataka - R omogućuje analizu i obradu neurednih podataka. Koristeći pakete poput dplyr i readr, R može transformirati neuredne podatke u strukturirani oblik.
- Kvalitetno crtanje i grafovi - R pojednostavljuje izradu kvalitetnih grafova. Biblioteke kao što su ggplot2 i plotly omogućuju kreiranje vizualno privlačnih i estetskih grafova koji izdvajaju R od drugih programskih jezika.
- Velik broj paketa - R ima bogatu ponudu paketa. Repozitorij CRAN sadrži više od 10 000 paketa koji neprestano rastu, pružajući širok spektar alata za podatkovnu znanost i strojno učenje.
- Statističke mogućnosti - R je najpoznatiji kao jezik statistike te se često koristi za razvoj statističkih alata. To je ključni razlog zašto je R preferiran u usporedbi s drugim jezicima za izradu statističkih aplikacija.⁵

⁵ R Advantages and Disadvantages, <https://www.javatpoint.com/r-advantages-and-disadvantages>, (22.7.2023)

3. METODOLOGIJA

U nastavku su detaljno opisani svi ključni dijelovi koda potrebni za postavljanje i provođenje simulacije uz korištenje R simmer paketa. Obradeni su postupci kreiranja trajektorija, upravljanja resursima, definiranja dolazaka entiteta, kao i prikupljanje i analiza podataka, s ciljem osiguravanja jasne i precizne strukture simulacijskog modela.

Kreiranje trajektorija

Osnova svake simulacije je definiranje trajektorija (engl. *trajectories*) koje predstavljaju slijed aktivnosti kroz koje entiteti prolaze tijekom simulacije. Funkcija `trajectory()` inicira praznu trajektoriju na koju se naknadno dodaju aktivnosti pomoću metoda kao što su `seize()`, `timeout()` i `release()`.

```
traj <- trajectory("Osnovna trajektorija")
```

Upravljanje resursima

Resursi predstavljaju ključni dio simulacijskog modela i mogu uključivati ljudske resurse, opremu ili prostorije. Metode poput `seize()` i `release()` koriste se za alokaciju i oslobađanje resursa, dok `set_capacity()` i `set_queue_size()` definiraju kapacitet resursa i veličinu reda čekanja.

```
add_resource("operater", capacity = 1)
```

Kašnjenje i vrijeme obrade

Metoda `timeout()` koristi se za simulaciju kašnjenja ili čekanja u procesima. Vrijeme čekanja može biti fiksno ili ovisiti o atributima entiteta, a koristi se kako bi se modeliralo vrijeme obrade, kašnjenja i pauze unutar simulacije.

```
traj <- traj %>%  
  seize("operater") %>%  
  timeout(5) %>%  
  release("operater")
```

Grananje i paralelni procesi

Metoda `branch()` omogućuje uvjetno grananje unutar trajektorije, gdje se simulacijski put može razlikovati ovisno o specifičnim uvjetima. Ova metoda je korisna za modeliranje složenih sustava s više mogućih ishoda.

```
traj <- traj %>%  
  branch(  
    option = function() sample(1:2, 1),  
    continue = c(TRUE, FALSE),  
    traj1, traj2  
  )
```

Napuštanje sustava

Entiteti mogu napustiti simulaciju prije dovršetka trajektorije na temelju određenih uvjeta. Funkcije `renege_in()` i `renege_if()` omogućuju definiranje takvih uvjeta, primjerice ako entitet predugo čeka na resurs.

```
traj <- traj %>%
  renege_in(10) %>%
  seize("operater") %>%
  release("operater")
```

Prikupljanje podataka i praćenje

Simulacijsko okruženje stvoreno funkcijom `simmer()` omogućuje praćenje statističkih podataka, poput dolazaka entiteta, zauzetosti resursa i vremena obrade. Praćenje se izvodi pomoću funkcija kao što su `monitor()` i `get_mon_arrivals()`, što omogućuje kasniju analizu performansi sustava.

```
env <- simmer("Simulacija") %>%
  add_generator("entiteti", traj, function() rexp(1, 0.1)) %>%
  add_resource("operater", 1)

env %>%
  run(until = 100) %>%
  get_mon_arrivals()
```


Pokretanje simulacije

Nakon postavljanja trajektorija i resursa, simulacija se pokreće pomoću funkcije `run()`, koja omogućuje izvođenje simulacije u unaprijed definiranim vremenskim granicama.

Eksperimenti se provode mijenjanjem parametara resursa i aktivnosti kako bi se procijenili različiti scenariji.

```
env %>% run(until = 100)
```

U ovoj metodologiji objasnili smo sve bitne korake za izradu simulacije s R `simmer` paketom. Pokazali smo kako kreirati trajektorije, upravljati resursima i pratiti podatke, čime smo postavili temelje za jasnu i učinkovitu simulaciju.⁶

⁶ Function reference, <https://r-simmer.org/reference/>, (10.08.2024)

4. STUDIJA SLUČAJA

Ova simulacija koristi R *simmer* paket i temelji se na stvarnom primjeru iz mog okruženja, gdje hospicij ima samo jednu medicinsku sestru i jednog doktora. Ova situacija stvara značajne izazove u procesu prijema i pregleda pacijenata, jer s toliko malo osoblja cijeli sustav postaje prilično spor. Razmišljajući o ovome, postao sam znatiželjan koliko takva organizacija utječe na kvalitetu usluge i brzinu kojom pacijenti prolaze kroz sustav. Također, želim razumjeti gdje se pojavljuju potencijalna uska grla.

Simulacija je osmišljena da odražava stvarne postupke prijema pacijenata u ovom hospiciju. Dakle, prvo, medicinska sestra registrira pacijente, a zatim doktor obavlja pregled. U stvarnosti, to znači da svaki pacijent mora čekati na oboje, što može uzrokovati duže redove i čekanja. Cilj ove studije je vidjeti kako se sve to odvija i analizirati koliko se stanje može poboljšati.

Očekujem da će rezultati pokazati koliko su ograničenja u broju zaposlenih problematična i možda ćemo čak otkriti gdje se točno stvaraju najčešće gužve. Planiram provesti nekoliko simulacija kako bih isprobao različite scenarije, poput dodavanja dodatnog osoblja ili smanjenja vremena čekanja, kako bih vidio što bi moglo pomoći u rješavanju tih problema. Nadam se da ću tako doći do rješenja koja mogu pomoći hospiciju u optimizaciji svog rada i poboljšanju iskustva pacijenata.

5. RAZRADA I ANALIZA SIMULACIJSKOG KODA

U sljedećim primjerima prikazat ćemo ključne dijelove programskog koda uz pomoć vizualnih prikaza.

5.1. Korišteni paketi i definiranje parametara

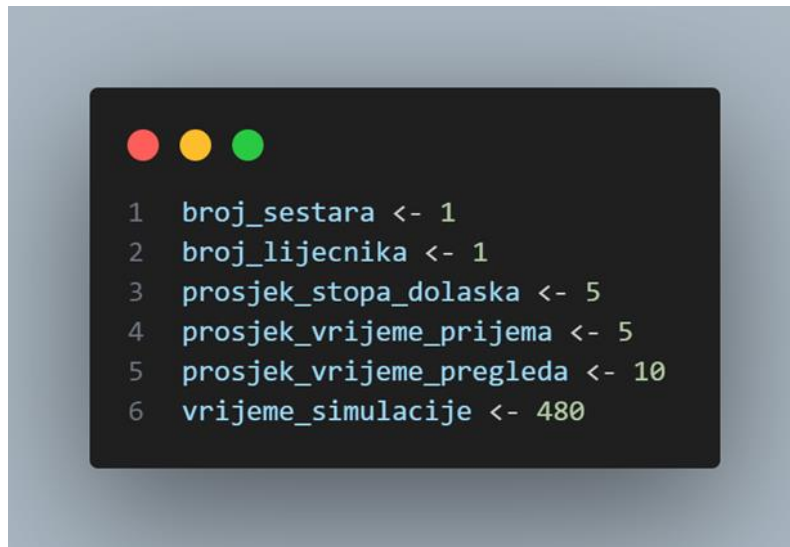
Za ovu simulaciju koriste se paketi `simmer`, `ggplot2` i `dplyr`. Ovi paketi omogućuju simulaciju procesa, analizu podataka i vizualizaciju rezultata.

A screenshot of a terminal window with a dark background and light-colored text. At the top left, there are three colored circles: red, yellow, and green. Below them, three lines of code are displayed, each preceded by a line number. The code is: 1 library(simmer), 2 library(ggplot2), and 3 library(dplyr).

```
1 library(simmer)
2 library(ggplot2)
3 library(dplyr)
```

Slika 1: Lista korištenih paketa (Izvor: Izradio autor)

Definirani su sljedeći parametri za simulaciju:

A screenshot of a terminal window with a dark background and light text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code is as follows:

```
1 broj_sestara <- 1
2 broj_lijecnika <- 1
3 prosjek_stopa_dolaska <- 5
4 prosjek_vrijeme_prijema <- 5
5 prosjek_vrijeme_pregleda <- 10
6 vrijeme_simulacije <- 480
```

Slika 2: Parametri (Izvor: Izradio autor)

Broj sestara: 1

Broj liječnika: 1

Prosječna stopa dolaska pacijenata: 1 pacijent svakih 5 minuta

Prosječno vrijeme prijema pacijenta: 5 minuta

Prosječno vrijeme pregleda pacijenta: 10 minuta

Ukupno vrijeme simulacije: 480 minuta (8 sati rada hospicija)

Ovi parametri odražavaju realne radne uvjete gdje jedan liječnik i jedna medicinska sestra brinu o pacijentima tijekom radnog dana. Simulacija pokriva 8 sati rada kako bi se dobio uvid u dnevne varijacije u redovima čekanja i korištenju resursa.

5.2. Definiranje putanje pacijenta

Putanja pacijenta simulira se pomoću funkcije *trajectory*, koja opisuje proces kroz koji pacijent prolazi od prijema do pregleda.

```
1 putanja_pacijenta <- trajectory("Putanja pacijenta") %>%
2   seize("sestra", 1) %>%
3   timeout(function() rnorm(1, prosjek_vrijeme_prijema, 0.5)) %>%
4   release("sestra", 1) %>%
5   seize("lijecnik", 1) %>%
6   timeout(function() rnorm(1, prosjek_vrijeme_pregleda, 1)) %>%
7   release("lijecnik", 1)
```

Slika 3: Putanja pacijenta kroz sustav (Izvor: Izradio autor)

Ovaj dio koda simulira tok pacijenta kroz sustav. Pacijent prvo čeka prijem kod medicinske sestre, gdje provodi vrijeme prijave, a zatim čeka pregled kod liječnika. Vrijeme prijema i pregleda nasumično se generira pomoću normalne distribucije, čime se simulira stvarna varijabilnost u trajanju tih procesa.

5.3. Stvaranje simulacijskog okruženja

Nakon definiranja putanje pacijenta, potrebno je stvoriti simulacijsko okruženje koristeći *simmer*. Ovdje definiramo resurse i generator pacijenata koji simulira njihov dolazak.

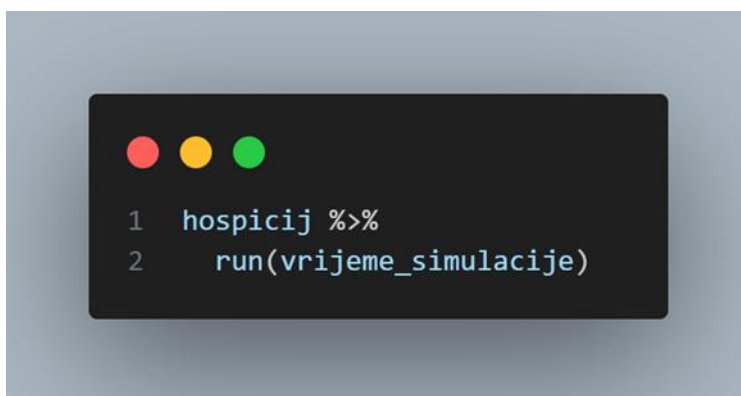
```
1 hospicij <- simmer("Hospicij") %>%
2   add_resource("sestra", broj_sestara) %>%
3   add_resource("lijecnik", broj_lijecnika) %>%
4   add_generator("Pacijent", putanja_pacijenta, function() rexp(1, 1 / prosjek_stopa_dolaska))
```

Slika 4: Simulacijsko okruženje (Izvor: Izradio autor)

U ovom koraku dodajemo resurse, kao što su medicinska sestra i liječnik, te definiramo kako pacijenti dolaze u sustav. *rexp* funkcija koristi se za simuliranje dolaska pacijenata u vremenskim intervalima na temelju eksponencijalne distribucije.

5.4. Pokretanje simulacije i dohvaćanje podataka

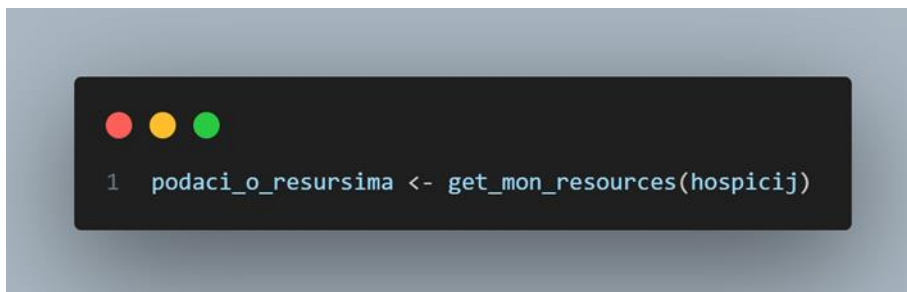
Simulacija se pokreće pomoću funkcije *run*, pri čemu se prikupljaju podaci o korištenju resursa i redovima čekanja.

A screenshot of a terminal window with a dark background and light text. At the top left, there are three colored circles: red, yellow, and green. Below them, two lines of R code are displayed:

```
1 hospicij %>%  
2   run(vrijeme_simulacije)
```

Slika 5: Pokretanje simulacije (Izvor: Izradio autor)

Nakon završetka simulacije za period od 480 minuta, podaci o resursima kao što su medicinske sestre i liječnici te podaci o duljinama redova čekanja dohvaćaju se pomoću funkcije *get_mon_resources*.

A screenshot of a terminal window with a dark background and light text. At the top left, there are three colored circles: red, yellow, and green. Below them, one line of R code is displayed:

```
1 podaci_o_resursima <- get_mon_resources(hospicij)
```

Slika 6: Dohvaćanje podataka (Izvor: Izradio autor)

5.5. Analiza redova čekanja

Analiza se fokusira na duljine redova čekanja kod resursa (sestre i liječnici). Podaci o redovima filtriraju se kako bi se prikazali samo slučajevi kada je duljina reda veća od nule.

A screenshot of a terminal window with a dark background and light text. At the top left, there are three colored circles: red, yellow, and green. Below them, there are four lines of R code, each preceded by a line number from 1 to 4. The code uses the dplyr package functions: select, rename, and filter. The code filters rows where the 'queue' variable is greater than zero.

```
1 podaci_o_redu <- podaci_o_resursima %>%  
2   select(time, resource, queue) %>%  
3   rename(duzina_reda = queue) %>%  
4   filter(duzina_reda > 0)
```

Slika 7: Filtriranje podataka (Izvor: Izradio autor)

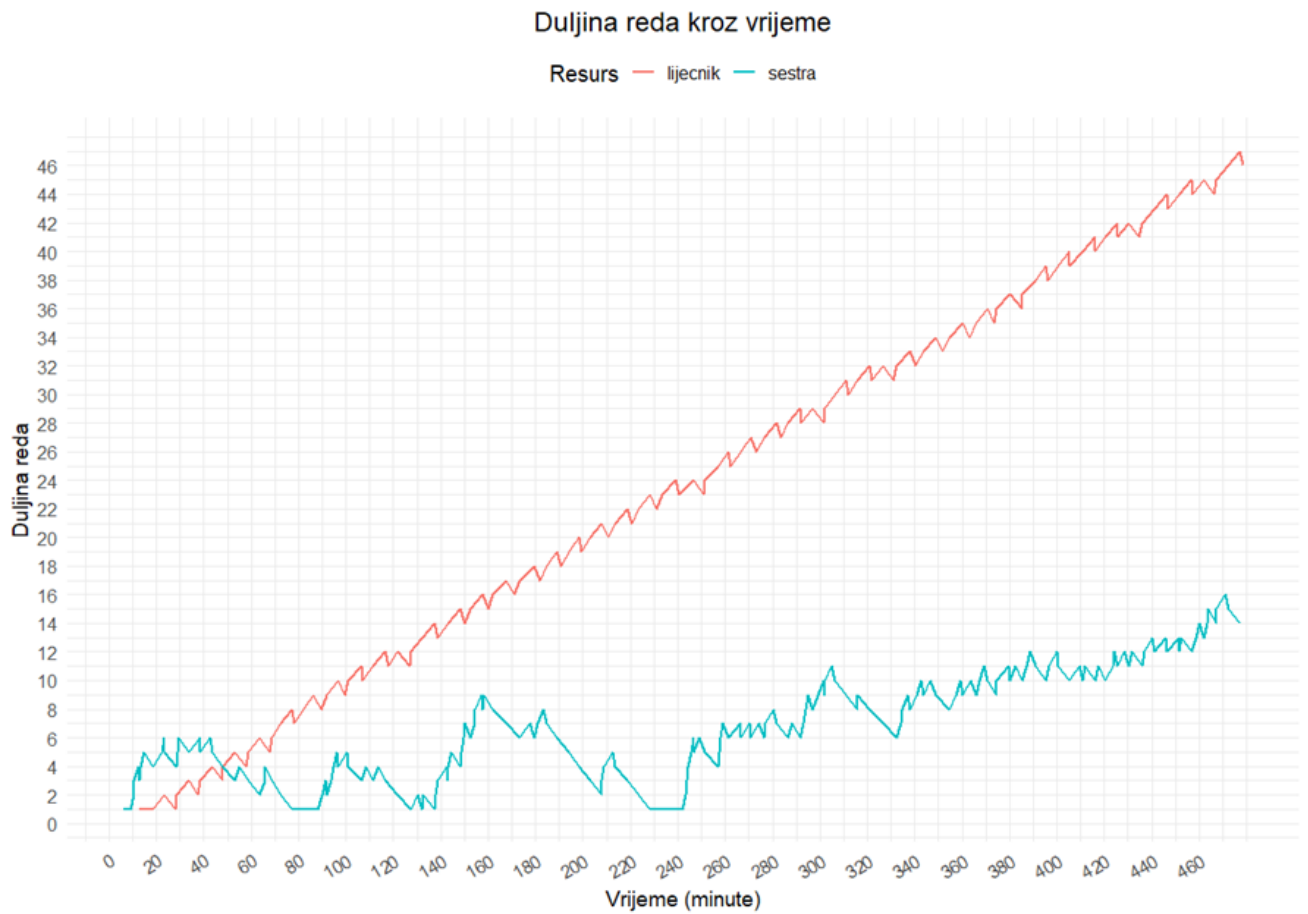
5.6. Vizualizacija rezultata

Rezultati simulacije prikazuju se grafički pomoću ggplot2 paketa, s fokusom na duljinu redova čekanja kroz vrijeme.

```
1 x_intervali <- seq(0, max(podaci_o_redu$time, na.rm = TRUE), by = 20)
2 y_intervali <- seq(0, max(podaci_o_redu$duzina_reda, na.rm = TRUE), by = 2)
3
4
5 ggplot(podaci_o_redu, aes(x = time, y = duzina_reda, color = resource)) +
6   geom_line(size = 1) +
7   labs(
8     title = "Duljina reda kroz vrijeme",
9     x = "Vrijeme (minute)",
10    y = "Duljina reda",
11    color = "Resurs"
12  ) +
13  scale_x_continuous(breaks = x_intervali, labels = scales::label_number()) +
14  scale_y_continuous(breaks = y_intervali, labels = scales::label_number()) +
15  theme_minimal(base_size = 15) +
16  theme(
17    plot.title = element_text(hjust = 0.5),
18    legend.position = "top",
19    axis.text.x = element_text(size = 12, angle = 30, hjust = 1),
20    axis.text.y = element_text(size = 12),
21    axis.title.x = element_text(size = 14),
22    axis.title.y = element_text(size = 14)
23  )
```

Slika 8: Kod za vizualizaciju (Izvor: Izradio autor)

Pomoću prethodnog koda dobivamo ovakav graf:



Slika 9: Graf 1 (Izvor: Izradio autor)

Ovaj graf prikazuje kako se duljina redova kod resursa mijenja kroz vrijeme, omogućujući identifikaciju razdoblja s najviše pacijenata koji čekaju na prijem ili pregled. Vizualizacija pruža vrijedan uvid u efikasnost raspodjele resursa i može pomoći u donošenju odluka o optimizaciji rada hospicija.

5.7. Rezultati Simulacije

Nakon pokretanja simulacije s definiranim parametrima, dobili smo sljedeći rezultat koji prikazuje stanje resursa na kraju 8-satnog radnog vremena hospicija:



```
1 simmer environment: Hospicij | now: 480 | next: 482.403040852701
2 { Monitor: in memory }
3 { Resource: sestra | monitored: TRUE | server status: 1(1) | queue status: 14(Inf) }
4 { Resource: liječnik | monitored: TRUE | server status: 1(1) | queue status: 46(Inf) }
5 { Source: Pacijent | monitored: 1 | n_generated: 109 }
```

Slika 10: Stanje resursa (Izvor: Izradio autor)

Trenutno vrijeme simulacije (now): Vrijeme simulacije je završilo nakon 480 minuta, što odgovara 8 sati, koliko traje radno vrijeme hospicija. Sljedeći događaj (next): Sljedeći događaj bi se dogodio u 482,4 minute, što znači da bi se novi pacijent pojavio neposredno nakon završetka simulacije.

Resurs "sestra": Server status: Sestra je trenutno zauzeta, što znači da je 1 od 1 sestre u tom trenutku angažirana.

Queue status (red čekanja): Trenutno ima 14 pacijenata koji čekaju na pregled kod sestre.

Resurs "liječnik": Server status: Liječnik je također potpuno angažiran, jer je 1 od 1 liječnika u tom trenutku zauzet.

Queue status (red čekanja): Pred liječnikom se nalazi 46 pacijenata u redu čekanja.

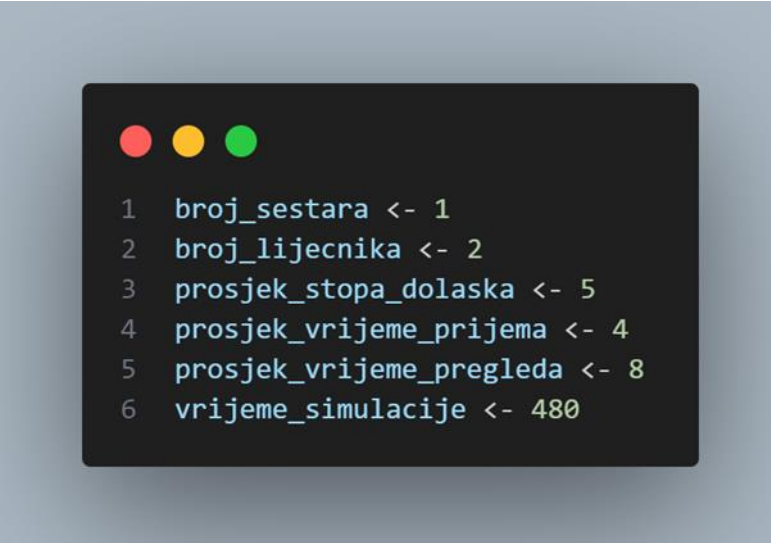
Generirani pacijenti: Ukupno je tijekom simulacije 109 pacijenata prošlo kroz sustav hospicija.

Rezultati simulacije jasno pokazuju opterećenje sestre i liječnika tijekom radnog vremena hospicija. Oba resursa su konstantno zauzeta, dok je red čekanja za oba resursa značajan, što ukazuje na nedovoljnu dostupnost resursa u odnosu na dolazak pacijenata. Red čekanja kod

liječnika je posebno izražen s obzirom na to da je pred njim 46 pacijenata na kraju simulacije, dok je pred sestrom 14 pacijenata. Ovi rezultati sugeriraju potrebu za povećanjem broja resursa ili smanjenjem vremena pregleda i prijema kako bi se smanjilo zagušenje u sustavu.

5.8. Novi parametri

Sada ćemo pokušati promijeniti nekoliko parametara kako bismo djelomično riješili problem čekanja u redu sa sljedećim kodom:

A screenshot of a terminal window with a dark background and light-colored text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code is as follows:

```
1 broj_sestara <- 1
2 broj_lijecnika <- 2
3 prosjek_stopa_dolaska <- 5
4 prosjek_vrijeme_prijema <- 4
5 prosjek_vrijeme_pregleda <- 8
6 vrijeme_simulacije <- 480
```

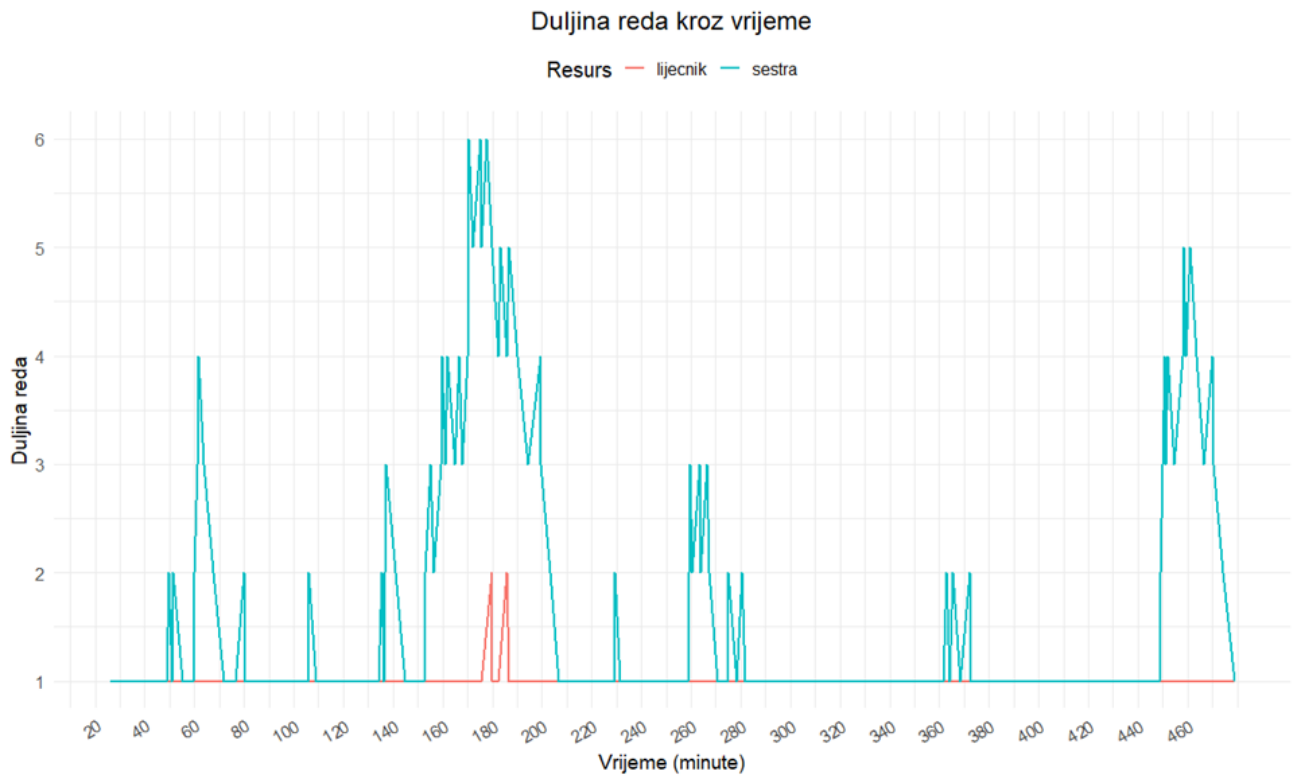
Slika 11: Ubacivanje novih parametara (Izvor: Izradio autor)

Broj liječnika povećan je s 1 na 2, što omogućuje da dva pacijenta budu pregledana u isto vrijeme. Ova izmjena bi trebala smanjiti red čekanja kod liječnika, ali povećati pritisak na medicinsku sestru koja mora obaviti prijem pacijenata.

Vrijeme prijema kod medicinske sestre je smanjeno s 5 na 4 minute, kako bi se ubrzao prijem pacijenata. Ovo je pokušaj da se smanji red čekanja kod sestara.

Vrijeme pregleda kod liječnika je smanjeno s 10 na 8 minuta. Time liječnici mogu brže obavljati preglede, čime se smanjuje potencijalni red čekanja kod liječnika.

Dobivamo sljedeći graf:



Slika 12: Graf 2 (Izvor: Izradio autor)

Rezultati simulacije ukazuju na pozitivan utjecaj uvođenja dodatnog liječnika i smanjenja vremena za prijem i pregled pacijenata:

```
1 simmer environment: Hospicij | now: 480 | next: 483.159388705805
2 { Monitor: in memory }
3 { Resource: sestra | monitored: TRUE | server status: 1(1) | queue status: 1(Inf) }
4 { Resource: lijecnik | monitored: TRUE | server status: 2(2) | queue status: 0(Inf) }
5 { Source: Pacijent | monitored: 1 | n_generated: 93 }
```

Slika 13: Novo stanje resursa (Izvor: Izradio autor)

Ukupno generirani pacijenti: Tokom simulacije je generirano 93 pacijenta, što odgovara predviđenoj stopi dolaska (jedan pacijent svake 5 minute).

Status resursa:

Sestra: Na kraju simulacije, sestra je u potpunosti angažirana (1 od 1 raspoložive sestre je zauzeta), dok u redu čeka samo jedan pacijent. Ovo sugerira da je skraćenje vremena prijema donekle smanjilo red, iako još uvijek postoji minimalno čekanje. Liječnici: Oba liječnika su zauzeta (2 od 2), bez pacijenata u redu. Ovaj rezultat jasno pokazuje da je dodavanje dodatnog liječnika uklonilo čekanje kod njih. Zaključak:

Skraćenjem vremena pregleda i dodavanjem drugog liječnika postigli smo ključni cilj – uklonili smo čekanje kod liječnika. Red kod medicinskih sestara je smanjen na jednog pacijenta, što predstavlja poboljšanje

6. ZAKLJUČAK

U ovom radu smo pokazali kako R programski jezik i paket „simmer“ mogu biti iznimno korisni za analizu i poboljšanje redova čekanja. Korištenjem simulacijskih metoda, uspjeli smo izraditi modele koji nam pomažu razumjeti dinamiku redova i prepoznati uska grla u različitim sustavima, uključujući primjer s ograničenim resursima. Analizom rezultata simulacija, uključujući vizualizaciju duljina redova čekanja, pokazali smo kako ograničenja u resursima utječu na učinkovitost i zadovoljstvo korisnika.

Simulacije su pokazale da manjak zaposlenika može značajno usporiti uslugu i smanjiti njezinu kvalitetu. Ovo je ključno za identifikaciju problema i pronalaženje rješenja za poboljšanje. U budućim istraživanjima, prilagodba simulacijskih parametara, poput povećanja broja zaposlenika ili promjene procesa rada, mogla bi pružiti dodatne uvide i poboljšanja.

Korištenje R-a i „simmer“ paketa omogućava detaljno modeliranje i testiranje različitih scenarija. Ovo može pomoći organizacijama da unaprijede svoje operativne procese i pružanje usluga. Ovaj rad pokazuje kako napredne statističke i simulacijske tehnike mogu biti primijenjene u stvarnim situacijama za poboljšanje redova čekanja i ukupne učinkovitosti sustava.

7. LITERATURA

- 1) Queuing Theory Definition, Elements, and Example, <https://www.investopedia.com/terms/q/queuing-theory.asp>, (15.7.2023)
- 2) Performance Analysis of M/M/1 Queuing Model, <https://turcomat.org/index.php/turkbilmata/article/view/11792>, (15.7.2023)
- 3) Queueing System Analysis A case study, https://bibliotecadigital.ipb.pt/bitstream/10198/22878/1/Ferreira_Janaina.pdf, (16.7.2023)
- 4) Queuing Theory, <https://www.wallstreetmojo.com/queuing-theory/>, (20.7.2023)
- 5) R Advantages and Disadvantages, <https://www.javatpoint.com/r-advantages-and-disadvantages>, (22.7.2023)
- 6) Function reference, <https://r-simmer.org/reference/>, (10.08.2024)

8. POPIS SLIKA

Slika 1: Lista korištenih paketa (Izvor: Izradio autor)	11
Slika 2: Parametri (Izvor: Izradio autor)	12
Slika 3: Putanja pacijenta kroz sustav (Izvor: Izradio autor)	13
Slika 4: Simulacijsko okruženje (Izvor: Izradio autor)	13
Slika 5: Pokretanje simulacije (Izvor: Izradio autor)	14
Slika 6: Dohvaćanje podataka (Izvor: Izradio autor)	14
Slika 7: Filtriranje podataka (Izvor: Izradio autor)	15
Slika 8: Kod za vizualizaciju (Izvor: Izradio autor)	16
Slika 9: Graf 1 (Izvor: Izradio autor)	17
Slika 10: Stanje resursa (Izvor: Izradio autor)	18
Slika 11: Ubacivanje novih parametara (Izvor: Izradio autor)	19
Slika 12: Graf 2 (Izvor: Izradio autor)	20
Slika 13: Novo stanje resursa (Izvor: Izradio autor)	20