

Analiza putanja zrakoplova korištenjem opažanja iz mreže OpenSky Network

Makar, Domagoj

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:547850>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-12-03**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

**ANALIZA PUTANJA ZRAKOPLOVA KORIŠTENJEM
OPAŽANJA IZ MREŽE OPENSKY NETWORK**

Rijeka, rujan 2022.

Domagoj Makar

9996000309

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

**ANALIZA PUTANJA ZRAKOPLOVA KORIŠTENJEM
OPAŽANJA IZ MREŽE OPENSKY NETWORK**

Mentor prof. dr. sc. Renato Filjar

Rijeka, rujan 2022.

Domagoj Makar

9996000309

IZJAVA

Izjavljujem da sam samostalno izradio diplomski rad uz znanja stečena tijekom studija te korištenjem navedene literature.

Rijeka, rujan 2022.

Domagoj Makar

SADRŽAJ

1. UVOD	1
2. PRETHODNA ISTRAŽIVANJA I TEHNOLOGIJE	2
2.1. OpenSky Network	2
2.2. Programsko okruženje za statističko računarstvo R	6
2.3. R programska knjižica openSkies.....	6
2.4. R programska knjižica osn	8
2.5. R programska knjižica openskyr	9
2.6. Trajr	11
2.8. Različite vrste hoda u trajektorijama	13
2.9. Indeks ravnosti (eng. straightness index)	13
2.10. Izgladivanje trajektorija (eng. smoothing trajectories)	14
2.11. Ponovno uzorkovanje putanja (eng. resampling trajectories).....	15
2.12. Analiza brzine	16
2.13. Indeks vijugavosti (Sinuosity index)	17
2.14. Maksimalni očekivani pomak trajektorije Emax (eng. maximum expected displacement of a trajectory)	18
2.15. Fraktalna dimenzija (eng. fractal dimension)	19
2.16. Promjena smjera kretanja (eng. directional change).....	20
2.17. Autokorelacija smjera (eng. directional autocorrelation)	21
2.18. Shiny	22
3. IZRADA RJEŠENJA.....	25
4. REZULTATI ISTRAŽIVANJA	29
5. ZAKLJUČAK	39
6. DODATAK.....	40
7. LITERATURA	46
8. SAŽETAK	48

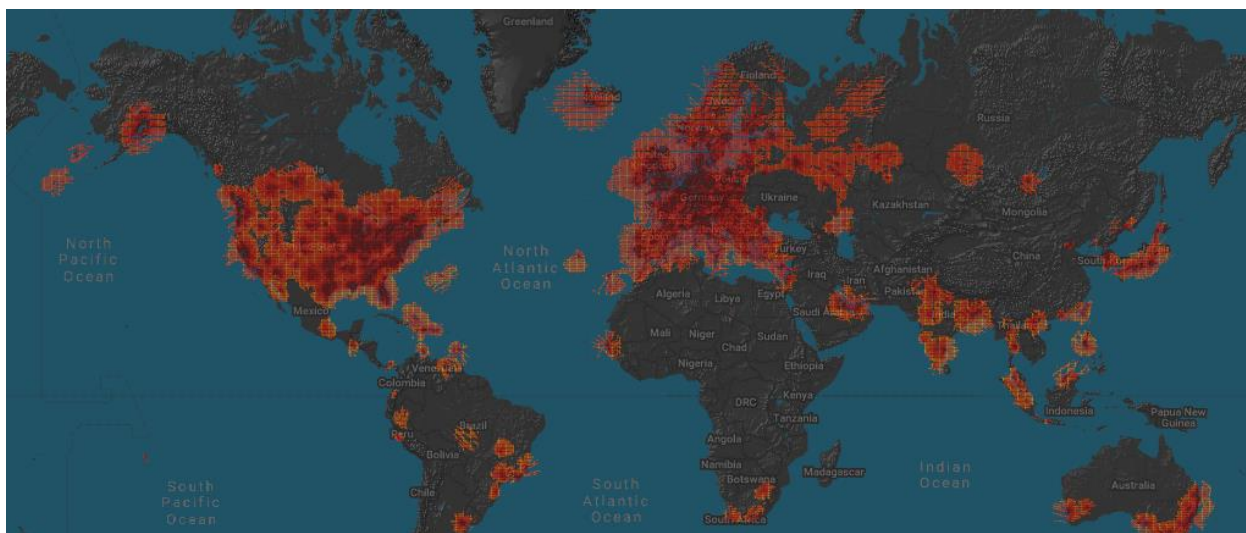
1. UVOD

Brojni sustavi izvještavanja o gibanju u prostoru (zrakoplovstvo, pomorstvo, osobna navigacija) kreiraju masivne skupove podataka koji predstavljaju podlogu za ciljanu analizu putanja, omogućujući spoznaje o uzrocima ponašanja i karakterističnim obilježjima putanja. Navedene spoznaje mogu biti korištene u strateškom planiranju, optimizaciji navigacijskih postupaka i upravljanja prometom, kao i poboljšanju sigurnosti i smanjenju štetnih učinaka na okoliš. Cilj ovog rada demonstracija je analize putanja zrakoplova korištenjem opažanja prikupljenih i spremljenih u OpenSky Network inicijativi putem razvoja okvira za analizu putanja zrakoplova u programskom okruženju R. Broj letova raste kroz godine, izuzev 2020. zbog pojave pandemije koronavirusa, tokom koje je zabilježen pad od 46.77% [1]. Između 2020. te 2021. godine zabilježen je rast od 35.36% (sa 16.4 na 22.2 milijuna letova). Zbog ovih indikacija naglog porasta otvara se mogućnost boljeg planiranja postojećih i novih letova analizom karakterističnih obilježja putanja. U drugom poglavlju, opisani su alati korišteni za izradu programskog rješenja: OpenSky Network baza podataka, programski jezik za statističko računarstvo R, programske knjižice za programski jezik R koje omogućavaju pristup OpenSky Network bazi podataka, Trajr, knjižica za praćenje putanja životinja te Shiny, paket za izradu grafičkih korisničkih sučelja za programski jezik R. Nakon opisa pojedinih alata, u trećem poglavlju opisano je njihovo korištenje za izradu rješenja. Dobiveno programsko rješenje testirano je na različitim letovima radi vrednovanja njegove korisnosti.

2. PRETHODNA ISTRAŽIVANJA I TEHNOLOGIJE

2.1. OpenSky Network

OpenSky Network neprofitna je mreža prijemnika nastala 2013, na inicijativu istraživača iz Švicarskog ureda za obranu, Sveučilišta Kaiserslautern te Sveučilišta u Oxfordu. Cilj OpenSky mreže prijemnika prikupljanje je podataka iz zračnog prometa u svrhu istraživanja. Način na koji se podatci prikupljaju jest pomoću tehnologije automatski ovisnih prijemnika za nadzor i emitiranje (eng. Automatic Dependent Surveillance-Broadcast receivers), kratica ADS-B. Zbog ogromne količine podataka koje se spremaju ADS-B prijemnicima, OpenSky mreža čuva neobrađene povijesne podatke samo u kraćim vremenskim intervalima. Kod naziva ADS-B prijamnici se smatraju automatskima jer se poruke periodički odašilju bez ikakve potrebe za vanjskim utjecajima te ovisnima, zato što u potpunosti ovise o podacima koji se odašilju iz navigacijskog sustava letjelice. ADS-B prijamnici koje OpenSky mreža koristi sponzorirani su i u vlasništvu krajnjih korisnika neovisnih o samoj OpenSky organizaciji, koji samovoljno pružaju podatke sa svojih prijemnika. OpenSky mreža dopušta potpun pristup svojim neobrađenim i obrađenim podacima sveučilišnim institucijama, istraživačima te neprofitnim udrugama. Za pristup svim podacima potrebno je izraditi profil te zatražiti dopuštenje od OpenSky mreže. Neki od setova podataka koje OpenSky mreža prikuplja te kojima daje pristup jesu: 24 satni podatci o vektorima stanja koji se izdaju svaki ponedjeljak, „OpenSky Raw Data“ unutar koje su neobrađeni podatci, „LocaRDS“ baza podataka koja sadrži podatke koji se koriste za treniranje lokalizacije, „Aircraft Metadata“ koja sadrži podatke o pojedinim letjelicama, te referentne setove podataka za hitne slučajeve. Pokrivenost OpenSky mreže najbolja je na području Europe i Sjeverne Amerike[2].



Slika 1. Pokrivenost OpenSky mreže

Letjelice koje nisko lete teže je nadgledati, pa je time pokrivenost mreže bolja, pogotovo u tom slučaju, ako se u blizini nalaze ADS-B prijemnici (na prethodnoj slici ta mjesta su označena crvenom bojom, tamnije crvena boja označava nižu nadmorsku visinu na kojoj je moguće pratiti letjelice). Dva bitna tipa podataka koje OpenSky baza podataka zaprima jesu FLARM (eng. flight alarm) podatci te Mode-S (eng. Mode Select) kodovi. FLARM podatci koriste se za izbjegavanje mogućih sudara letjelica, dok su Mode-S kodovi za nadzor i komunikacijske sustave letjelica za kontrolu leta. Razlika između rada FLARM i Mode-S sustava, jest taj što se kod FLARM sustava poruke izmjenjuju direktno između FLARM sustava obližnjih letjelica, dok se kod Mode-S sustava poruke izmjenjuju s radarskim stanicama koje se nalaze na površini zemlje.

Tablica 2.1. Broj podataka u OpenSky bazi

Broj skupljenih MODE-S poruka	22.000.000.000
Broj MODE-S odašiljača	10.400
Broj jedinstvenih zrakoplova	477.918
Broj skupljenih FLARM poruka	4.205.900.000
Broj MODE-S poruka po sekundi	34.730
Broj registriranih prijemnika	4.936

OpenSky bazi podataka može se pristupiti preko različitih sučelja za programiranje aplikacija (eng. Application programming interface, u daljnjem tekstu koristit će se kratica API), te pomoću baze povijesnih podataka (eng. Impala shell). Razlika između API-a i povijesne baze podataka jest u vremenu nastanka podataka. Kao što joj samo ime kaže, povijesna baza podataka sastoji se od starijih podataka, dok API pruža pristup trenutnim podacima u stvarnom vremenu. Trenutno podržana sučelja su REST API (eng. representational state transfer), Python API, Java API te Impala Shell. Pristup podacima u stvarnom vremenu ograničen je na 8 minuta dnevno za neregistrirane te 90 minuta dnevno za registrirane korisnike. Zahtjevi prema OpenSky bazi podataka predaju se ovisno o tome koja informacija je željena. U svrhu dobivanja vektora stanja za određene letjelice ili određeno vrijeme, potrebni su parametri s tablice (2.2.).

Tablica 2.2 Parametri zahtjeva koji se šalju bazi podataka

Svojstvo	Tip	Opis
time	integer	Vrijeme (eng. time) proteklo od epohe.
icao24	string	Jedinstvena 24 bitna adresa letjelice
lamin	float	Niža granica zemljopisne širine
lomin	float	Niža granica zemljopisne duljine
lamax	float	Viša granica zemljopisne širine
lomax	float	Viša granica zemljopisne duljine
extended	integer	Ispis kategorije letjelice

U tablici (2.2) stupac svojstvo označuje parametar pomoću kojeg zahtijevamo podatke od baze podataka. Stupac tip označava tip podataka kojim se bazi prenosi informacija koja filtrira rezultate. Integer predstavlja cijeli broj. Float predstavlja broj s decimalnom točkom. String tip podataka predstavlja niz slova. Vrijeme se ne prenosi u ljudski čitljivom obliku, već u broju sekunda koje su prošle od epohe. Epoha je referentno vrijeme u povijesti. U slučaju OpenSky baze podataka, epohom se smatra Unix epoha: 1. siječnja 1970. u 00:00:00 po univerzalnom koordiniranom vremenu (UTC). Nakon što se bazi podataka pošalje važeći zahtjev, baza podataka odgovara s JSON objektom opisanim u tablici (2.3.).

Tablica 2.3. struktura JSON objekta kojim baza podataka odgovara na zahtjev

Indeks	Svojstvo	Tip	Opis
0	icao24	string	Jedinstvena ICAO 24-bitna adresa u hex string reprezentaciji
1	callsign	string	Pozivni znak letjelice
2	origin_country	string	Ime države podrijetla
3	time_posiion	int	Unix vrijeme u sekundama za posljednju informaciju o poziciji letjelice
4	last_contact	int	Unix vrijeme koje označava kada je posljednja informacija primljena
5	longitude	float	Zemljopisna dužina u WGS-84 formatu
6	lattitude	float	Zemljopisna širina u WGS-84 formatu
7	baro_altitude	float	Barometarska visina u metrima
8	on_ground	boolean	Odgovara na pitanje je li informacija o poziciji primljena s tla
9	velocity	float	Brzina u metrima po sekundi
10	true_track	float	U stupnjevima, u pozitivnom smjeru
11	vertical_rate	float	Promjena visine u metrima po sekundi
12	sensors	int[]	Jedinstvena identifikacija prijemnika koji su prenijeli informaciju
13	geo_altitude	float	Visina u metrima
14	squawk	string	Identifikacija odašiljača, tzv. Squawk
15	spi	boolean	Ukazuje li status leta na neki od posebnih indikatora
16	position_source	int	Izvor pozicije
17	category	int	Kategorija letjelice

WGS-84 predstavlja svjetski geodetski sustav, trenutno standardiziran sustav koji se koristi u avijaciji. Barometarska visina određuje se mjerenjem razlike tlaka zraka na trenutnom položaju letjelice te tlaka zraka na razini mora.

2.2. Programsko okruženje za statističko računarstvo R

Programski jezik R programski je jezik otvorenog koda koji se najviše koristi za statističko računarstvo i grafiku. Razvijen je kao alternativa programskom jeziku S, no s naglaskom na kod otvorenog tipa. Programski jezik R je pod GNU licencom, što omogućava krajnjem korisniku potpunu kontrolu nad programskim jezikom. Posebnosti programskog jezika R u odnosu na klasičnije programske jezike (kao što je C, Java, C# i slično), jest način na koji se vrši manipulacija nad podacima. Većina funkcija napisanih u programskom jeziku R su vektorske. Ovo znači da se operacije ne izvode nad zasebnim jedinicama, kao što je jedan element unutar polja, već se operacije izvode nad cijelim poljem [3]. Podatkovni okvir (eng. data frame) također je bitan dio R programskog jezika. Podatkovni okvir struktura je slična matricama (dvodimenzionalnom polju), no unutar njega svaki od stupaca sadrži zaseban tip podataka kojemu se može pristupiti po imenu stupca. U daljnjem tekstu proći ćemo kroz knjižice bazirane na programskom jeziku R, koje omogućavaju lakši pristup OpenSky API-u ili OpenSky povijesnoj bazi podataka (ili oboje).

2.3. R programska knjižica openSkies

Programska knjižica openSkies jedina je R knjižica za pristup OpenSky API-u i OpenSky povijesnoj bazi podataka koja je objavljena na CRAN mreži [4]. Omogućava dobivanje specifičnih informacija o letu kao i vektore stanja pojedinih letova. Za programsku knjižicu openSkies napisan je referentni priručnik s primjerima i opisima pojedinih funkcija te vinjeta s kratkim opisom same knjižice i njenog principa rada. Testiranjem knjižice openSkies pronađene su određene limitiranosti same baze podataka. Dobivanje informacija o letovima iz određene zračne luke, koja je nama potrebno za analizu letova između dvije zračne luke ograničeno je na najviše tjedan dana. Pristup podacima leta putem openSkies knjižice može se vršiti na nekoliko različitih načina: prema letjelici, prema vremenskom intervalu, prema aerodromu (polaznom ili dolaznom, no ne oba istovremeno). Vektori stanja vraćaju se u obliku podatkovnih okvira koji sadrže informacije o poziciji letjelice, njenoj visini, državi u kojoj je registrirana i brzini u određenom momentu. Kao pokušaj standardizacije podatkovne strukture za analizu podataka letova, openSkies knjižica podatke vraća u posebnoj klasi nazvanoj „R6“, koja sadrži informacije o letjelici, letu, ruti, vektorima stanja te setu vektora stanja. Primjer zahtjeva prenesenog pomoću knjižice openSkies:

```

planes <- getSingleTimeStateVectors(minLatitude=45.8389, maxLatitude=47.8229,
                                   minLongitude=5.9962, maxLongitude=10.5226)
plotPlanes(planes)

getAirportDepartures(airport = "LEBL", startTime = "2021-10-12 05:00:00",
                    endTime = "2021-10-12 07:00:00", timeZone = "Europe/Madrid",
                    useImpalaShell = TRUE,
                    includeAirportsMetadata = TRUE)

getAircraftFlights("346190", startTime = "2019-07-26 00:00:00",
                  endTime = "2019-07-26 23:59:59", timeZone = "Europe/Madrid")

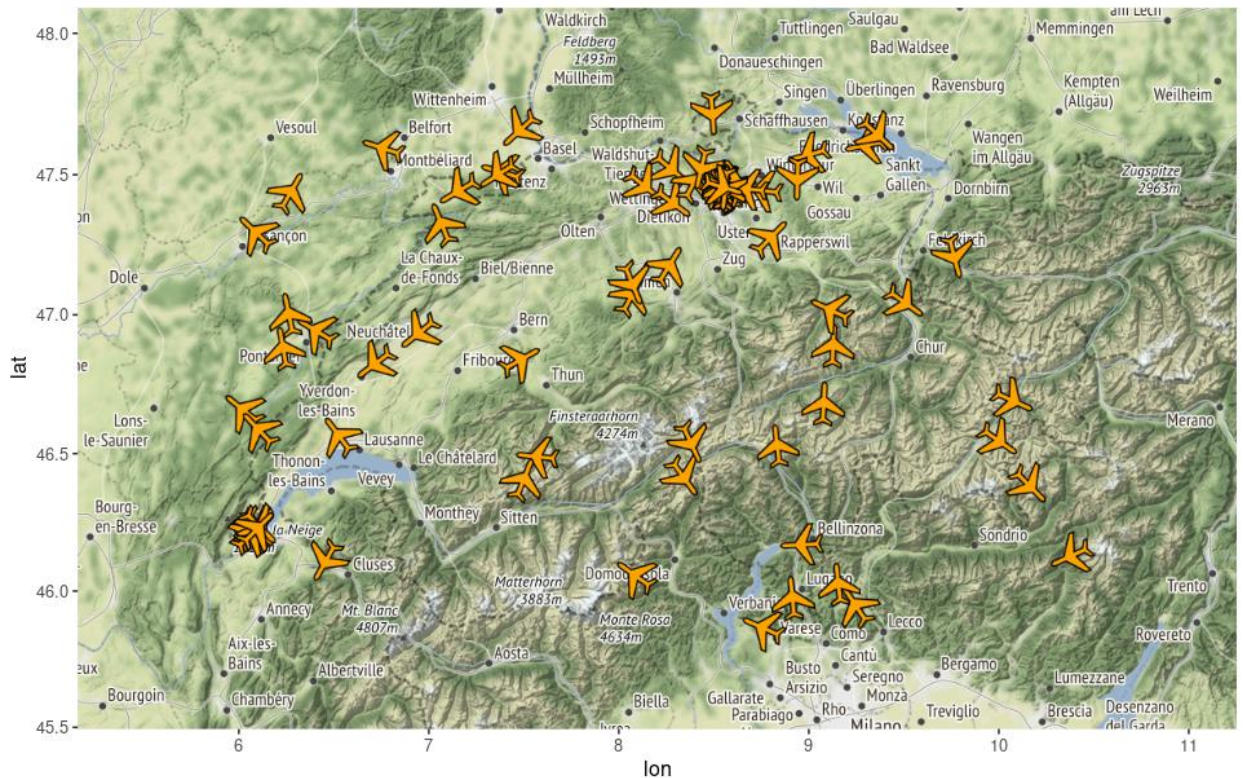
flights <- getIntervalFlights(startTime = "2019-11-16 09:00:00",
                              endTime = "2019-11-16 10:00:00",
                              timeZone = "Europe/London")

length(flights)

```

Slika 2. primjer dohvaćanja informacija pomoću programske knjižice openSkies

Na prethodnoj slici (*Slika 2.*) vidi se zahtjev za svim trenutnim vektorima stanja koji se nalaze od zemljopisne širine 45.8389 do 47.8229 te od zemljopisne duljine 5.9962 do 10.5526. Nakon što baza podataka vrati odgovor, taj odgovor se zatim ispisuje na mapi (*slika 3*). Osim ove funkcionalnosti, u isječku koda vidi se i način dobivanja informacija o uzletima s određene zračne luke pomoću funkcije „getAirportDepartures“ (LEBL je kratica za zračnu luku Josep Tarradellas Barcelona), između 5 i 7 sati na dan 12.10.2021. Funkcija „getAircraftFlights“ dohvaća vektore stanja letova određene letjelice, u slučaju slike 2, letjelice sa icao24 oznakom 346190. Funkcija „getIntervalFlights“ u određenom razdoblju vraća sve vektore letova.



Slika 3. ispis trenutnih vektora stanja na mapi.

Na slici 3 narančastim ikonama označene su sve letjelice koje se u momentu u kojemu je poslan zahtjev nalaze unutar dane zemljopisne širine i duljine. Osim samih prostornih lokacija, dobivene su i informacije o smjeru, brzini pojedinih letjelica, te nalaze li se one trenutno u zraku.

2.4. R programska knjižica osn

Programska knjižica osn je R knjižica koja pruža pristup OpenSky povijesnoj bazi podataka. Osn programska knjižica diskretizira dobivene podatke kako bi bili lakše obrađeni, npr. podatci o visini diskretizirani su na višekratnike od 1000 u stopama. Pomoću osn programske knjižice moguće je dobiti povijesne podatke o letovima, podatke o vektorima stanja letjelica koje slijeću, vektorima stanja letjelica koje uzlijeću s pojedinih zračnih luka ili slijeću u pojedine zračne luke te status pojedinih letjelica prema njihovom identitetu.

2.5. R programska knjižica openskyr

Programska knjižica openskyr je R knjižica koja pruža pristup podacima u stvarnom vremenu. Uz kod, pružena je i kratka vinjeta koja objašnjava pristup podacima, vektorima stanja trenutnih letjelica, pristup informacija o letovima koji su poletjeli iz pojedinih zračnih luka u danom vremenu, pristup informacijama o pojedinoj letjelici za određeno vrijeme te trajektorijama letjelice u danom vremenu.

```
data_airport_df <- get_airport_data(option="departures", airport = "LDZA",
                                   begin = 1656911452, end = 1657058400)

data_airport_df
nrow(data_airport_df[data_airport_df$estArrivalAirport == "EDDM", ])
data_airport_df[data_airport_df$estArrivalAirport == "EDDM", c(1,2,4)]

data_airport_df[data_airport_df$estArrivalAirport == "EDDM",
c(1,2,4)]$icao24[[1]]

data_specific_flight <- get_flights_data(icao24 = "501da9", begin = "1656885600",
end = "1656972000")

data_all_flight <- get_flights_data(begin = 1517227200, end = 1517230800)

data_flight_track <- get_track_data(icao24 = "494103", time = 1587126600)

state_vectors_one <- getIntervalStateVectors(aircraft = "501da9",
                                             startTime = "2022-07-04 07:10:52",
                                             endTime = "2022-07-04 08:05:56")
state_vectors_two <- getIntervalStateVectors(aircraft = "501da8",
                                             startTime = "2022-07-05 06:58:59",
                                             endTime = "2022-07-05 07:52:51")

svf1 <- state_vectors_one$split_into_flights()
svf2 <- state_vectors_two$split_into_flights()

path <- vector(mode = "list", 2)
for(i in 1:length(paths)){
  path[[i]] <- paths[[i]]$state_vectors
}

plotRoutes(path, pathColors = rainbow(2))
```

Slika 4. prikaz koda potrebnog za dohvaćanje dva vektora leta pravac Zagreb – Muenchen.

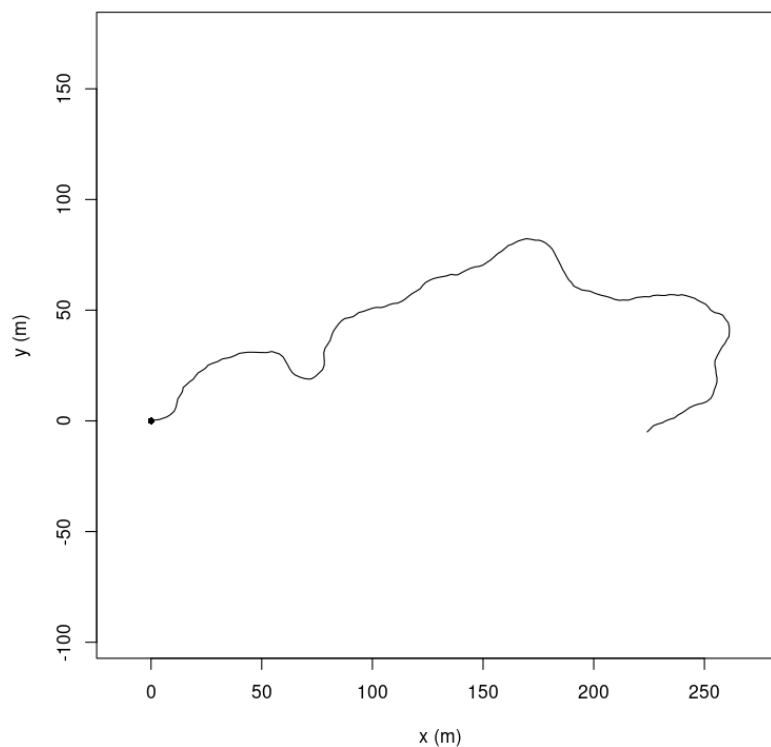
Kako bi se dobile vektorske informacije o letu između Zagreba i Muenchena, potrebno je prvo dohvatiti sve letove koji polijeću sa zračne luke Zagreb. Kratica LDZA označava zračnu luku

smjeru sjeveroistoka. Uzlijetanje i slijetanje zrakoplova, kao i položaj samih pista na zračnoj luci ovisi o ruži vjetrova. Kod uzleta letjelice uvijek idu u smjeru vjetra, dok kod slijetanja idu protiv vjetra. Ovo se čini kako bi se minimizirala količina goriva te duljina piste potrebnih za uzlijetanje i slijetanje zrakoplova.

2.6.Trajr

Trajr je ime knjižice za statističku analizu kretanja u dvodimenzionalnom prostoru. Trajr je nastao radi analiziranja putanja životinja. Trenutno je korišten u analizi migracija životinja, povezivanja njihovih kretanja s traženjem hrane te na pobližim analizama kretanja kao što je analiza sličnosti između kretanja između više različitih vrsta životinja (bihevioralna mimikrija) [5]. Analiza se vrši u odnosu na pružene trajektorije te njihova obilježja. Ovdje kao obilježja trajektorija smatra se koordinate u prostoru, kroz određeno vrijeme. Trajr knjižica je otvorenog tipa što omogućava krajnjem korisniku potpuni pristup izvornom kodu, njegovo izmjenjivanje te prilagodbu specifičnim slučajevima upotrebe. Knjižica Trajr paket je za R programski jezik. Objavljena je na CRAN mreži čime se olakšava integracija knjižice u R projekt činjenicom da je potrebna samo jedna R naredba kako bi se preuzela posljednja stabilna verzija Trajr knjižice. Bitan nedostatak Trajr knjižice za ovaj projekt jest to što se analiza putanja vrši u dvodimenzionalnom prostoru, dok se informacije iz OpenSky baze podataka odlikuju s tri dimenzije: položaj na zemljopisnoj širini, položaj na zemljopisnoj dužini, te barometarska visina. Kako bi se olakšalo testiranje Trajr knjižice, s njom dolaze funkcije namijenjene specifično izradi testnih podataka i simulacija pomoću kojih se mogu procijeniti pojedine funkcionalnosti.

```
trj <- TrajGenerate(200, random = TRUE, angularErrorSd = .25)
plot(trj, lwd = 1, lty = 1)
```



Slika 6. Trajr kod te generirana trajektorija pomoću Trajr knjižice

Generirana trajektorija prikazana je na koordinatnom sustavu. Na osima koordinatnog sustava y os označava udaljenost u metrima od početnog položaja u vertikalnom smjeru. x os označava udaljenost u metrima od početnog položaja u horizontalnom smjeru. Generirana putanja iskoristit će se za procjenu funkcionalnosti knjižice u daljnjem tekstu. Putanja je generirana pomoću funkcije „TrajGenerate“, u 200 koraka, nasumičnim hodom.

2.7. Alati Trajr knjižice

Trajr knjižica sastoji se od alata za stvaranje trajektorija iz postojećih podataka kao što su CSV datoteke, tablice ili video datoteke. Za trajektorije je moguće:

- Vršiti skaliranje, naglasak je na trajektorijama koje su dobivene iz video datoteka, odnosno, pretvorba iz piksela u jedinice koje se mogu prikazati u prostoru (centimetri, metri, kilometri i sl.)
- Iz video datoteka dobiti informacije o slikama u sekundi za pojedinu trajektoriju
- Izvršiti izgladivanje
- Izvršiti simulacije postojećih trajektorija unutar ograđenog prostora

- Detektirati zaustavljanje i započinjanje kretanja
- Izvršiti ponovno višestruko uzorkovanje
- Izvršiti rotaciju, translaciju, preokretanje
- Dobiti informacije o dimenzionalnim jedinicama korištenim u analizi trajektorija.
- Dobiti informaciju o udaljenosti početne i završne pozicije trajektorije
- Dobiti medijan brzine trajektorije
- Dobiti kutove skretanja unutar trajektorija, uključujući i srednji vektor skretanja

2.8. Različite vrste hoda u trajektorijama

Hod u trajektoriji klasificira se u dvije bitne skupine: nasumičan hod te usmjeren hod. Nasumičan hod način je kretanja u prostoru prema kojemu svaki korak (pozicija između dviju promatranja) ne može biti predviđen, dok usmjeren hod teži od neke početne točke A prema završnoj točki B.

2.9. Indeks ravnosti (eng. straightness index)

Indeks ravnosti računa se kao omjer između zračne distance dviju točaka te stvarnog prijeđenog puta. Indeks ravnosti ST računa se prema izrazu (2.1):

$$ST = \frac{DE}{L} \quad (2.1)$$

gdje je:

DE Euklidska udaljenost između početka i kraja puta,

L Stvarna duljina prijeđenog puta.

Vrijednosti indeksa ravnosti moraju biti između 0 i 1, gdje 1 označava savršeno poklapanje zračne distance i stvarnog prijeđenog puta (prijedeći put je ravna linija).

```
function (trj)
{
    TrajDistance(trj)/TrajLength(trj)
}
```

Slika 7. prikaz računanja indeksa ravnosti u Trajr knjižici

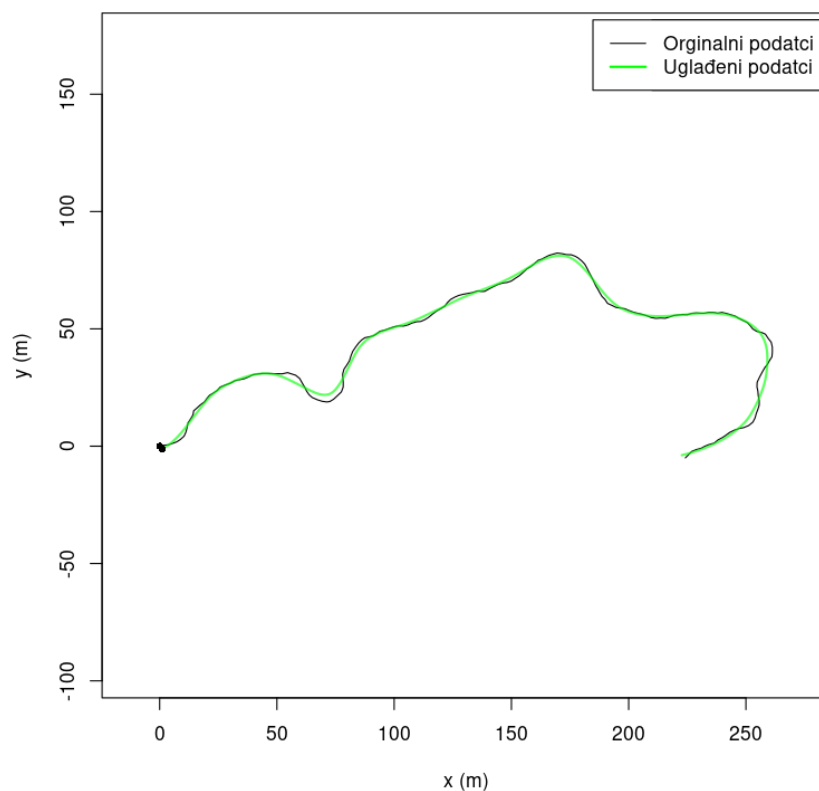
2.10. Izgladivanje trajektorija (eng. smoothing trajectories)

Cilj izgladivanja trajektorija jest smanjenje utjecaja šuma na rezultate analiza. Algoritam za izgladivanje trajektorija knjižice Trajr bazira se na Savitzky-Golay filteru. Savitzky-Golay filter vrši izgladivanje trajektorije bez da utječe na generalni ishod grafa. Najveće razlike kod primjene Savitzky-Golay filtera vide se na područjima trajektorija koje imaju nagle promjene u odnosu na susjedne trajektorije, ovaj filter takve trajektorije izgladuje, odnosno, smanjuje njihov utjecaj.

```
function (x, p = 3, n = p + 3 - p%2, m = 0, ts = 1)
{
    len = length(x)
    if (class(p) == "sgolayFilter" || (!is.null(dim(p)) && dim(p) >
        1)) {
        F = p
        n = nrow(F)
    }
    else F = sgolay(p, n, m, ts)
    k = floor(n/2)
    z = filter(F[k + 1, n:1], 1, x)
    c(F[1:k, ] %*% x[1:n], z[n:len], F[(k + 2):n, ] %*% x[(len -
        n + 1):len])
}
```

Slika 8. prikaz funkcije koja primjenjuje Savitzky-Golaj filter nad unesenim podacima

Savitzky-Golay filter radi na principu primjene polinoma niskog reda između dviju susjednih točaka[6]. Trajr knjižica omogućuje odabir reda filtra, te duljine filtra, koja mora biti neparna.

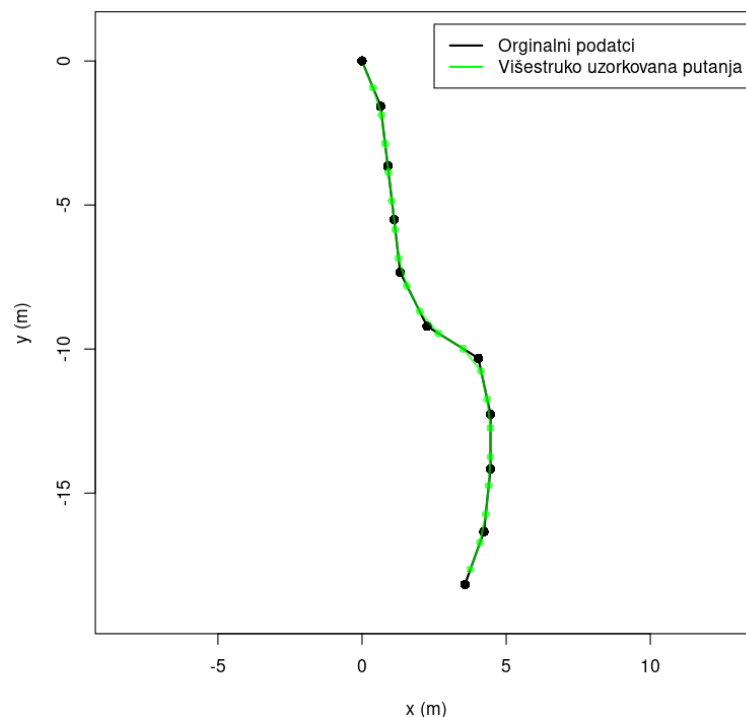


Slika 9. Prikaz izgladivanja trajektorije iz slike 6

Za izgladivanje trajektorije sa slike 6, korišten je Savitzky-Golay filter s polinomom 3. stupnja, na 31 točaka. Iz slike 9. vidi se kako Savitzky-Golay filter ublažuje velike promjene u putanji.

2.11. Ponovno uzorkovanje putanja (eng. resampling trajectories)

Cilj ponovnog uzorkovanja puta dobiti je više trajektorija u putanji nego što je dostupno iz izvornog seta podataka. Ponovno uzorkovanje puta u Trajr knjižici vrši se tako da se između dviju postojećih trajektorija izvrši linearna interpolacija. Linearna interpolacija uvijek će rezultirati pozicijskim vrijednostima koje su prosjek početne i završne vrijednosti.



Slika 10. Primjer višestrukog uzorkovanja originalne putanje

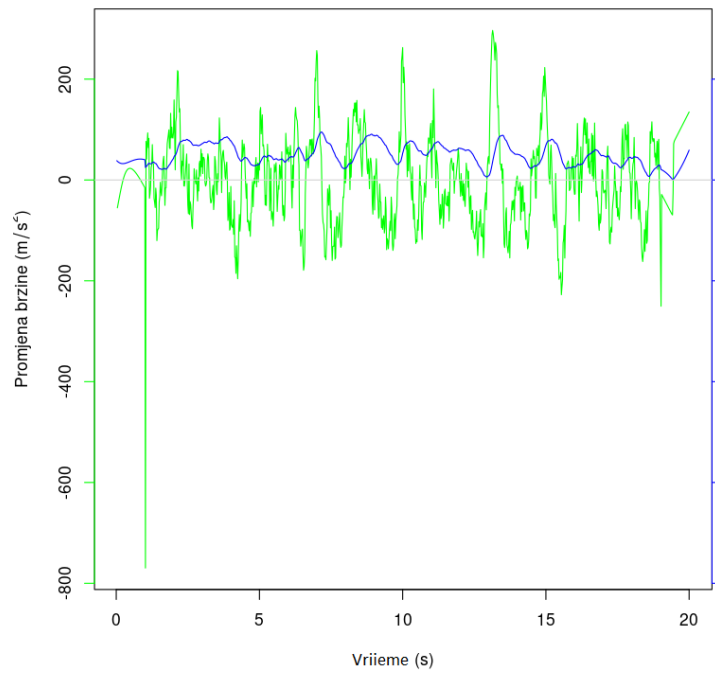
Na slici 10. višestruko uzorkovanje vršeno je na 50% manjem koraku od početnog seta podataka. Višestruko uzorkovanje prikazano je zelenim točkama na zelenoj putanji, dok je početni set podataka prikazan crnim točkama.

2.12. Analiza brzine

Trajr knjižica podržava analizu brzine u intervalima. Analiza brzine vrši se kako bi se pronašli intervali unutar kojih je kretanje brže ili sporije od tražene brzine kretanja.

```
function (trj)
{
  .checkTrajHasTime(trj)
  d <- Mod(trj$displacement)
  t <- trj$displacementTime
  v <- d[2:length(d)]/diff(t)
  vt <- t[2:length(t)]
  a <- diff(v)/diff(vt)
  at <- vt[2:length(vt)]
  list(speed = v, speedTimes = vt, acceleration = a, accelerationTimes = at)
}
```

Slika 11. Kod korišten u Trajr knjižici za analizu brzine i ubrzanja



Slika 12. Graf odnosa brzine i promjene brzine kroz vrijeme

Graf iz slike 12 dobiven je nad umjetnim podacima generiranim pomoću „TrajGenerate,, funkcije, prikazuje odnos brzine i promjene brzine kroz vrijeme na različitim skalama.

2.13. Indeks vijugavosti (Sinuosity index)

Indeks vijugavosti definira vijugavost trajektorije S kao:

$$S = 1.18 * \frac{\sigma}{\sqrt{q}}$$

gdje je:

σ standardna devijacija kuta koraka

q medijan veličine koraka

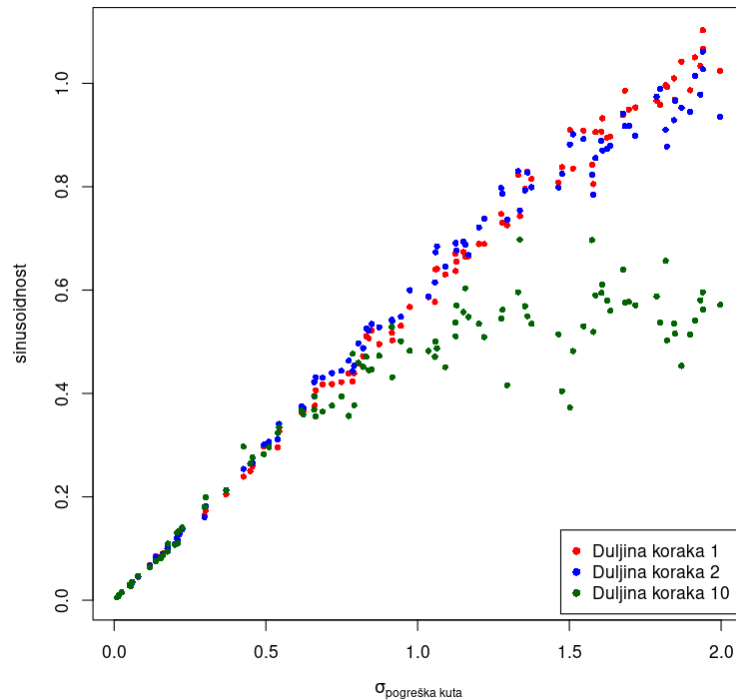
Trajr knjižica omogućuje i pristup poboljšanom indeksu vijugavosti S, za koji nije potrebno imati konstantnu duljinu koraka [7]:

$$S = 2 * \left[p * \frac{1 + c}{1 - c} + b^2 \right]^{-0.5}$$

gdje je:

c medijan kosinusa kuta

b koeficijent varijacije duljine koraka



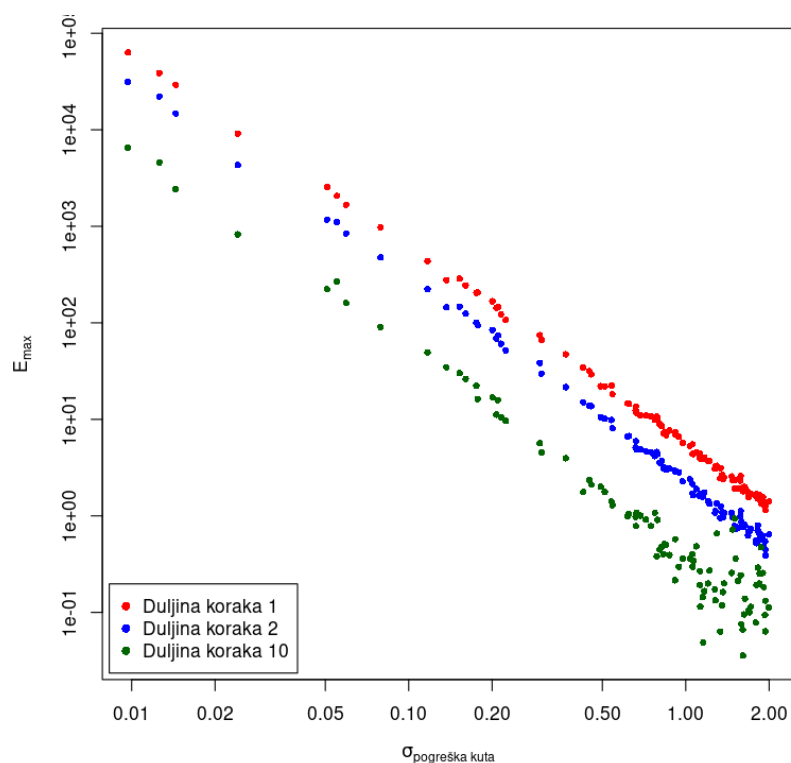
Slika 13. Omjer indeksa vijugavosti i pogreške kuta, u odnosu na različite duljine koraka

Na grafu iz slike 13, može se iščitati omjer indeksa vijugavosti i pogreške kuta, koji imaju pozitivnu korelaciju. Duljina koraka ima pozitivnu korelaciju s pogreškom kuta, te negativnu korelaciju s indeksom vijugavosti.

2.14. **Maksimalni očekivani pomak trajektorije Emax (eng. maximum expected displacement of a trajectory)**

Maksimalan očekivani pomak trajektorije Emax, mjera je ravnosti trajektorije. Veća vrijednost Emax varijable označava ravniju trajektoriju, dok za sinusoidne trajektorije teži u 0. Maksimalan očekivani pomak trajektorije može se vršiti pomoću dva različita algoritma, nazvanih Emax-a te Emax-b. Svaki od ovih algoritama može se izračunavati za podatke koji su nasumični ili usmjereni.

Ovime se dobiva četiri različite moguće kombinacije računanja maksimalnog očekivanog pomaka trajektorije.



Slika 14. Graf maksimalnog očekivanog pomaka trajektorije E_{max} u odnosu na pogrešku kuta

Na slici 14. prikazana je mjera ravnosti iste trajektorije kao i na slici 13 u odnosu na pogrešku kuta. Iz grafa vidi se negativna korelacija između očekivanog pomaka trajektorije i pogreške kuta. Duljina koraka negativno je korelirana s očekivanim pomakom trajektorije.

2.15. Fraktalna dimenzija (eng. fractal dimension)

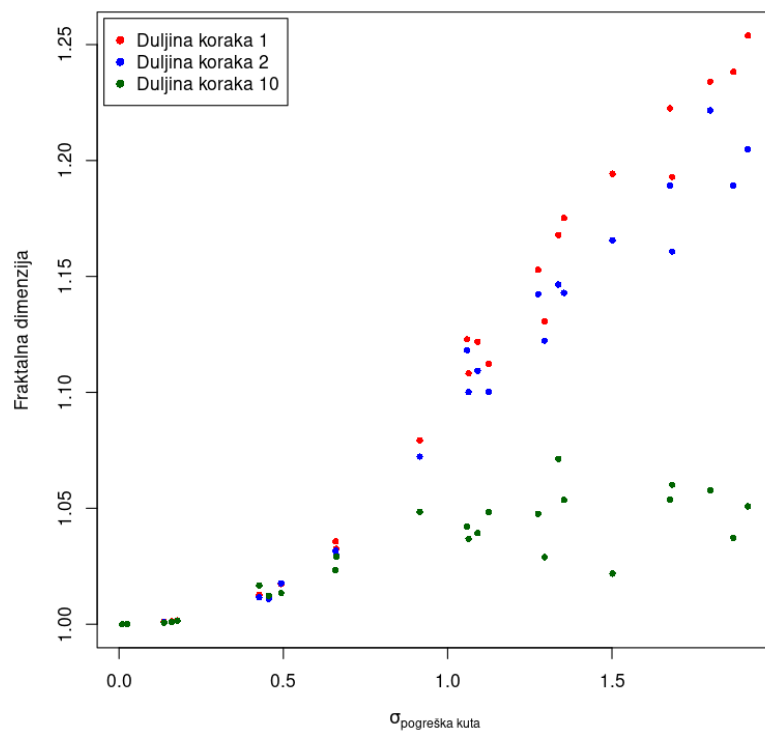
Knjižica Trajr omogućuje testiranje pripadnosti trajektorija fraktalnoj krivini. Kako bi trajektorija bila fraktalna krivina, unutar sebe mora sadržavati pod-trajektorije koje su savršena replika potpune trajektorije. Fraktalna dimenzija D računa se kao:

$$D = \frac{\log(N)}{\log(M)}$$

gdje je:

N broj sebi sličnih dijelova

M faktor uvećavanja

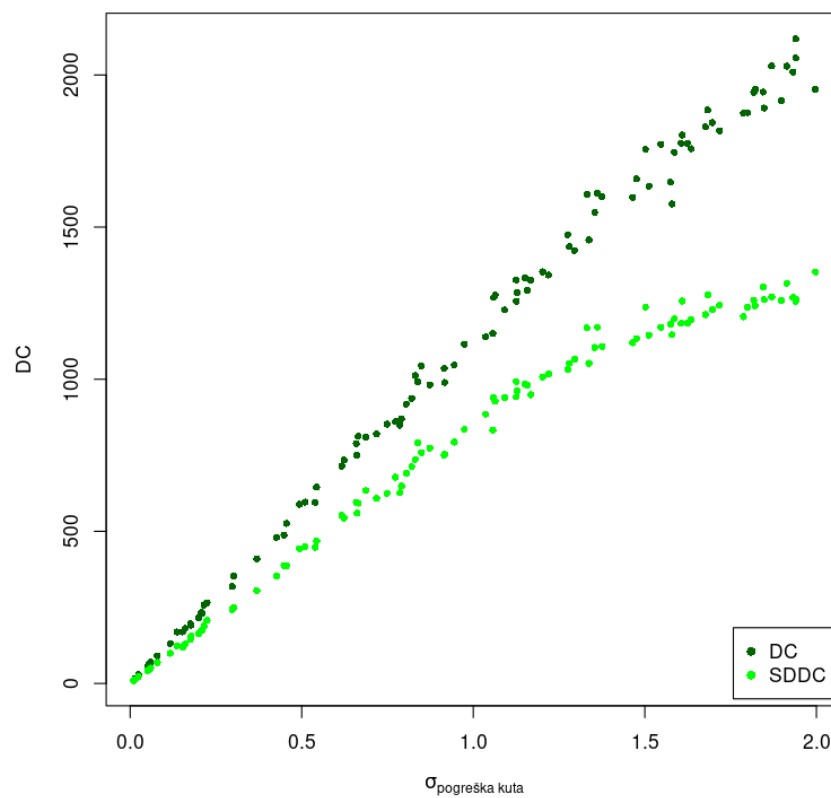


Slika 15. graf fraktalne dimenzije u odnosu na pogrešku kuta

Fraktalna dimenzija pozitivno korelira s pogreškom kuta u slučajevima manje duljine koraka.

2.16. Promjena smjera kretanja (eng. directional change)

Promjena smjera kretanja DC označava kutnu promjenu u stupnjevima između bilo koje dvije točke u trajektoriji, podijeljena vremenskom razlikom između te dvije točke:

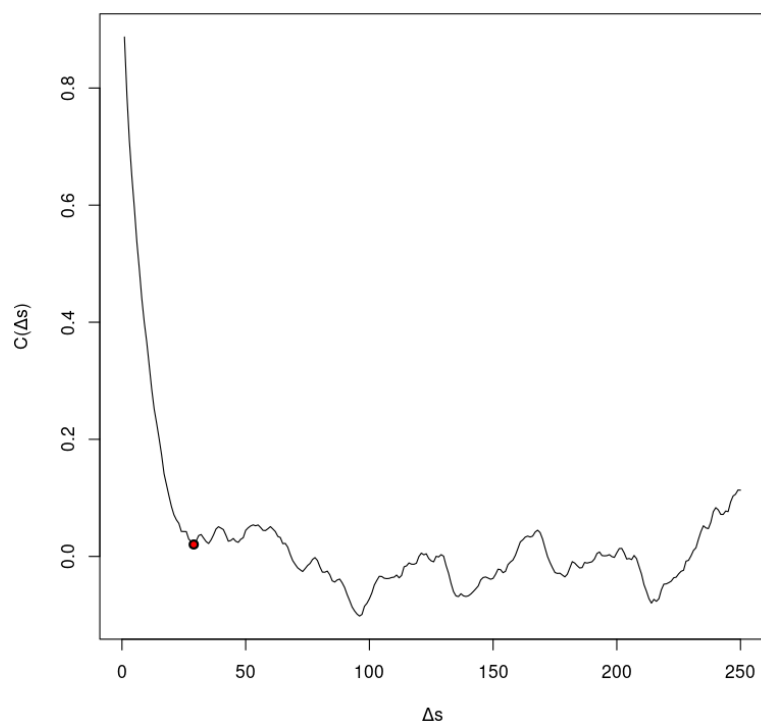


Slika 16. Graf prikaza promjene smjera kretanja u odnosu na pogrešku kuta

Promjena smjera kretanja ima pozitivnu korelaciju s pogreškom kuta.

2.17. Autokorelacija smjera (eng. directional autocorrelation)

Računanje autokorelacije smjera vrši se na potpunoj trajektoriji. Svi dijelovi trajektorije moraju biti jednakih duljina.



Slika 17. prikaz autokorelacije smjera na trajektoriji.

Slika 14. označava autokorelaciju smjera, crvenom točkom označen je 1. pronađen lokalni minimum. Lokalni minimum ne mora uvijek postojati. Lokalni minimum ne postoji u slučaju da autokorelacija smjera na trajektoriji tijekom cijele trajektorije pada.

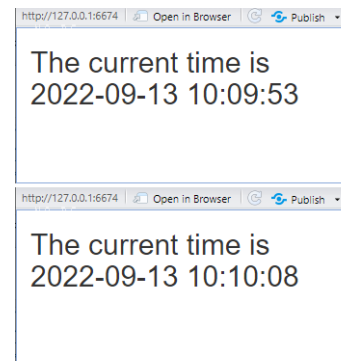
2.18. Shiny

Shiny je naziv R paketa koji služi za izradu interaktivnih grafičkih sučelja za R programska rješenja. Shiny omogućava laku integraciju HTML i CSS elemenata u R programskom okruženju. Osim HTML i CSS elemenata, podržano je i dinamičko mijenjanje grafičkog sučelja te dinamičko računanje funkcija. Programi napravljeni u Shiny R paketu dijele se na tri dijela, „ui“, „server“ te „shinyApp“. Ui (eng. user interface), označava dio koda namijenjen za grafičko sučelje.

```

library(shiny)
# Definiranje grafičkog sučelja za prikaz trenutnog vremena
ui <- fluidPage(
  h2("The current time is", textOutput("currentTime"))
)
# Definiranje serverske logike za prikaz trenutnog vremena
# mijenja se u vremenskoj rezoluciji od sekunde
server <- function(input, output, session) {
  output$currentTime <- renderText({
    invalidateLater(1000, session)
    paste(Sys.time())
  })
}
# Pokreni Shiny grafičku aplikaciju
shinyApp(ui, server)

```



Slika 18. Shiny programski kod i grafičko sučelje dobiveno pokretanjem koda

Lijeva strana slike 18 kod je za stvaranje jednostavnog interaktivnog korisničkog sučelja. Grafički element koji je izrađen jest zaglavlje veličine 2 (eng. header 2, kratica h2). Grafički element podijeljen je na statički element koji se ne mijenja („The current time is“) te dinamički element koji se mijenja svake sekunde (poziv funkcije textOutput(„currentTime“)). Iako se vidljiva promjena dešava u rezoluciji od jedne sekunde, sam kod izvršit će se mnogo češće, ovisno o procesorskoj moći računala na kojemu se kod odvija. Radi smanjenja bespotrebnog računanja funkcija, tvorcii Shiny paketa koriste reaktivne funkcije. Pomoću reaktivnih funkcija mijenja se samo dio koda koji je potrebno mijenjati, kada ga je potrebno mijenjati.



Slika 19. primjer složenijeg grafičkog sučelja izrađeno pomoću Shiny paketa

Na slici 19. vidi se primjer složenijeg grafičkog sučelja koje koristi nekoliko reaktivnih funkcija. Podatci o cijenama dionica dohvaćaju se s udaljenog poslužitelja. Kako se poslužitelj ne bi bespotrebno zatrpao zahtjevima, reaktivnim funkcijama odvojeno je dohvaćanje cijena dionica, koje ovisi samo o simbolu dionice te odabranom rasponu datuma. Ako korisnik odabere iscrtavanje grafa na logaritamskoj skali, ili prilagodbu cijena dionica inflaciji, nije potrebno ponovno dohvaćati cijene dionica s poslužitelja.

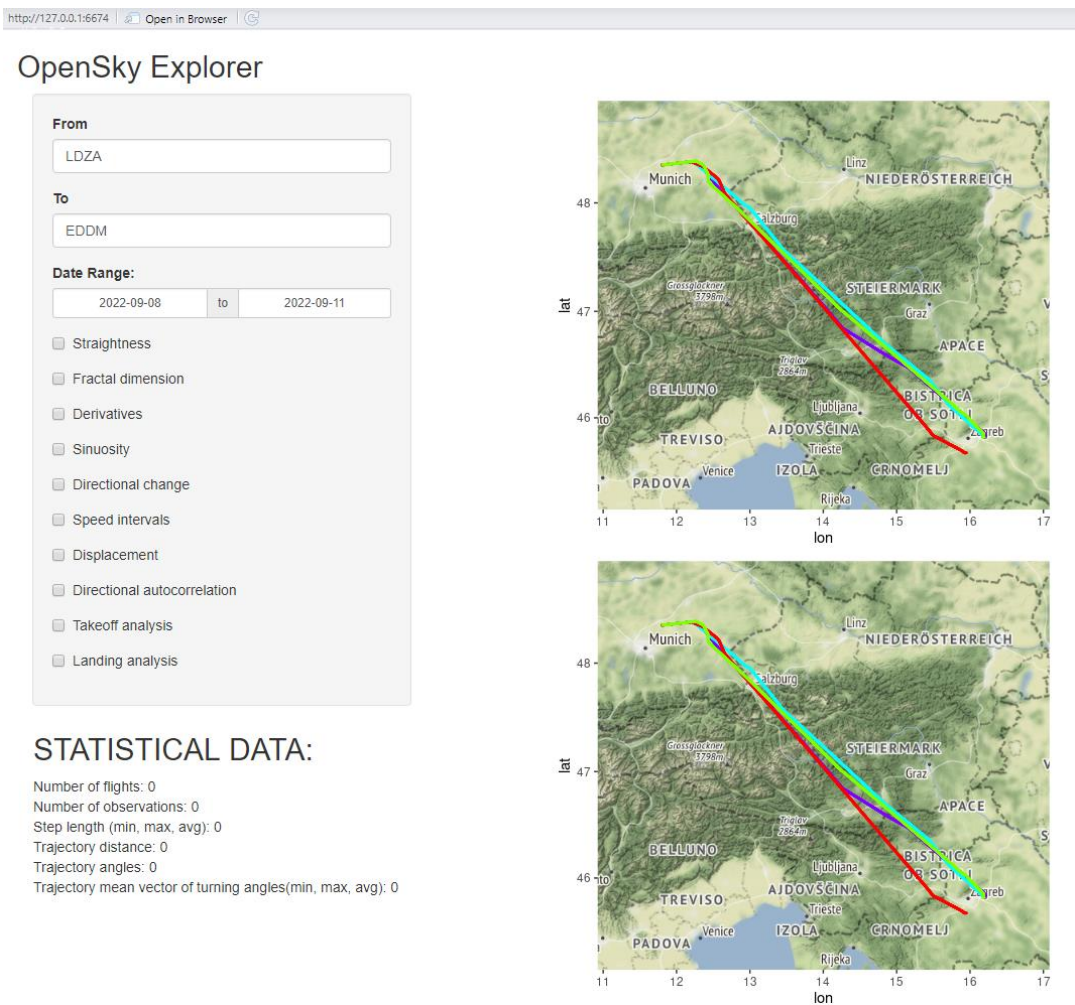
```
# Poslužiteljska logika
server <- function(input, output) {
  dataInput <- reactive({
    getSymbols(input$symb, src = "yahoo",
              from = input$dates[1],
              to = input$dates[2],
              auto.assign = FALSE)
  })
  finalInput <- reactive({
    if(!input$adjust){
      return(dataInput())
    }
    else {
      return(adjust(dataInput()))
    }
  })
  output$plot <- renderPlot({
    chartSeries(finalInput(), theme = chartTheme("white"),
              type = "line", log.scale = input$log, TA = NULL)
  })
}
```

Slika 20. primjer korištenja reaktivnih funkcija za izradu grafičkog sučelja iz slike 15.

Dijelovi koda za grafičko sučelje jesu grafički elementi poput gumbova, teksta, slika, tablica, grafova i slično. „Server“, odnosno poslužitelj označava dio koda koji se odvija u pozadini, te ga krajnji korisnik izravno ne vidi, već samo njegove rezultate. Pomoću „shinyApp“ naredbe pokreće se Shiny aplikacija, uobičajeno se nalazi na kraju svake Shiny R skripte.

3. IZRADA RJEŠENJA

Pregledavajući postojeća rješenja, zaključeno je kako trenutno za pristup OpenSky bazi podataka i analizu podataka iz dotične baze, ne postoji rješenje s grafičkim sučeljem izrađenim u programskom jeziku R. Kako bi se ovaj problem riješio, odlučeno je kako će se grafičko sučelje izraditi pomoću paketa za izradu grafičkih sučelja u programskom jeziku R, Shiny. Kao aplikacijska logika za dohvaćanje podataka iz OpenSky baze podataka, iskoristit će se knjižice `openskyr` te `openSkies`. Unutar knjižice `openskyr` implementirano je samo četiri funkcije za dohvaćanje podataka, no te četiri funkcije besprijekorno izvršavaju svoje funkcionalnosti. Zbog ove činjenice za osnovne funkcionalnosti iskoristit će se funkcije iz knjižice `openskyr`, dok će se složenije funkcionalnosti (kao što su: sortiranje vektora stanja u letove, filtriranje vektora stanja prema avionu, grafički prikaz podataka o letu pojedine letjelice) uzeti iz knjižice `openSkies`. Zato što nijedna knjižica ne podržava filtriranje i dohvaćanje letova od zračne luke A do zračne luke B, ova funkcionalnost će se dodati krajnjem programskom rješenju. Nad dobivenim podacima, izvršit će se nekoliko jednostavnijih analiza podataka: broj letova između odabranih zračnih luka u odabranom vremenskom intervalu, broj informacija (misli se na ukupan broj dohvaćenih vektora stanja), duljina koraka između dva vektora stanja (minimalna, maksimalna te prosječna), ukupna udaljenost trajektorije, prosječni kutovi trajektorije (po trajektoriji). Kako bi se olakšao odabir polazne i dolazne zračne luke, omogućit će se tekstualni unos željenih zračnih luka. Da bi se olakšao te standardizirao unos vremenskog intervala, koristit će se poseban element iz paketa Shiny za odabir vremenskog intervala. Ovakav način odabira vremenskog intervala umanjit će rezoluciju mogućeg odabira intervala na jedan dan, no pojednostavit će način odabira vremenskog intervala pružajući korisniku lako čitljiv i interaktivan način odabira vremenskog intervala, koji će se zatim, u pozadini pretvoriti u oblik koji odgovara pojedinoj knjižici za pristup bazi podataka OpenSky ili samoj bazi podataka (Unix format vremena).



Slika 21. inicijalan primjer izgleda grafičkog sučelja

Shiny paket za izradu grafičkih sučelja podržava statičke i dinamičke funkcije u pozadini. Statičke funkcije smatraju se one funkcije koje se izvršavaju izravno korisničkim zahtjevom (pritiskom na gumb) dok se dinamičke funkcije u pozadini izvršavaju bez potrebe za eksplicitnim odabirom korisnika. U ovom radu korištena su oba načina rada funkcija. Statičke funkcije korištene su za vremenski intenzivnije dijelove koda, kao što je dobavljanje informacija iz baze podataka te filtriranje tih podataka. Dinamičke funkcije korištene su za obavljanje pojedinih statističkih analiza koje ne ovise jedna o drugoj. Kada korisnik odabere neku od željenih statističkih funkcija, njezino računanje odmah će se pokrenuti u pozadini, za vrijeme tokom kojeg korisnik odabire druge statističke funkcije. Ovo se čini kako bi se korisniku pružio privid bržeg računanja funkcija, te da se izbjegne računanje svih statističkih analiza odjednom. Potencijalan problem koji se javlja u ovakvom načinu rada, jest ako korisnik odabere, te zatim makne iz odabira neku od funkcija, u kojem slučaju nije bilo potrebno računati tu funkciju. Funkcije koje su odabrane za statističku analizu nad podatcima jesu: Analiza indeksa ravnosti (2.9.), analiza indeksa

vijugavosti (2.13.), analiza maksimalnog očekivanog pomaka trajektorije (2.14), analiza fraktalne dimenzije (2.15.), analiza promjene smjera kretanja (2.16.), analiza brzine i akceleracije (2.12.).

```
function(departureAirport, destinationAirport, beginTime, endTime){
  beginTime <- paste(beginTime, "00:00:00")
  endTime <- paste(endTime, "00:00:00")
  data_airport_df <- get_airport_data(username = "", password = "",
                                     option = "departures", airport = depar-
tectureAirport,
                                     begin = as.integer(as.POSIXct(beginTime)),
                                     end = as.integer(as.POSIXct(endTime)))
  num_of_flights <- nrow(data_airport_df[data_airport_df$estArrivalAirport ==
destinationAirport, ])
  flights_vector <- vector(mode = "list", num_of_flights)
  airplane_identifiers <- data_airport_df[data_airport_df$estArrivalAirport ==
destinationAirport, ]$icao24
  airplane_first_seen <- data_airport_df[data_airport_df$estArrivalAirport ==
destinationAirport, ]$firstSeen
  airplane_last_seen <- data_airport_df[data_airport_df$estArrivalAirport == des-
tinationAirport, ]$lastSeen
  for(i in 1:num_of_flights){
    flights_vector[[i]] <- getIntervalStateVectors(aircraft = airplane_identifi-
ers[[i]],
                                                    startTime = as.POSIXct(air-
plane_first_seen[[i]], origin = "1970-01-01", tz = "Europe/Zagreb"),
                                                    endTime = as.POSIXct(air-
plane_last_seen[[i]], origin = "1970-01-01", tz = "Europe/Zagreb"))
  }
  length(flights_vector)
  flights <- vector(mode = "list", length = length(flights_vector))
  for(i in 1:length(flights)){
    flights[i] <- flights_vector[[i]]$split_into_flights()
  }
  paths <- vector(mode = "list", length = length(flights_vector))
  for(i in 1:length(flights)){
    paths[[i]] <- flights[[i]]$state_vectors
  }
  mapa <- plotRoutes(paths, pathColors = rainbow(length(flights)))
  return(list(paths, length(flights)))
}
```

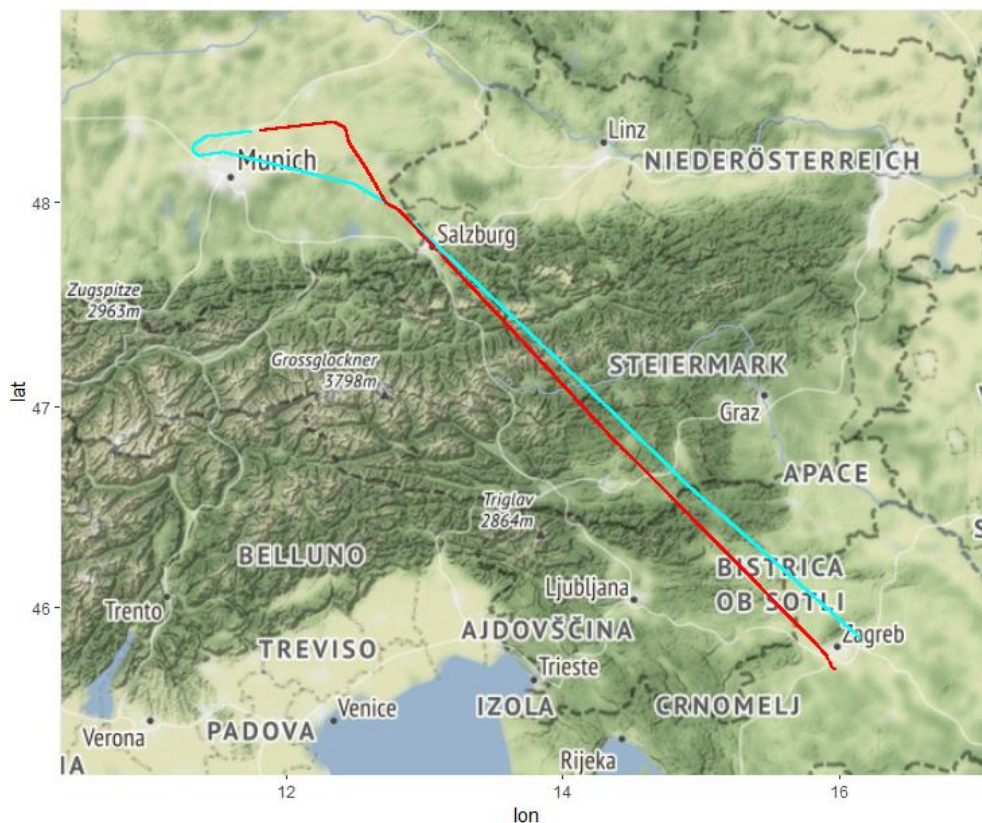
Slika 22. Funkcija za dohvaćanje letova iz zračne luke A u zračnu luku B

Grafičko sučelje bit će podijeljeno na dva velika dijela: lijevi dio, koji predstavlja tekstualan dio grafičkog sučelja, te desni dio, namijenjen grafičkim prikazima i analizama. Lijevi dio podijeljen je na tri manja dijela: dijelovi grafičkog sučelja čijom će se promjenom statički pokretati

funkcije u pozadini (odabir zračnih luka te vremenskog intervala), dijelovi grafičkog sučelja čijom će se promjenom dinamički pokretati funkcije u pozadini (odabir statističkih analiza), te dio namijenjen za rezultate statističkih analiza.

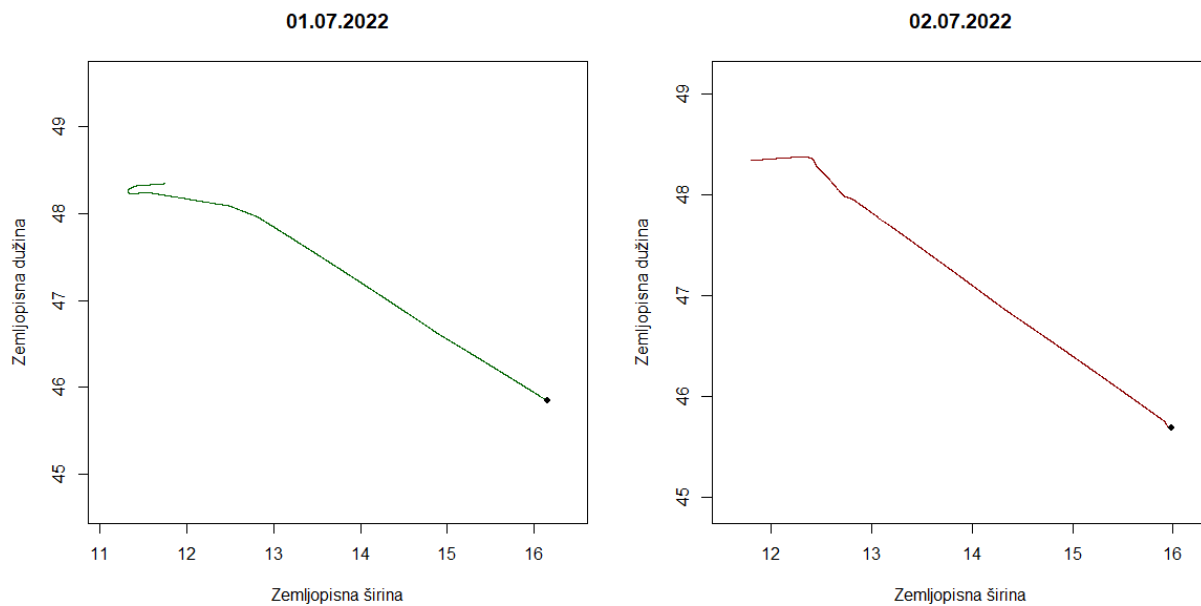
4. REZULTATI ISTRAŽIVANJA

U svrhu testiranja programskog rješenja, za početak, sve analize izvršile su se nad dva različita leta: let iz Zagreba u Muenchen na datum 01.07.2022., te let iz Zagreba u Muenchen na datum, 02.07.2022.



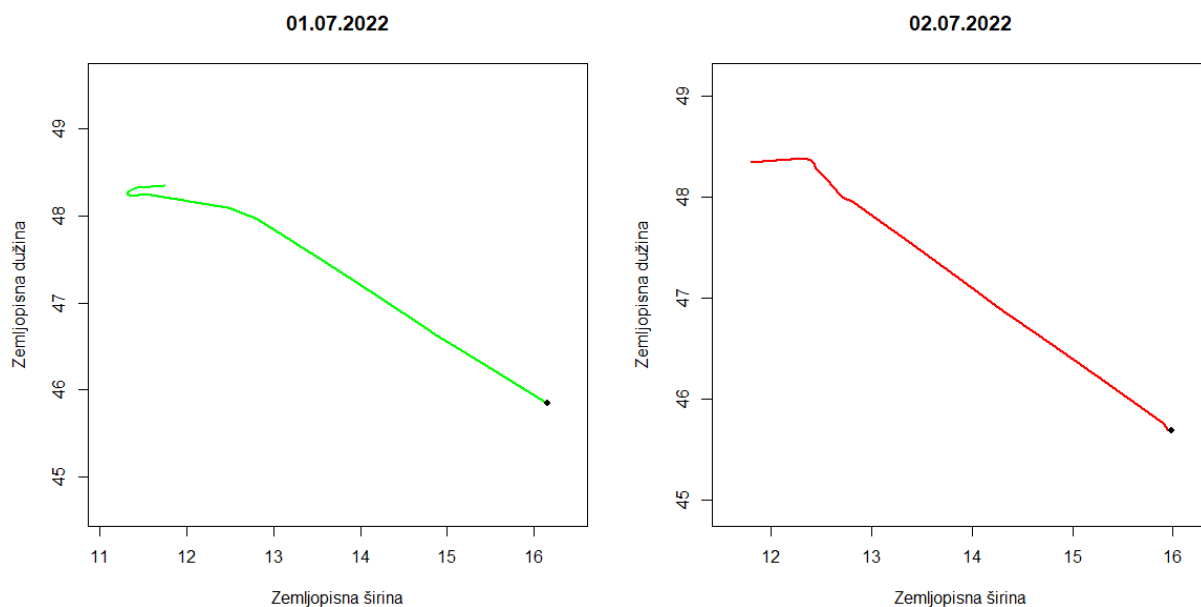
Slika 23. Prikaz leta Zagreb - Muenchen

Let izvršen na datum 01.07.2022. prikazan je svijetlo plavom bojom, dok je let izvršen na datum 02.07.2022. prikazan crvenom bojom. Već iz početnog grafa, može se naslutiti kako ova dva leta sadrže neke različitosti. Iako se kreću na istoj relaciji (Zagreb - Muenchen), letovi polijeću s različitih strana zračne luke u Zagrebu, te slijeću na različite strane zračne luke u Muenchenu. Daljnjom analizom podataka dobit će se uvid u to koliko ovo utječe na indikatore leta.



Slika 24. Originalne putanje iscrtane na dva grafa

Putanje su razdvojene grafički te brojevno u svrhu izvršavanja lakše analize (vizualne i numeričke) podataka. Kako bi se izbjegle nagle promjene i umanjio utjecaj šuma, putanje su izglađene pomoću Savitzky-Golay filtera.



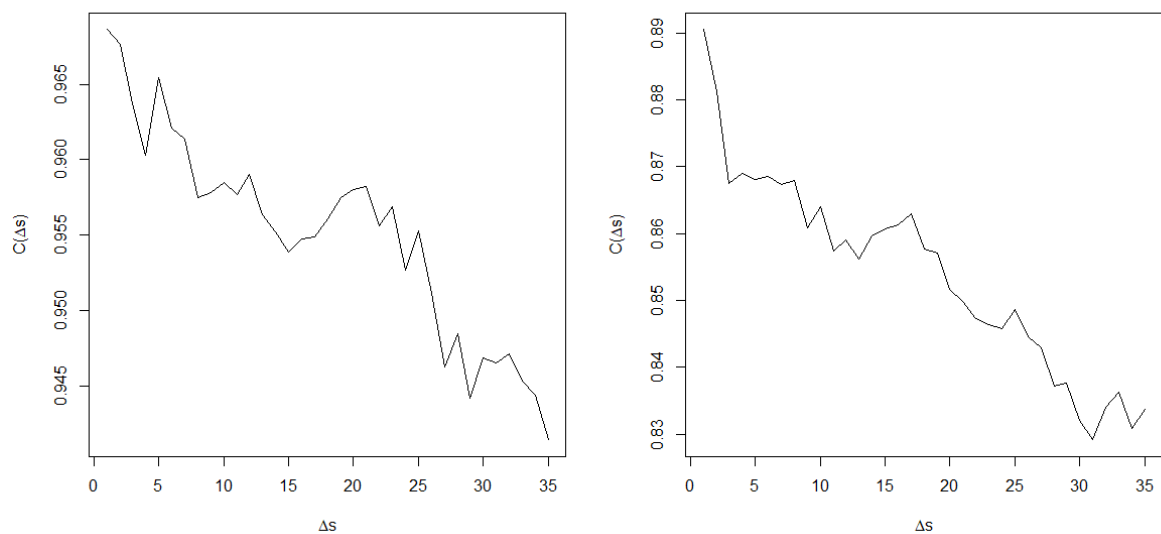
Slika 25. Izglađene putanje pomoću algoritma (2.10.)

Za izgladivanje putanja, korišten je polinom 3. reda, na veličini koraka od 101. Broj opaženih putanja ovisi o, duljini leta, brzini leta te broju trenutno dostupnih prijemnika u blizini trajektorija leta. Zbog velikog broja opaženih putanja te tendencije zrakoplova da ne vrše nagla skretanja, iz grafova se vidi kako izgladene putanje skoro savršeno odgovaraju originalnim putanjama.

Tablica 5.1. rezultati statističkih analiza letova za 2 različita leta.

Izvršena analiza	Let 01.07.2022.	Let 02.07.2022.
Broj opaženih trajektorija	3527	2981
Indeks ravnosti	0.8527582	0.9706249
Indeks vijugavosti 1.	10.02956	18.05028
Indeks vijugavosti 2.	6.384211	11.6538
Usmjeren maksimalni očekivani pomak Emax-a	2.732778	4.674633
Maksimalni očekivani pomak Emax-a	28.54818	7.815213
Usmjeren maksimalni očekivani pomak Emax-b	0.00460312	0.007993793
Maksimalni očekivani pomak Emax-b	0.04808686	0.0133643
Fraktalna dimenzija	1.037215	1.007624
Promjena smjera kretanja (medijan)	93.60188	254.2541
Promjena smjera kretanja (standardna devijacija)	490.827	869.6646

Bitna razlika koja se vidi iz analize podataka jest broj opaženih trajektorija. Let 01.07.2022. ima 546 opažanja više u odnosu na let 02.07.2022. Analiza indeksa ravnosti potvrđuje ono što se može naslutiti iz grafova leta, to da je let izvršen 02.07.2022. ravniji prema indeksu ravnosti (0.9706249) od leta izvršenog 01.07.2022. Po indeksu ravnosti (2.9.) trajektorija se smatra ravnijom ako je njezin indeks bliži vrijednosti 1 (što je i maksimalna vrijednost, za savršeno ravnu trajektoriju). Indeks vijugavosti izračunat je pomoću oba algoritma vijugavosti (2.13.), te su za oba dobiveni slični rezultati. Prema ovim pokazateljima, let izvršen 01.07.2022. jače je vijugav od leta izvršenog 02.07.2022. Trajektorija se smatra jače vijugavom, ako njezina vrijednost indeksa vijugavosti više teži nuli.



Slika 26. Graf auto korelacija smjera

Graf auto korelacija smjera pokazuje kako let izvršen 1.7.2022. ima višu auto korelaciju smjera od leta 2.7.2022. Auto korelacija smjera osjetljiva je na nagle pomake u početku izvršavanja trajektorije.

Tablica 5.3. rezultati analiza brzine za dva različita mjeseca

Izvršena analiza	Veljača 2022.	Srpanj 2022.
Broj opaženih trajektorija	89847	78079
BRZINA		
Maksimalna brzina	218.2916 m/s	247.0756 m/s
Minimalna brzina	38.29067 m/s	53.91854 m/s
Medijan brzine	135.3951 m/s	143.2299 m/s
Standardna devijacija brzine	30.56778 m/s	29.3451 m/s
INDEKS RAVNOSTI	Veljača 2022.	Srpanj 2022.
Maksimalan indeks ravnosti	0.9867944	0.9897891
Minimalan indeks ravnosti	0.8611952	0.8527582
Medijan indeksa ravnosti	0.9666587	0.9330051
Standardna devijacija indeksa ravnosti	0.0322326	0.05068917
INDEKS VIJUGAVOSTI		

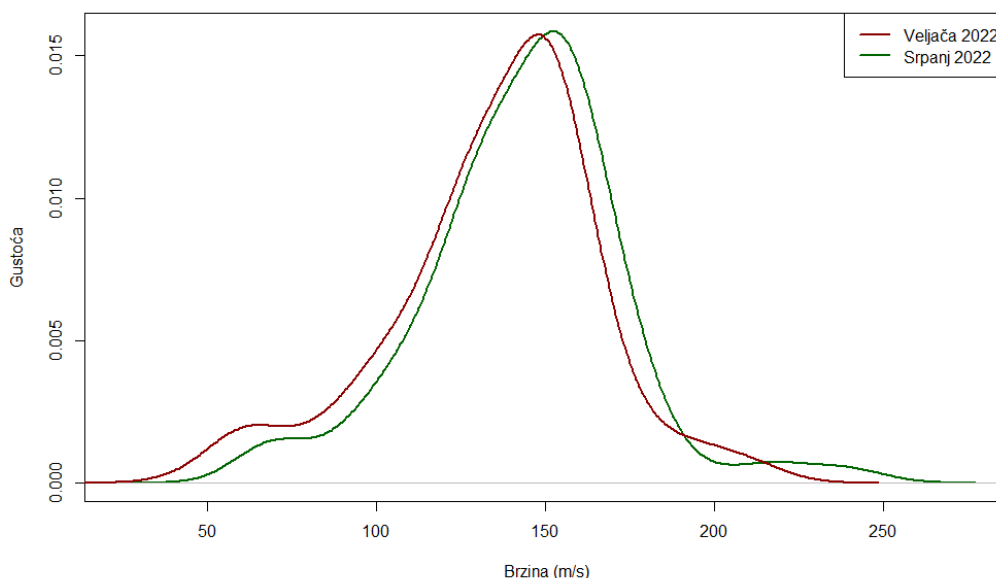
Maksimalan indeks vijugavosti	18.06531	25.91525
Minimalan indeks vijugavosti	6.808681	7.869075
Medijan indeksa vijugavosti	10.57091	13.5166
Standardna devijacija indeksa vijugavosti	2.640424	3.447526
INDEKS VIJUGAVOSTI 2		
Maksimalan indeks vijugavosti	11.58145	16.20399
Minimalan indeks vijugavosti	4.649456	5.041137
Medijan indeksa vijugavosti	6.852001	8.582339
Standardna devijacija indeksa vijugavosti	1.653927	2.152917
EMAX-A NEUSMJERENI		
Maksimalan Emax-a neusmjereni	61.33119	47.62896
Minimalan Emax-a neusmjereni	8.698035	4.375598
Medijan Emax-a neusmjereni	30.53969	18.18933
Standardna devijacija Emax-a neusmjereni	11.51771	8.557571
EMAX-A USMJERENI		
Maksimalan Emax-a usmjereni	13.85676	11.13117
Minimalan Emax-a usmjereni	2.856737	1.994188
Medijan Emax-a usmjereni	9.85797	5.506725
Standardna devijacija Emax-a usmjereni	2.947228	2.804183
EMAX-B NEUSMJERENI		
Maksimalan Emax-b neusmjereni	0.09168642	0.07754597
Minimalan Emax-b neusmjereni	0.01390009	0.006512267
Medijan Emax-b neusmjereni	0.04772329	0.03023522
Standardna devijacija Emax-b neusmjereni	0.01897477	0.01393648
EMAX-B USMJERENI		
Maksimalan Emax-b usmjereni	0.02274976	0.01735043
Minimalan Emax-b usmjereni	0.004701844	0.004593673
Medijan Emax-b usmjereni	0.01518738	0.008943763
Standardna devijacija Emax-b usmjereni	0.004460494	0.004153516
FRAKTALNA DIMENZIJA		
Maksimalna fraktalna dimenzija	1.034717	1.037215
Minimalna fraktalna dimenzija	1.002619	1.002177
Medijan fraktalne dimenzije	1.00813	1.016623
Standardna devijacija fraktalne dimenzije	0.008371216	0.01284791

PROMJENA SMJERA KRETANJA		
Maksimalni medijan	235.5936	1.037215
Minimalan medijan	63.27704	1.002177
Medijan, medijana promjene smjera kretanja	105.6381	1.016623
Standardna devijacija standardne devijacije	39.75782	0.01284791
Maksimalna standardna devijacija	1.034717	1.037215
Minimalna standardna devijacija	1.002619	1.002177

Tablica 5.3. prikazuje rezultate dobivene iz analize opaženih trajektorija tijekom veljače 2022. te srpnja 2022. Tijekom veljače, između Zagreba i Muenchena izvršeno je 27 letova, tokom kojih je zabilježeno 89847 trajektorija. Tijekom srpnja, izvršeno je 24 leta sa zabilježenih 78079 trajektorija. Početna analiza podataka koja je izvršena jest analiza brzina letjelica u pojedinim trajektorijama.

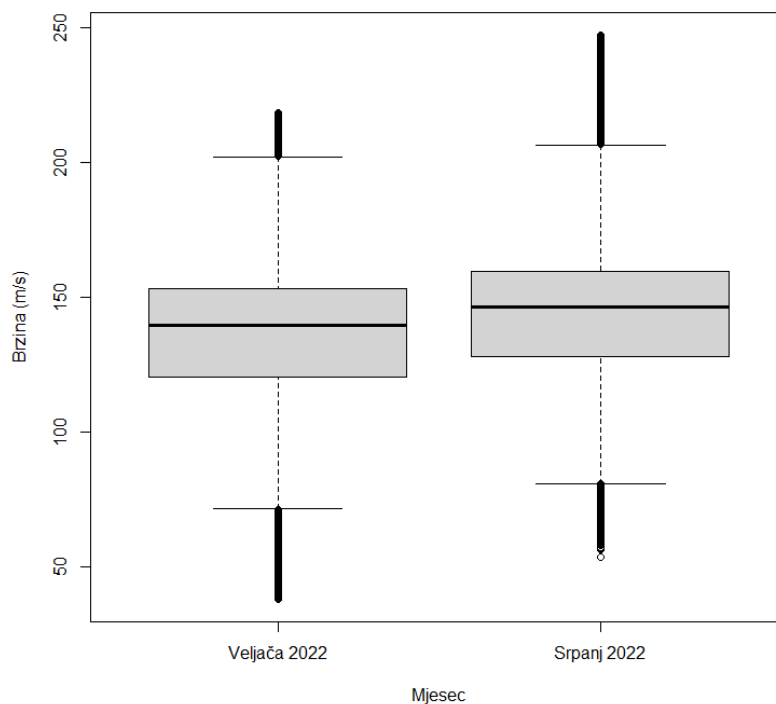
Tablica 5.4. kvantili brzine

	10%	25%	50%	75%	90%
Veljača	95.2722	120.53040	139.69405	153.22488	165.6486
Srpanj	106.4104	128.1203	146.5588	159.47515	171.8228



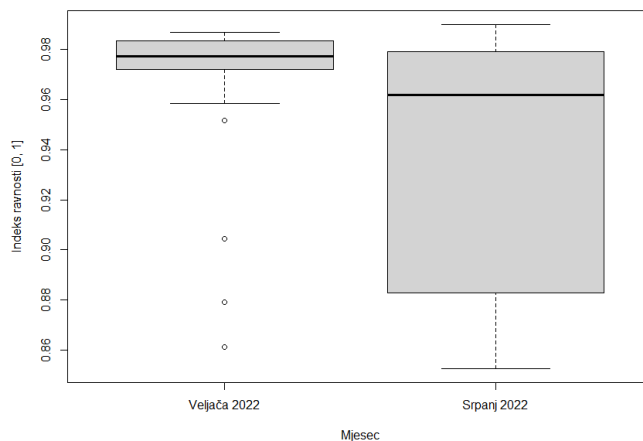
Slika 27. Funkcija gustoće brzine

Izračunate su maksimalne i minimalne vrijednosti brzine kroz dostupne podatke (rubni slučajevi), te prosječna vrijednost brzine. Podatci ukazuju na razlike između zapaženih brzina za veljaču i srpanj.



Slika 28. Kutijasti dijagram brzina leta

Kutijasti dijagram slike 28 pokazuje rezultate analize brzine. Tijekom srpnja, prosječna brzina letjelice na relaciji Zagreb – Muenchen veća je nego u veljači. Maksimalna brzina letjelice u srpnju 2022. za 28.784 m/s je veća od maksimalne brzine letjelice u veljači. Razlika između njihovih prosječnih vrijednosti iznosi 7.8348 m/s.



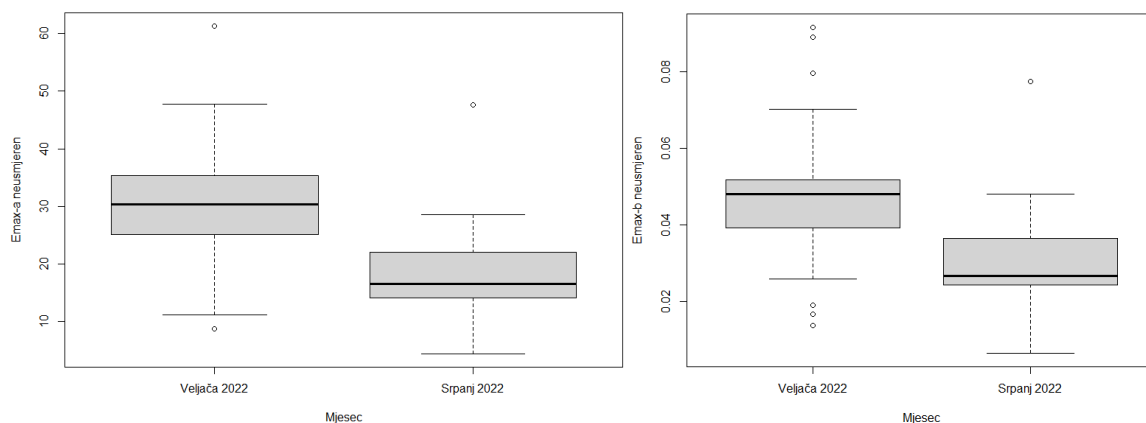
Slika 29. Kutijasti dijagram indeksa ravnosti leta

Analizom indeksa ravnosti, zaključujemo kako se same putanje leta tijekom veljače 2022. te srpnja 2022. razlikuju.

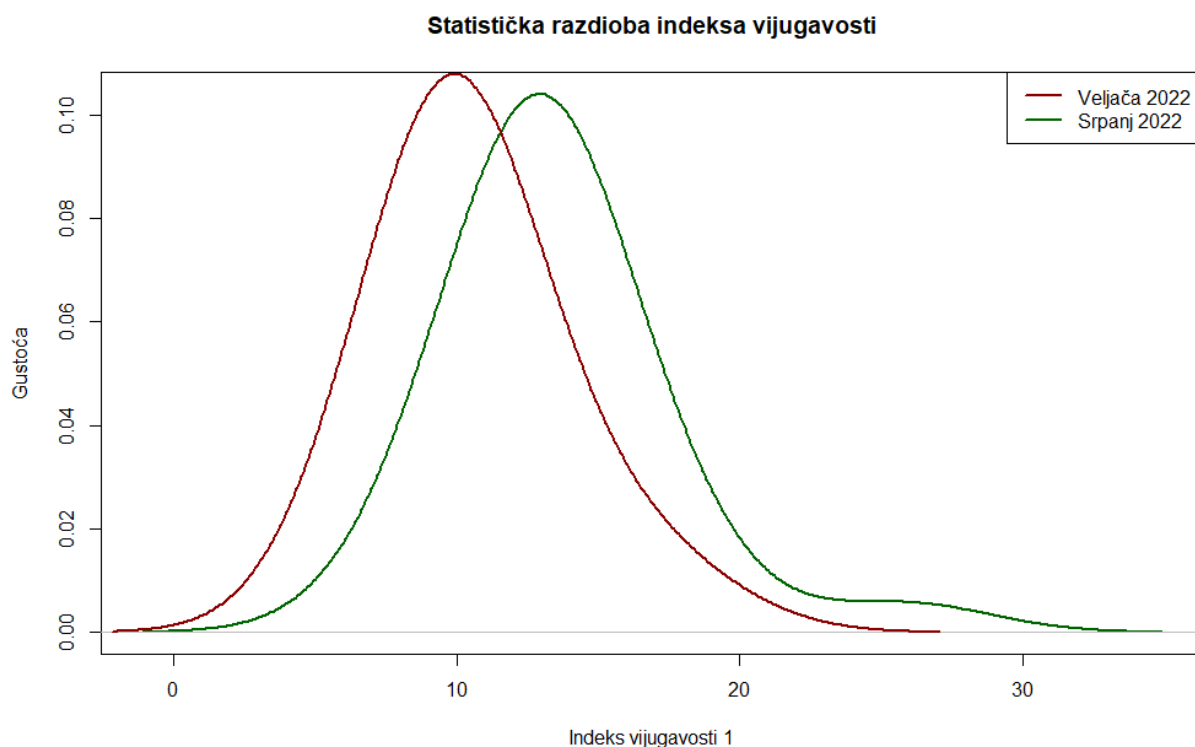
Tablica 5.5. kvantili indeksa ravnosti

	10%	25%	50%	75%	90%
Veljača	0.9328014	0.9718746	0.9772180	0.9832105	0.9854640
Srpanj	0.8704727	0.8830056	0.9618748	0.9775717	0.9863257

Letovi tijekom veljače međusobno su sličniji prema indeksu ravnosti nego letovi tijekom srpnja. Prema analizi ravnosti, letovi tijekom veljače ravniji su nego letovi tijekom srpnja. Ovu tvrdnju dodatno potvrđujemo analizom indeksa vijugavosti (slika 32).

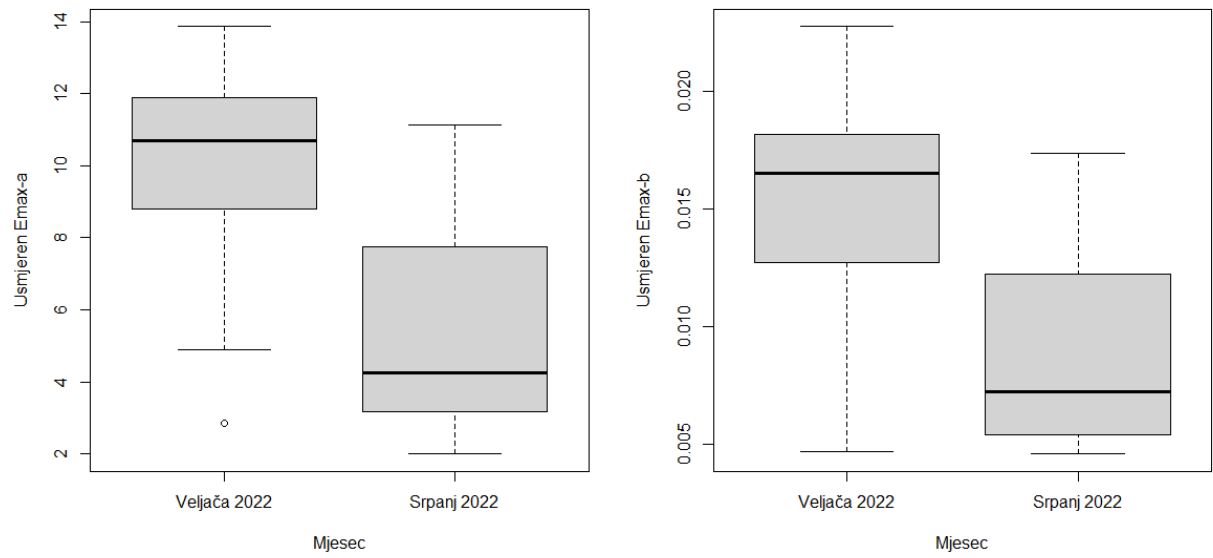


Slika 30. Kutijasti dijagrami neusmjerenog maksimalnog očekivanog pomaka



Slika 31. statistička razdioba indeksa vijugavosti

Kod analize maksimalnog očekivanog pomaka izvršena je analiza na 4 različita načina. Analiza neusmjerenog maksimalnog očekivanog pomaka Emax-a analizom, analiza neusmjerenog maksimalnog očekivanog pomaka Emax-b analizom, analiza usmjerenog maksimalnog očekivanog pomaka Emax-a analizom te analiza usmjerenog maksimalnog očekivanog pomaka Emax-b analizom. Sve četiri analize pokazuju slične rezultate. Letovi tijekom veljače bolje su usmjereni u smjeru Zagreb – Muenchen, nego letovi tijekom srpnja. Ovaj zaključak može se potkrijepiti i analizom indeksa ravnosti leta (slika 29.). U nastavku, prikazani su kutijasti dijagrami usmjerenog maksimalnog očekivanog pomaka, analizom Emax-a i Emax-b, kutijasti dijagram indeksa vijugavosti, kutijasti dijagram fraktalne dimenzije, te kutijasti dijagram standardne devijacije promjene smjera kretanja.



Slika 32. Kutijasti dijagrami usmjerenog maksimalnog očekivanog pomaka

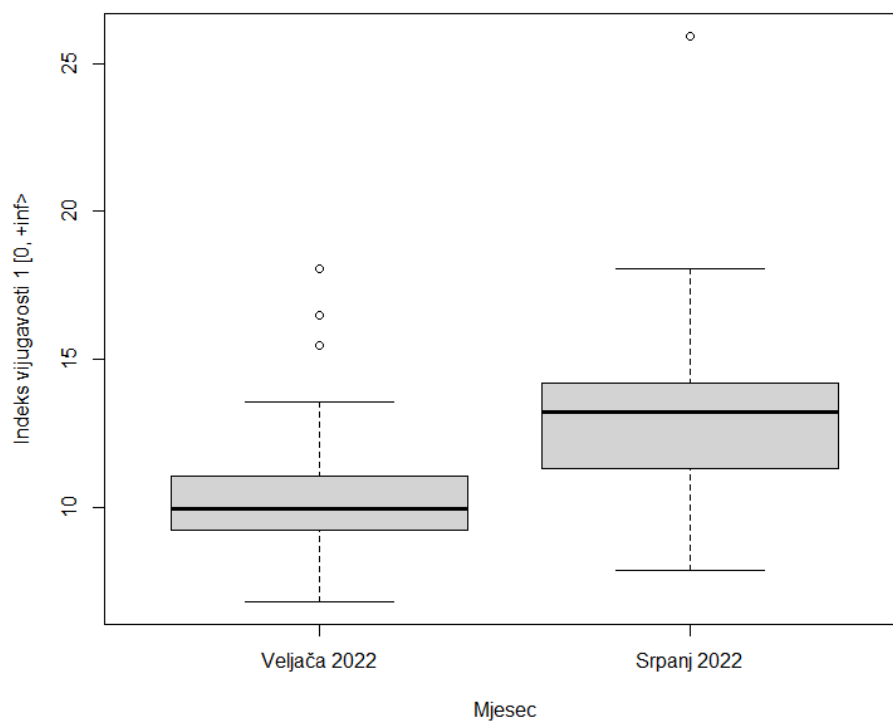
5. ZAKLJUČAK

U ovome radu, predstavili smo OpenSky bazu podataka, programski jezik R, nekoliko aplikacijskih sučelja izrađenih u programskom jeziku R za pristup OpenSky bazi podataka, knjižicu Trajr za dvodimenzionalnu analizu putanja te Shiny, alat za izradu grafičkih sučelja za programski jezik R.

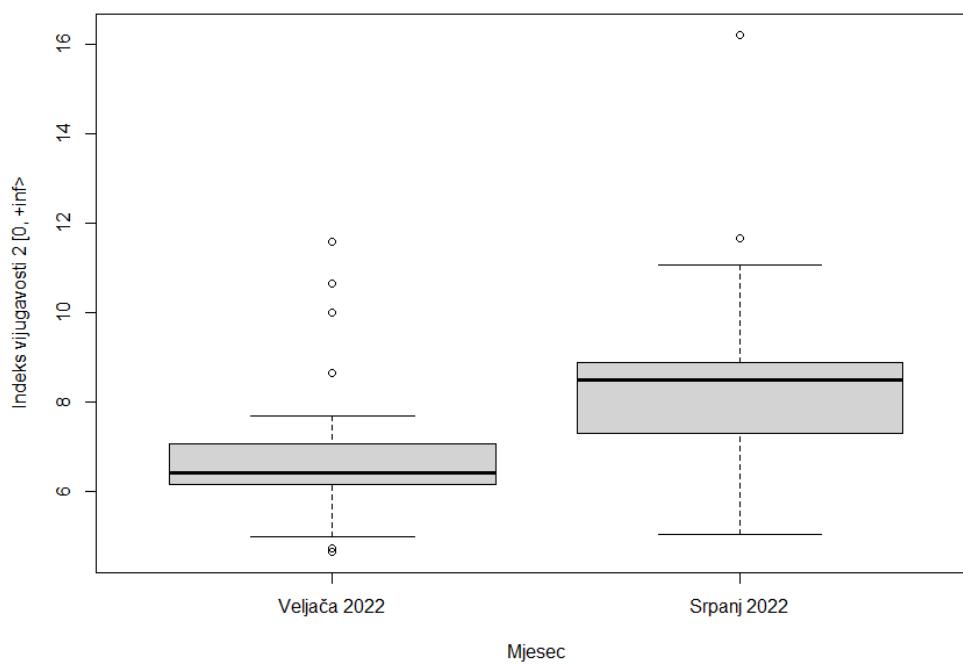
Pregledom postojećih rješenja zaključeno je kako trenutno za programski jezik R ne postoji službeno podržan alat koji omogućuje analizu putanja zrakoplova. Analiza putanja zrakoplova izvršena je uz pomoć knjižice Trajr. Kako bi se olakšalo korištenje alata, izrađeno je i grafičko sučelje pomoću kojeg se na interaktivan način može unijeti podatke nad kojima se želi izvršiti analiza. Trenutni alati za pristup OpenSky bazi podataka ne omogućuju filtriranje podataka prema željenim zračnim lukama, pa je i ova funkcionalnost dodana završnom programskom rješenju.

Dobiveni rezultati dokazuju kako je korištenjem izrađenog alata moguće je odrediti karakteristična obilježja putanja zrakoplova između dvije zračne luke.

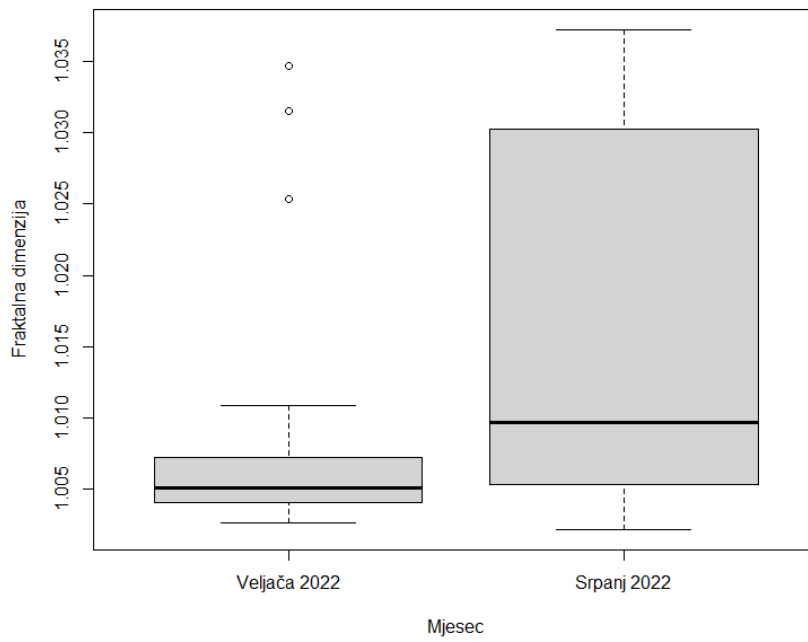
6. DODATAK



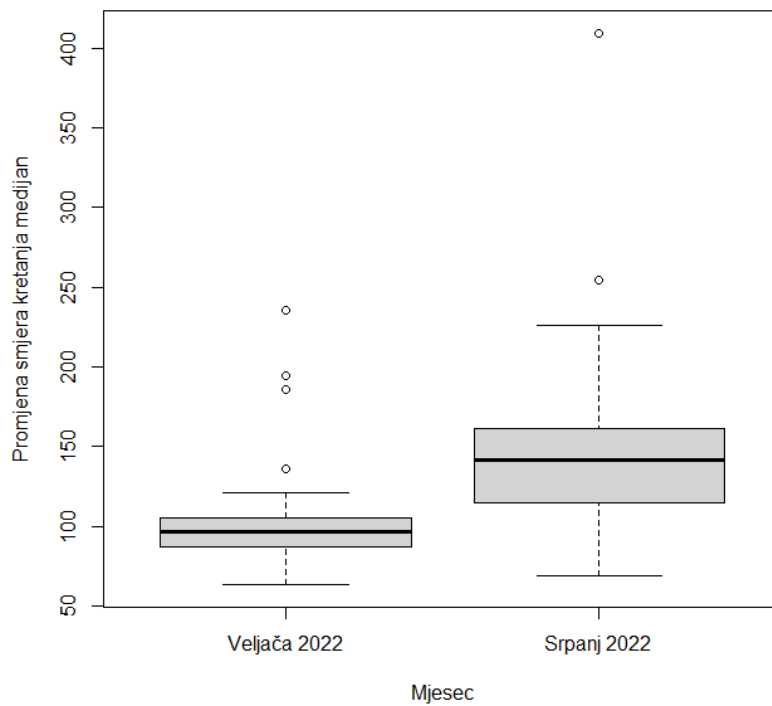
Slika 33. Kutijasti dijagram indeksa vijugavosti 1



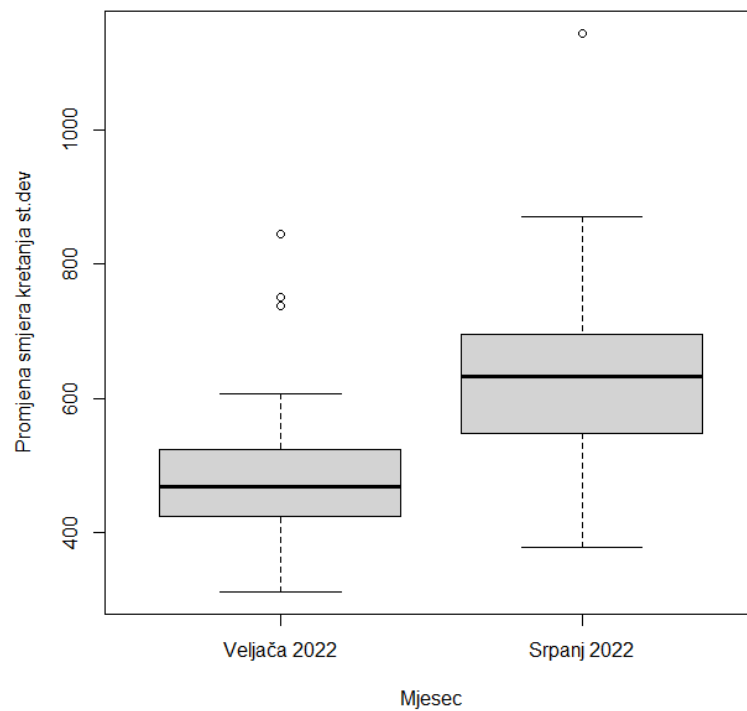
Slika 34. Kutijasti dijagram indeksa vijugavosti 2



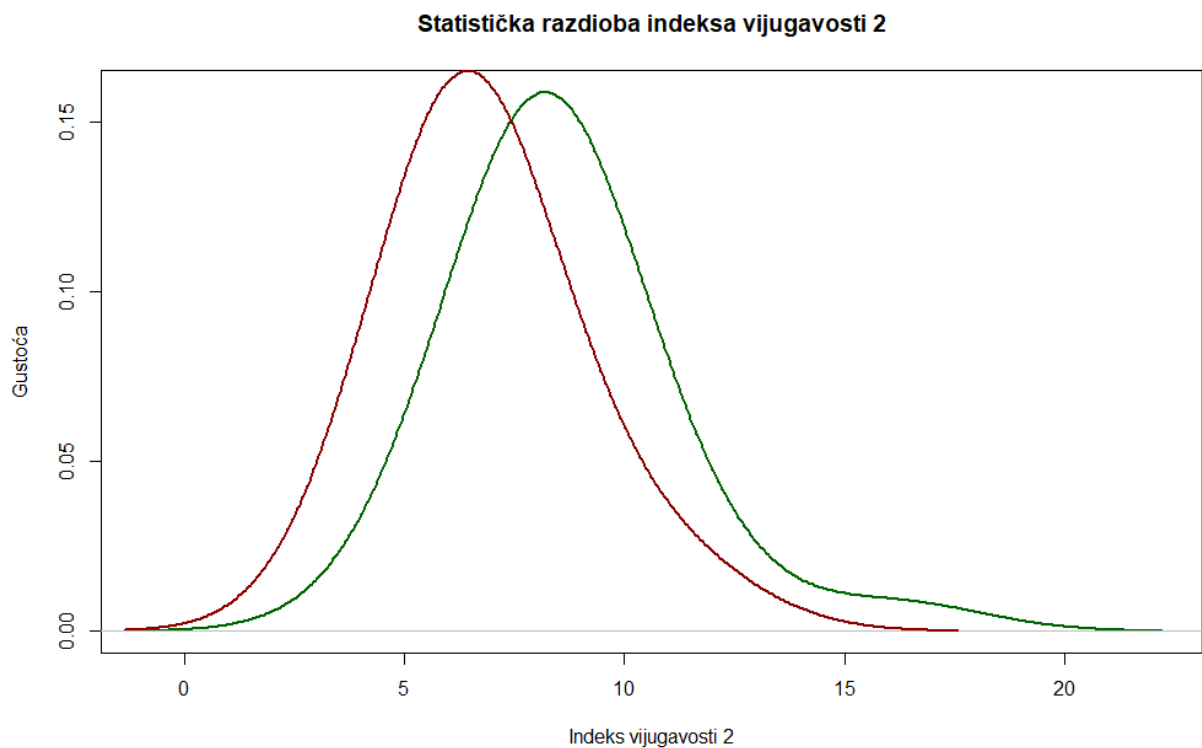
Slika 35. Kutijasti dijagram fraktalne dimenzije



Slika 36. Kutijasti dijagram medijana promjene smjera kretanja

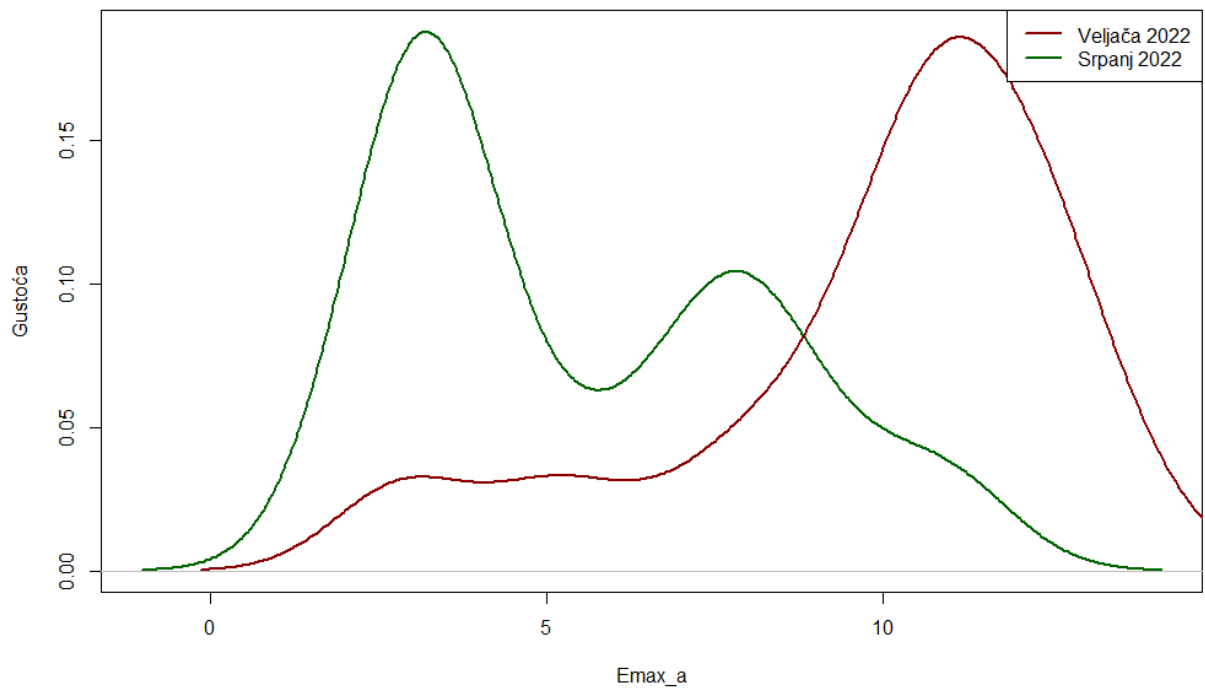


Slika 37. Kutijasti dijagram standardne devijacije promjene smjera kretanja



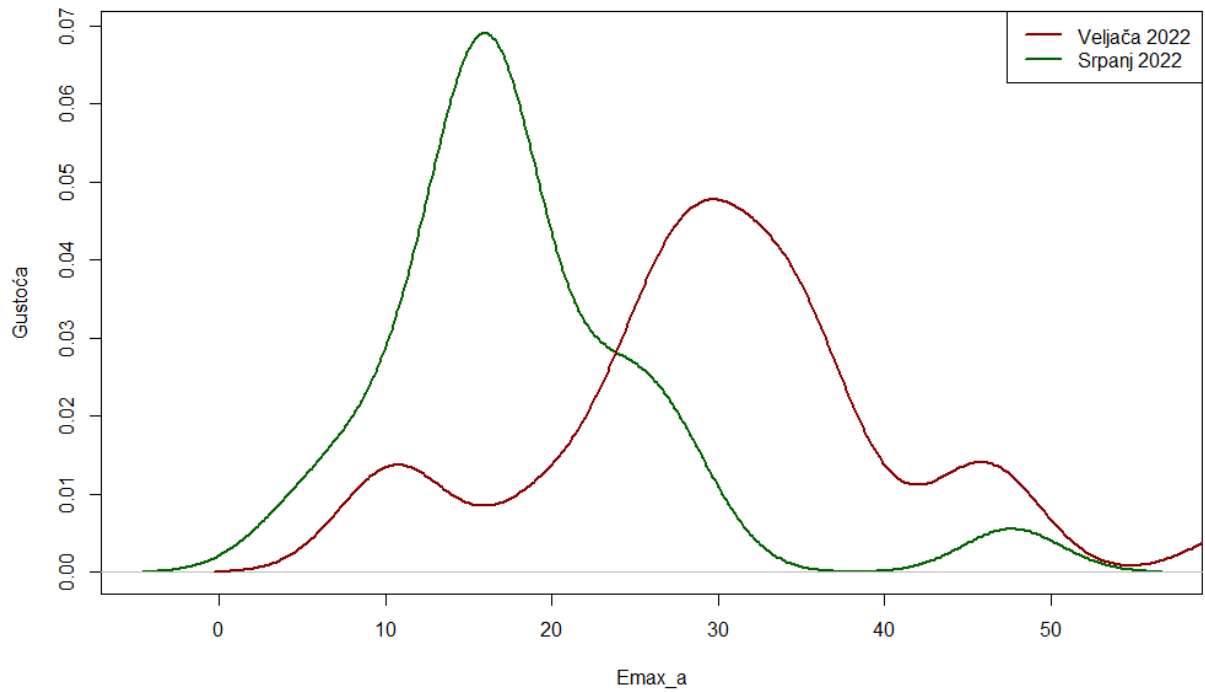
Slika 38. Statistička razdioba indeksa vijugavosti 2

Statistička razdioba usmjerene Emax-a analize



Slika 39. Statistička razdioba Emax-a analize

Statistička razdioba neusmjerene Emax-a analize



Slika 40. Statistička razdioba Emax-a analize

Tablica 6.1. kvantili indeksa vijugavosti 1

	10%	25%	50%	75%	90%
Veljača	7.856995	9.219761	9.925617	11.034163	14.310254
Srpanj	10.54906	11.35183	13.23184	14.17682	16.75790

Tablica 6.2. kvantili indeksa vijugavosti 2

	10%	25%	50%	75%	90%
Veljača	5.173999	6.162585	6.405393	7.072708	9.180412
Srpanj	6.669700	7.323256	8.488796	8.867732	10.702493

Tablica 6.3. kvantili Emax-a usmjeren

	10%	25%	50%	75%	90%
Veljača	5.213812	8.805781	10.684590	11.900171	12.660267
Srpanj	2.769610	3.202625	4.236552	7.684039	9.112412

Tablica 6.4. kvantili Emax-a neusmjeren

	10%	25%	50%	75%	90%
Veljača	15.57004	25.14643	30.35265	35.36575	45.26852
Srpanj	10.02361	14.58860	16.55604	21.72676	26.13385

Tablica 6.5. kvantili Emax-b usmjeren

	10%	25%	50%	75%	90%
Veljača	0.008624601	0.012742569	0.016525604	0.018163544	0.019781905
Srpanj	0.004732931	0.005435069	0.007248348	0.012119479	0.014364584

Tablica 6.6. kvantili Emax-b neusmjeren

	10%	25%	50%	75%	90%
Veljača	0.02324770	0.03932296	0.04800891	0.05184095	0.07404471
Srpanj	0.01659459	0.02453901	0.02678595	0.03635453	0.04393913

Tablica 6.7. kvantili fraktalne dimenzije

	10%	25%	50%	75%	90%
Veljača	1.003502	1.004040	1.005094	1.007237	1.016670
Srpanj	1.003059	1.005540	1.009694	1.030077	1.033179

Tablica 6.8. kvantili standardne devijacije promjene smjera kretanja

	10%	25%	50%	75%	90%
Veljača	70.93783	86.68431	96.36860	105.34797	155.88603
Srpanj	103.0349	115.5170	141.5421	157.1346	212.1119

Tablica 6.9. kvantili medijana promjene smjera kretanja

	10%	25%	50%	75%	90%
Veljača	383.5205	425.2836	469.0863	524.7611	659.1200
Srpanj	517.3251	551.4057	632.9597	683.2690	808.2191

7. LITERATURA

- [1] Number of Flights Worldwide in 2022/2023: Passenger Traffic, Behaviors, and Revenue, s interneta, <https://financesonline.com/number-of-flights-worldwide/>, 5. srpnja 2022.
- [2] Data Tools provided by the OpenSky Community, s interneta, <https://opensky-network.org/data/data-tools>, 5. srpnja 2022.
- [3] R For Dummies, second edition book by Andrie de Vries and Joris Meys, ISBN: 978-1-119-05580-8
- [4] openSkies: Retrieval, Analysis and Visualization of Air Traffic Dana, s interneta, <https://cran.r-project.org/web/packages/openSkies/index.html>, 5. srpnja 2022.
- [5] trajr: An R package for characterisation of animal trajectories, s interneta, <https://onlinelibrary.wiley.com/doi/10.1111/eth.12739>, 5 srpnja 2022.
- [6] Computers in Physics 4, 669 (1990); doi: 10.1063/1.4822961, s interneta, <https://aip.scitation.org/doi/pdf/10.1063/1.4822961>, 5. srpnja 2022.
- [7] Benhamou, S. (2004). How to reliably estimate the tortuosity of an animal's path. Journal of Theoretical Biology. <https://doi.org/10.1016/j.jtbi.2004.03.016>, Dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/S0022519304001353>
- Cheung, A., Zhang, S., Stricker, C., & Srinivasan, M. V. (2007). Animal navigation: the difficulty of moving in a straight line. Biological Cybernetics, 97(1), 47-61. doi:10.1007/s00422-007-0158-0. Dostupno na: https://www.researchgate.net/publication/6312996_Animal_navigation_The_difficulty_of_moving_in_a_straight_line
- Batschelet, E. (1981). Circular statistics in biology. ACADEMIC PRESS, 111 FIFTH AVE., NEW YORK, NY 10003, 1981, 388.
- Bovet, P., & Benhamou, S. (1988). Spatial analysis of animals' movements using a correlated random walk model. Journal of Theoretical Biology, 131(4), 419-433. doi:10.1016/S0022-5193(88)80038-9
- Shamble, P. S., Hoy, R. R., Cohen, I., & Beatus, T. (2017). Walking like an ant: a quantitative and experimental approach to understanding locomotor mimicry in the jumping spider *Myrmarachne formicaria*. Proceedings of the Royal Society B: Biological Sciences, 284(1858). doi:10.1098/rspb.2017.0308

Bringing up OpenSky: A large-scale ADS-B sensor network for research Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, Matthias Wilhelm ACM/IEEE International Conference on Information Processing in Sensor Networks, April 2014

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Makar Domagoj (2022). Diplomski rad. doi: 10.6084/m9.figshare.21167368, Dostupno na: https://figshare.com/articles/software/Diplomski_rad_Domagoj_Makar/21167368

8. SAŽETAK

OpenSky Network baza je podataka koja se bavi prikupljanjem i spremanjem opažanja trajektorija letova zrakoplova. Ovaj rad, nadopunjuje postojeće načine pristupa OpenSky bazi podataka, dodajući funkcionalnosti filtracije putanja postavljajući željene zračne luke te vremenski interval. Nad dobivenim podacima vrši se analiza pomoću Trajr knjižice za analizu putanja životinja. Analitičko rješenje nadopunjeno je grafičkim sučeljem izrađenim pomoću alata Shiny. Korisnost dobivenog programskog rješenja testirano je traženjem karakterističnih obilježja nad putanjama zrakoplova na relaciji Zagreb – Muenchen u različitim mjesecima.

Ključne riječi: R, OpenSky Network, Shiny, Trajr, analiza putanja, avijacija

ABSTRACT

OpenSky Network is a database that collects and stores observations of aircraft flight trajectories. This work complements the existing ways of accessing the OpenSky database, adding the functionality of path filtering by setting the desired airports and time interval. The obtained data is analyzed using the Trajr library for the analysis of animal trajectories. The analytical solution is complemented by a graphical interface created using the Shiny package. The usefulness of the obtained software solution was tested by searching for characteristic features of the aircraft trajectories on the route Zagreb - Munich during different months.

Keywords: R, OpenSky Network, Shiny, Trajr, analysis of trajectories, aviation

9. KLJUČNE RIJEČI, OZNAKE I KRATICE

POJAM	ZNAČENJE
Library	Knjižica programskih naredbi te gotovih funkcija
Database	Baza podataka
Network	Mreža
R	Programski jezik za statističku analizu
Trajr	Programska knjižica za analizu trajektorija životinja u dvije dimenzije
Shiny	Paket za izradu grafičkih sučelja
ADS-B	Automatic Dependent Surveillance-Broadcast receiver, automatski ovisni prijemnici za nadzor i emitiranje
FLARM	Flight alarm
Mode-S	Kodovi za nadzor i komunikacijske sustave letjelica
API	Application programming interface, sučelje za programiranje aplikacija
REST	Representation state transfer
Python	Programski jezik
Java	Programski jezik
Impala Shell	Alat za izradu i pristup bazama podataka

UTC	Coordinated Universal Time, univerzalno koordinirano vrijeme
Epoha	Moment u vremenu koji služi kao referentni moment
JSON	JavaScript Object Notation, format za razmjenu podataka
WGS-84	World Geodetic System, koordinatni sustav
GNU	GNU's Not Unix!, skup programa otvorenog tipa
Data frame	Podatkovni okvir
CRAN	The Comprehensive R Archive Network
R6	Klasa programske knjižice openSkies, sadrži informacije o letjelici, letu, ruti, vektorima stanja te setu vektora stanja
LEBL	Oznaka za zračnu luku Josep Tarradellas Barcelona
EDDM	ICAO oznaka zračne luke Muenchen (Munich International Airport – Franz Josef Strauss), više na: https://www.world-airport-codes.com/germany/franz-josef-strauss-5005.html
LDZA	ICAO oznaka zračne luke Franjo Tuđman Zagreb, više na: https://www.world-airport-codes.com/croatia/zracna-luka-zagreb-7829.html