

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andraž Krašovec

**Prototip sistema za zaznavanje trkov  
plovil**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Tomaž Curk

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Z elektronskim nadzorom izposojenega plovila lahko lastnik spremlja stanje plovila, hitro ugotovi morebitne trke in ustrezno ukrepa. Razvite sistem za zaznavanje trkov plovil, ki temelji na podatkih o pospeškometrih, lokaciji ter hitrosti plovila, vremenskih podatkih in drugih podatkih. Ker so trki sorazmerno redek dogodek, preučite možnosti uporabe metod odkrivanja osamelcev za zaznavanje trkov. Sistem ovrednotite na realnih podatkih in poročajte o uspešnosti zgrajenih modelov.



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Orodja in tehnologije</b>	<b>3</b>
2.1	Programski jezik R . . . . .	3
2.2	Razvojno okolje RStudio . . . . .	7
2.3	Nadzor nad različicami z uporabo platforme <i>gitlab.com</i> . . . . .	7
2.4	Program netcdfall . . . . .	7
<b>3</b>	<b>Podatki</b>	<b>9</b>
3.1	Podatki o plovilih . . . . .	9
3.2	Podatki o vremenu . . . . .	11
3.3	Združevanje podatkov . . . . .	12
3.4	Priprava končne zbirke . . . . .	13
<b>4</b>	<b>Analiza in rezultati</b>	<b>17</b>
4.1	Analiza atributa <i>Shock amplitude</i> . . . . .	17
4.2	Analiza pospeškov po oseh . . . . .	28
4.3	Analiza <i>Shock count</i> . . . . .	33
4.4	Rezultati . . . . .	38

<b>5 Zaključek</b>	<b>41</b>
5.1 Nadaljnje delo . . . . .	42
<b>Literatura</b>	<b>43</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>CSV</b>	comma-separated values	z vejico ločene vrednosti
<b>GRIB</b>	gridded binary	mreža binarnih vrednosti
<b>netCDF</b>	network common data form	skupni omrežni zapis podatkov
<b>IDE</b>	integrated development environment	integrirano razvojno okolje
<b>GPS</b>	global positioning system	globalni pozicijski sistem





# Povzetek

**Naslov:** Prototip sistema za zaznavanje trkov plovil

**Avtor:** Andraž Krašovec

Biti lastnik plovila ali flote plovil je stresna naloga, še posebej, če s plovilom upravlja nekdo drug ali pa je plovilo nenadzorovano privezано v marini. Zgodi se lahko marsikaj. Plovilo lahko trči v drugo plovilo, obalo ali drug objekt. Za oddaljen nadzor plovil obstaja mnogo naprav, ki so zmožne lastniku posredovati marsikatero informacijo. Vendar taka, ki bi omogočala zaznavo trka, še ni ponujena na tržišču. V ta namen se v zadnjem obdobju v naprave vgrajujejo pospeškometri. Zgolj podatki, ki jih ti senzorji beležijo, niso prav koristni, dokler se jih ne osmisli. Zato je cilj diplomske naloge osmisлити pridobljene podatke. Ker trki na morju niso prav pogosti, je večina podatkov, ki se lahko kvalificirajo kot trk, lažno pozitivnih. Za boljše razumevanje smo podatke iz senzorjev povezali s podatki o vremenu, poiskali korelacije med meritvami pospeškometra in ostalimi atributi ter klasificirali vsako večje odstopanje vrednosti pospeškometra. Uspeli smo razložiti približno tri četrtine takih točk in tako pripraviti podatke na nadaljnjo obdelavo, najverjetneje z metodami strojnega učenja.

**Ključne besede:** plovilo, osamelci, zbirka podatkov, pospešek, zaznavanje trkov.



# Abstract

**Title:** Prototype of a vessel collision detection system

**Author:** Andraž Krašovec

Owning a vessel or a fleet can be stressful, especially if someone else is steering it or it is left unsupervised in the marina. Many things can go wrong and a collision with another vessel, the shore or other objects is possible. Of course there are many devices, created especially to monitor one's vessel that are capable of transmitting numerous different informations to the owner, but none of those are capable to detect collisions. Lately, accelerometers are being implemented into these devices, but the data they provide is not useful until it is processed properly. The goal of this thesis is to process vessel sensor data and classify it. Because collisions at sea are fairly rare, most of the data recognised as a collision is false positive. To better understand what exactly is going on with the vessel at any given moment, the accelerometer and all other vessel data is merged with weather data. After that, correlation between accelerometer data and any other given attribute is calculated. Outliers are then classified, according to the previous analysis. Roughly three quarters of all outliers were classified correctly and the data is prepared for further processing, most probably by applying machine learning algorithms.

**Keywords:** vessel, outliers, dataset, acceleration, collision detection.



# Poglavje 1

## Uvod

Pri izposoji različnih plovil je tista oseba, ki daje plovilo v najem, vedno v nelagodnem položaju. Nikoli namreč ne more biti povsem prepričana, kaj se dogaja z njegovim plovilom. Zato so v zadnjih letih postale priljubljene naprave za nadzor plovil na daljavo. Ker je področje še relativno mlado, se produktom nenehno dodajajo nove funkcionalnosti. Trenutno naprave omogočajo:

- prikaz lokacije v realnem času in dnevnik zgodovine lokacij,
- nadzor stanja napetosti akumulatorjev,
- nadzor stanja kalužne črpalke,
- zaznavanje nepooblaščenega vstopa v plovilo,
- možnost integracije senzorjev zunanjih proizvajalcev.

Nobena naprava na tržišču ne omogoča zaznavanja trka plovil, ki lahko predstavlja pomembno informacijo uporabniku. Trki plovil so sicer redek pojav. Po podatkih Hrvaškega Nacionalnega centra za iskanje in reševanje na morju so v letu 2016 na območju Hrvaške posredovali 477-krat [11]. Finančne posledice trkov so mnogokrat hude, v najhujšem primeru pa so ogrožena tudi človeška življenja.

Podjetje Sentinel d.o.o. [5], ki je tudi priskrbelo podatke za diplomsko nalogo, je pred kratkim v svoje naprave začelo vgrajevati pospeškometre - senzorje, ki merijo jakost pospeška v več smereh. Pričakovan končni rezultat je, da podatke, pridobljene s tega senzorja, povežemo z ostalimi relevantnimi podatki in tako določimo ali je prišlo do trka plovila z drugim plovilom, obalo ali drugim objektom.

Da bi to lahko zagotovili, je najprej potrebno podatke ustrezno pripraviti in osmisliti večje odklone pospeškov. Le tako lahko zagotovimo, da je naš končni produkt natančen in ne proži lažnih alarmov. Na kratko – zanima nas kateri izmed razpoložljivih podatkov so uporabni v našem primeru in nato s pomočjo teh podatkov razložiti točke, ki potencialno predstavljajo trk.

## Poglavje 2

# Orodja in tehnologije

Za vsako zastavljeno nalogo je velikega pomena izbira pravih orodij, saj so le-ta pogoj za učinkovito rešitev naloge. Zato je prvi korak, preden se delo sploh začne, opredeliti kakšen tip naloge bomo reševali, poiskati več ustreznih orodij in nato izbrati najprimernejšega.

V tem poglavju bomo predstavili in argumentirali izbiro programskega jezika in razvojnega okolja za izbrani jezik, našteji in opisali uporabljene razširitve v obliki programskih paketov ter omenili še ostala uporabljena orodja in tehnologije.

### 2.1 Programski jezik R

Za vsak problem, ki ga rešujemo v računalništvu, obstaja mnogo programskih jezikov, ki jih lahko uporabimo za reševanje problema. Seveda pa niso vsi enako ustrezni in ob izbiri napačnega, lahko kaj hitro precej podaljšamo cikel razvoja programske rešitve.

Ker se naša naloga nahaja v domeni podatkovnega rudarjenja, smo izbiro zožili na tri ustrezne kandidate – Matlab, Python in R. Zaradi preteklih izkušenj in želje po poglobljanju znanja, smo se odločili za programski jezik R [9], ki je namenjen statističnemu računanju, analizam in prikazu podatkov, aplikacijam metod strojnega učenja in drugih. Jezik R je odprtokodna imple-

mentacija programskega jezika S. Velika privlačnost jezika R je repozitorij z velikim naborom paketov, v času pisanja je le-teh 11.177, ki močno povečajo nabor funkcionalnosti programskega jezika. Sledi pregled nekaj glavnih paketov, ki smo jih uporabili pri naši nalogi.

### 2.1.1 Risanje grafov z uporabo paketa ggplot2

Paket ggplot2 [12] je namenjen načrtovanju najrazličnejših grafov. Seveda take funkcije obstajajo že v samem jedru jezika R, vendar so le-te precej omejene, izris je monoton in urejanje naprednejših nastavitev, na primer urejanje legende, je zamudno in kompleksno. Ker smo se pri delu marsikdaj oprli na izris in primerjavo različnih atributov, bi brez tega paketa samo delo trajalo precej dlje. Najpogosteje uporabljene funkcije:

- `ggplot` – funkcija, ki ustvari temelje za izris, podamo ji podatke za prikaz in estetske direktive. Je osnovna funkcija, kateri pripnemo vse ostale funkcije za kreacijo končnega izrisa.
- `geom_histogram` – funkcija, ki iz podanih atributov ustvari histogram.
- `geom_point` – funkcija, ki iz podanih koordinat izriše točke v ravnini ali prostoru.
- `labs` – funkcija, ki ureja oznake osi, legendo, naslov in pripis vasakega grafa.

Paket je uporaben tudi v kombinaciji z drugimi paketi, saj lahko elemente grafike izrisujemo tudi na druge objekte, ne le zgolj na ggplot. Tak primer bomo predstavili v naslednjem paketu.

### 2.1.2 Risanje zemljevidov s paketom ggmap

Kot izdaja že samo ime paketa, je ggmap [4] tesno povezan s paketom ggplot2. Je paket za nalaganje zemljevidov z različnih spletnih mest, ki ponujajo tovrstne storitve (Google Maps, Stamen Maps, OpenStreetMap). V naši nalogi smo se omejili na zemljevide podjetja Google.



Za izris podatkov na zemljevidih je mogoče uporabiti tudi funkcije paketa `ggmap`, vendar smo se zaradi doslednosti odločili tudi pri tovrstnem izrisu uporabiti funkcije paketa `ggplot2`. Tako smo iz paketa `ggmap` uporabili le 2 funkciji:

- `make_bbox` – funkcija, ki na podlagi množice geografskih koordinat izračuna obsegajoče območje ali t.i. „bounding box“ in je eden izmed vhodnih parametrov naslednje funkcije.
- `qmap` – funkcija, oziroma ovojnica, ki v sebi skriva dve funkciji, `get_map` in `ggmap`. Prva poskrbi za pridobitev zemljevida z interneta, druga pa je zadolžena za izris pridobljenega. Vhodni parameter je lahko zgoraj omenjeni bounding box, ime kraja, ali pa kar geografske koordinate. Rezultat te funkcije je objekt, ki ga nato lahko kombiniramo s funkcijami paketa `ggplot2`.

### 2.1.3 Paket `data.table`

Paket `data.table` [1] nadomešča že vgrajeno podatkovno strukturo `data.frame`. Omogoča iste funkcionalnosti kot omenjena podatkovna struktura, v večini primerov tudi z enako sintakso. Kjer se močno razlikujeta je v hitrosti delovanja. Za prototipiranje na manjših podatkovnih zbirkah in tudi malo večjih projektih, kjer manipulacije s podatki ni ali pa je je izredno malo, `data.frame` več kot zadostuje in je boljša izbira, saj je že vgrajen v samem jedru jezika R in je na trenutke sintaktično bolj prijazen, sploh do razvijalcev, ki se z jezikom R srečujejo prvič.

Za kakršnekoli napredne ali časovno zahtevnejše operacije pa močno prevlada `data.table`. Za kratek primer vzemimo nalaganje podatkov v obliki datoteke CSV z 215.339 vrsticami:

```
> system.time(read.csv("firstclass_may.csv"))
  user  system elapsed
 5.72   0.12   5.84
```

```
> system.time(fread("firstclass_may.csv"))
  user  system elapsed
 1.15    0.02    1.17
```

Vidimo lahko, da gre za skoraj 5-kratno pohitritev. Ta trend drži tudi pri ostalih operacijah paketa `data.table` in pohitritev lahko naraste vse do faktorja 10, kar je bilo opaziti tudi pri združevanju podatkov, opisanem v podglavju 3.3.

### 2.1.4 Paket `ncdf4`

Paket `ncdf4` [8] omogoča manipulacijo z datotekami tipa `netCDF` [2]. Vremenski podatki po navadi zapisani v datoteki tipa `GRIB` [7], to pa lahko pretvorimo v tip `netCDF`. Za `netCDF` obstaja paket, ki omogoča lažje in učinkovitejše branje podatkov (v jeziku R ni ustreznega paketa za delo z datotekami `GRIB`).

Kljub vsemu je še vedno potrebno zavihati rokave in pretvorba datoteke `netCDF` v `data.frame` in posledično `data.table` ni trivialna. Je pa močno olajšana in pohitrena s tem paketom. Najpogosteje uporabljene funkcije so:

- `nc_open` – funkcija, ki naloži datoteko tipa `netCDF` v pomnilnik in omogoči branje in urejanje te datoteke.
- `ncvar_get` – funkcija, ki iz datoteke naloži vrednosti podanega atributa.
- `nc_close` – funkcija, ki datoteko „zapre“ in jo naloži nazaj v trajno shrambo.

Ker so imena istih atributov v različnih datotekah različna, je za delovanje teh funkcij potrebna še uporaba mnogih drugih, da je program zmožen pravilno prebrati vse attribute, vendar to že presega okvirje tega poglavja.

## 2.2 Razvojno okolje RStudio

Ob namestitvi programskega jezika R na računalnik, se zraven naloži še osnovno integrirano razvojno okolje (IDE). Okolje RStudio [10] ponuja mnogo naprednejših opcij od osnovnega okolja, predvsem pa preglednejši uporabniški vmesnik in več informacij o okolju. Zato smo se odločili za uporabo slednjega.

Ključna funkcionalnost, ki jo ponuja Rstudio, je možnost pisanja „notebookov.“ To je oblika zapisa, ki omogoča pisanje besedila in kode v isti datoteki. Prav tako ponujajo možnost izvoza v datoteko tipa HTML, PDF in DOC. Pred izvozom je datoteko mogoče oblikovati s pomočjo označevalnega jezika, ravitega posebej za ta namen. Tako je lahko končni rezultat poročilo, ki vključuje tudi dele programske kode, ki jih avtor želi izpostaviti in tako olajša sporazumevanje med avtorjem in ostalimi vpletenimi.

## 2.3 Nadzor nad različicami z uporabo platforme *gitlab.com*

Platforma *gitlab.com* omogoča nadzor nad verzijami programske kode in olajša ekipno delo ter preprečuje izgubo podatkov, saj je v osnovi porazdeljen sistem, ki omogoča hrambo kode na oddaljeni lokaciji. Omogoča tudi obnovo stanja programske kode v katerokoli stanje v preteklosti.

## 2.4 Program *netcdfall*

Knjižnica [3], napisana v programskem jeziku *Java*, primarno omogoča delo z datotekami tipa netCDF. Omogoča tudi za nas bolj zanimivo pretvorbo med formati. Uporabili smo jo za pretvarjanje datotek tipa GRIB. Ker je pogonjanje javanskih programov in pretvarjanje vsake datoteke zamudno, smo poiskali rešitev, ki omogoča klicanje programa direktno iz programskega jezika R. Uporabili smo funkcijo `shell`, ki omogoča klicanje sistemskih funkcij preko terminala. Tako smo v jeziku R ustvarili funkcijo:

```
grib2netcdf <- function(grib, out) {  
  (shell(  
    paste(  
      "java -Xmx512m -classpath netcdfAll-4.6.10.jar  
      ucar.nc2.dataset.NetcdfDataset -in ",grib," -out ",  
      out, sep = "  
    ),  
    intern = TRUE  
  ))  
}
```

Funkcija sprejme dva parametra, prvi je ime datoteke GRIB, ki jo želimo pretvoriti, drugi pa je ime novonastale izhodne datoteke netCDF. Tako smo zagotovili možnost poteka celotnega programa v okviru okolja jezika R.

# Poglavje 3

## Podatki

Ker naprave nimajo senzorjev, ki bi omogočali merjenje vremenskih dejavnikov, delamo z dvema ločenima zbirkami podatkov – s podatki o stanju plovil in z vremenski podatki. V tem poglavju bomo natančneje pogledali obe zbirki in pa metode za združevanje obeh zbirk.

### 3.1 Podatki o plovilih

Zbirka podatkov o plovilih zajema podatke, ki jih pošilja naprava nameščena na plovilu. Zbirka je bila izvožena iz podatkovne baze v formatu CSV in vsebuje zapise desetih plovil med 14. in 17. majem 2017. Privzeto se vrednosti shranijo in pošljejo na strežnik vsakih 20 minut. V kolikor se vmes vklopi kateri izmed nastavljenih sprožilcev, npr., sprememba položaja, visok sunek sile, se vrednosti shranijo, pošljejo nemudoma in interval dvajsetih minut se ponastavi.

Podatki so predstavljeni v tabeli 3.1. Nekaj atributov zahteva dodatna pojasnila.

Atribut `created` ima podatkovni tip `POSIXct` - gre za razred v programskem jeziku R, ki predstavlja datum in čas. Notranji zapis predstavlja število sekund od začetka leta 1970 (časovni pas UTC), t.i. „Unix“ ali „Epoch“ time. Za lažje razumevanje se le-ta pred prikazom pretvori v vnaprej določen for-

atribut	podatkovni tip	opis
boat_id	int	identifikacijska številka plovila
created	POSIXct	datum meritve
st_x	double	zemljepisna dolžina
st_y	double	zemljepisna širina
speed	double	hitrost plovila
bearing	double	smer plovbe
battery_voltage	double	napetost akumulatorja
x	double	jakost pospeška po osi x
y	double	jakost pospeška po osi y
z	double	jakost pospeška po osi z
shock_amplitude	double	skupni pospešek
shock_count	double	večji pospeški na časovno enoto

Tabela 3.1: Struktura podatkovne zbirke o plovilih.

mat, po navadi je to „%Y-%m-%d %H:%M:%S“, kar v primeru notranje vrednosti 0, izpiše vrednost: 1970-01-01 00:00:00.

`Shock_amplitude` je atribut, ki ga naprava ne zabeleži, ampak izračuna na podlagi vrednosti jakosti pospeška v smeri osi x, y in z. Za izračun se uporablja formula za razdaljo med točko in koordinatnim izhodiščem:

$$\text{shock\_amplitude} = \sqrt{x^2 + y^2 + z^2} \quad (3.1)$$

`Shock_count` je naslednji atribut, ki ni preprosta vrednost, zabeležena s strani nekega senzorja. Je izračun kolikokrat na določeno časovno enoto je presežena vnaprej določena vrednost `shock_amplitude` in je normirana glede na časovni interval med dvema meritvama. Zavzema vrednost na intervalu med 0 in 6000.

## 3.2 Podatki o vremenu

Za razliko od podatkov o plovilih, so vremenski podatki pridobljeni iz zunanjega vira. Natančnost podatkov je omejena glede na časovni interval meritev in gostoto geografskih točk. Časovno se meritve izvajajo vsake tri ure, geografsko pa so točke meritev postavljene na vsakih 0,054 stopinje geografske širine in 0,04 stopinje geografske dolžine. Podatki so zapisani v formatu GRIB. Gre za standardiziran način zapisa meteoroloških podatkov.

Ker ne obstaja ustrezna knjižnica za uvoz podatkov v formatu GRIB za programski jezik R, oziroma je nismo uspeli najti, smo podatke najprej pretvorili v format netCDF. Zatem smo uporabili knjižnico „ncdf4“, ki omogoča branje podatkov v prej omenjenem zapisu. Podatki so predstavljeni v tabeli 3.2. Ponovno je potrebnih nekaj dodatnih pojasnil.

Atribut `time` predstavlja število ur od začetne meritve. Formata GRIB in netCDF, poleg samih podatkov vsebujeta tudi nekaj metapodatkov, med katerimi se nahaja tudi podatek o času začetka meritev, tako da je lahko potem časovni atribut predstavljen kot odmik v številu ur od tega datuma, kar poenostavi zapis v sami zbirki podatkov, predstavlja pa nekaj izzivov, opisanih spodaj.

Atributa `u-component` in `v-component` sta vektorja, ki opisujeta veter na določeni višini nad površjem. V našem primeru je ta višina 0. Vektorja sta vzporedna s tlemi. Pozitivna `u` komponenta predstavlja veter, ki piha proti vzhodu, pozitivna `v` komponenta pa veter, ki piha proti severu. Ti dve dejstvi sta ključnega pomena, če želimo izračunati hitrost in smer vetra, ki sta bistveno koristnejši za kasnejše delo kot zgoraj omenjeni komponenti. Formuli za izračun smeri in hitrosti vetra sta bili pridobljeni s spletne strani *Earth Observing Laboratory* [6]. Hitrost vetra izračunamo s formulo:

$$\text{hitrost\_vetra} = \sqrt{u^2 + v^2} \quad (3.2)$$

Smer vetra izračunamo s formulo:

$$\text{smer\_vetra} = \text{atan2}(-u, -v) * 180/\pi \quad (3.3)$$

atribut	podatkovni tip	opis
lat	double	zemljepisna širina
lon	double	zemljepisna dolžina
time	int	število ur od začetne meritve
u-component	double	u komponenta vetra
v-component	double	v komponenta vetra

Tabela 3.2: Struktura podatkovne zbirke vremenskih podatkov.

### 3.3 Združevanje podatkov

Za osnovno analizo dogajanja delovanja pospeškov na plovilu bi zadostovali že zajeti podatki o plovilu. Vremenski podatki razširijo razumevanje dogajanja na plovilu. Če jih želimo uporabiti, jih je potrebno združiti v zgolj eno zbirko podatkov, v nasprotnem primeru je nadaljnje delo precej oteženo.

Programski jezik R nam skozi različne knjižnice omogoča združevanje zbirk po izbranih atributih. Uporabili smo knjižnico `data.table`, ki omogoča združevanje zbirk zgolj z nastavitvijo ključev, brez klica posebnih funkcij in je daleč najhitrejša izmed ponujenih rešitev.

Zbirki je najprej potrebno ustrezno pripraviti. Določiti moramo attribute, po katerih podatkovni zbirki sploh želimo združiti. V našem primeru je to čas in geografske koordinate, saj gre za attribute, ki so skupni obema podatkovnima zbirkama in so tudi pomensko najbolj smiselni. S tem delo še ni končano, saj so izbrani atributi pri podatkih o plovilu zvezni, medtem ko so pri vremenskih podatkih diskretni. To dejstvo smo s pridom izkoristili in podatke o plovilu zgolj zaokrožili na najbližjo vrednost ustreznega atributa vremenskih podatkov. Tako smo izpolnili vse pogoje za združitev zbirk v eno združeno zbirko.

Zaradi ohranjanja natančnosti smo pred združevanjem shranili prvotne vrednosti atributov podatkov o plovilu, ki smo jih zatem zaokrožili. Tako smo lahko le-te ohranili tudi med združevanjem in jih nato uporabili kot *de facto* vrednosti v združeni podatkovni zbirki.



atribut	prvotna zbirka
boat_id	plovila
created	plovila
st_x	plovila
st_y	plovila
speed	plovila
bearing	plovila
battery_voltage	plovila
x	plovila
y	plovila
z	plovila
shock_amplitude	plovila
shock_count	plovila
u-component	vreme
v-component	vreme

Tabela 3.3: Struktura združene podatkovne zbirke.

Struktura združene podatkovne zbirke je predstavljena v tabeli 3.3. Sami podatki in njihovi tipi so ostali nespremenjeni.

### 3.4 Priprava končne zbirke

Za lažje razumevanje podatkov je določene attribute potrebno še ustrezno urediti, ustvariti nekaj novih in posledično kakšnega tudi odstraniti.

Že zgoraj omenjeni sta komponenti vetra ( $u$ ,  $v$ ), ki pomensko ne doprinesejo veliko. Bistveno koristnejša sta podatka o smeri in hitrosti vetra. Zato smo ustvarili dva nova atributa – smer vetra in hitrost vetra (`wind_direction` in `wind_speed`). Izračunali smo ju s pomočjo formul 3.2 in 3.3. S tem sta postala atributa `u-component` in `v-component` odvečna in smo ju zato odstranili.

Za dobro razumevanje dogajanja na plovilu, predvsem na jadrnica, je zelo pomemben še podatek, ki ga lahko izračunamo iz predstavljenih atributov: kot vetra glede na smer plovbe plovila. Izračun tega atributa je relativno preprost. Potrebujemo atributa *bearing* in novonastali *wind\_direction*, ki zavzemata vrednosti na intervalu med -180 in 180 stopinj. Domensko znanje o jadraniu smo pridobili s spleta [13]. Najprej izračunamo kot:

$$\alpha = bearing - wind\_direction \quad (3.4)$$

Ker nas predvsem zanima v kateri del plovila piha veter, točno vrednost nadomestimo z enim izmed štirih možnih razredov:

- PREMEC  $|\alpha| \leq 70^\circ$
- LEVI BOK  $-70^\circ < \alpha \leq -110^\circ$
- DESNI BOK  $70^\circ < \alpha \leq 110^\circ$
- KRMA  $|\alpha| > 110^\circ$

Novonastali atribut poimenujemo „Sailpoint“ in ga dodamo končni podatkovni zbirki, medtem ko nas smer vetra ne zanima več in jo zato odstranimo. Podatki so tako pripravljene na nadaljnje delo in so prikazani v tabeli 3.4.

atribut	podatkovni tip	opis
boat_id	int	identifikacijska številka plovila
created	POSIXct	datum meritve
st_x	double	zemljepisna dolžina
st_y	double	zemljepisna širina
speed	double	hitrost plovbe
bearing	double	smer plovbe
battery_voltage	double	napetost akumulatorja
x	double	jakost pospeška po osi x
y	double	jakost pospeška po osi y
z	double	jakost pospešek po osi z
shock_amplitude	double	skupni pospešek
shock_count	double	večji pospeški na časovno enoto
wind_speed	double	hitrost vetra
sailpoint	enum	kot vetra glede na smer plovbe

Tabela 3.4: Struktura končne podatkovne zbirke.



# Poglavje 4

## Analiza in rezultati

Kot opisano v uvodu, je naša naloga analizirati točke, ki potencialno predstavljajo trk plovila z drugim objektom, naj gre za drugo plovilo, morsko dno ali drug objekt, ki lahko poškoduje naše plovilo. Atributov v podatkih, ki so posledica zaznavanja pospeškometra, je več: skupni pospešek (*Shock amplitude*), jakosti pospeškov po oseh ( $x, y, z$ ) in število večjih sunkov na časovno enoto (*Shock count*). Zato je tudi analiza ustrezno tematsko razdeljena na tri ločene sklope, saj so atributi, kljub istemu izvoru, pomensko med sabo različni.

Točke, ki nas pri analizi zanimajo, so odkloni v jakosti pospeškov, večji od  $1g$ . Ta prag je bil določen s pomočjo strokovnjakov z domenskim znanjem s področja plovil in jadranja, in se je med kasnejšim delom izkazal za ustreznega. V podatkih je vrednost gravitacijskega pospeška  $1g$  predstavljena z vrednostjo 1000. To so tiste točke, ki potencialno predstavljajo trk, zato je pomembno da vsak tak odklon razumemo in ga znamo razložiti.

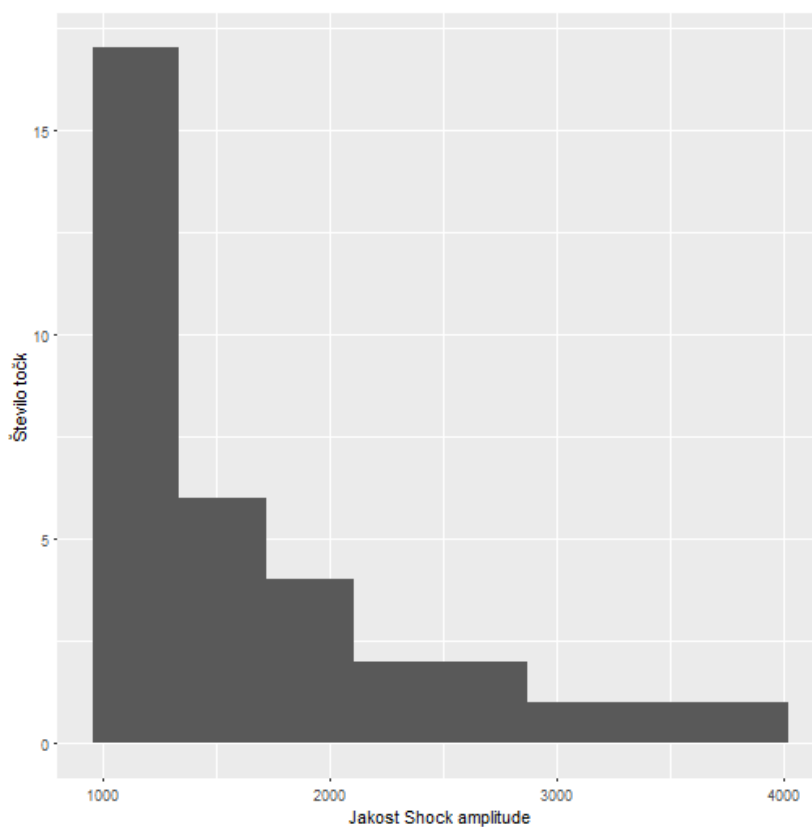
### 4.1 Analiza atributa *Shock amplitude*

Začeli bomo z atributom *Shock amplitude*, saj je le-ta predstavlja najnižjo stopnjo kompleksnosti in omogoča najlažjo analizo. Za ta del analize smo izločili attribute jakosti pospeškov po oseh in *Shock count*, saj so le-ti vsak

predmet svojega poglavja.

### 4.1.1 Porazdelitev osamelcev

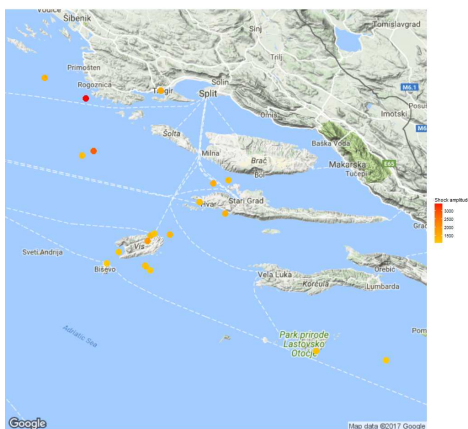
Histogram na sliki 4.1 predstavlja porazdelitev vseh točk, ki presegajo vrednost  $1g$ . Vidimo lahko, da se razpon vrednosti giblje od 1000 ( $1g$ ) do 4000 ( $4g$ ) in da število točk z naraščanjem *Shock amplitude*, upada. Zavoljo jedrnatosti bodo te točke od tu naprej imenovane osamelci.



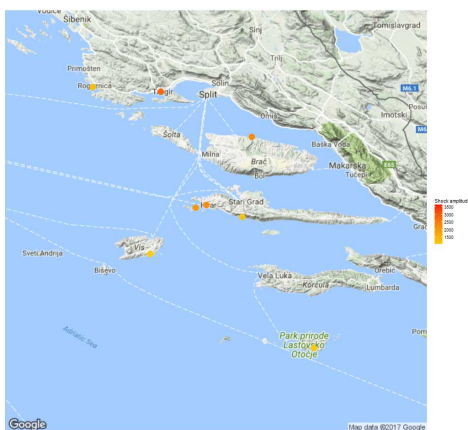
Slika 4.1: Porazdelitev osamelcev.

### 4.1.2 Geografski prikaz točk

Zemljevidi v tem podpoglavju prikazujejo geografsko lokacijo osamelcev. Osamelci so ločeni na dogodke med plovbo (slika 4.2) in mirovanje (slika 4.3). Ker gre za osamelce z vseh desetih plovil, lahko ugotovimo, da so poleg plutja ob istem času, plula tudi na približno enakem območju.



Slika 4.2: Osamelci med plovbo.



Slika 4.3: Osamelci v mirovanju.

### 4.1.3 Sailpoint

Atribut, ki smo ga ustvarili naknadno in je opisan v podpoglavju 3.4, je pomemben predvsem pri analizi osamelcev, ki nastanejo na jadrnicah, takrat ko le-te plujejo na veter. Takrat imajo jadrnice razprta jadra in so zato bistveno bolj dovzetne na kot delovanja vetra na plovilo, saj se površina upora silam vetra poveča za nekajkrat. Iz podatkov, ki sicer niso del te naloge, smo razbrali, da so vsa obravnavana plovila jadrnice, kar daje odlično izhodišče za analizo atributa *Sailpoint*. Ker je bil veter v dneh zbiranja podatkov relativno neugoden za jadrnanje, so bila jadra večino časa pospravljena in so pluli na motorni pogon.

Histogram na sliki 4.4 predstavlja število točk za vsak razred *Sailpoint*. Po pričakovanjih se največ osamelcev pripeti med plovbo z vetrom ali v veter, medtem ko je bistveno manj močnejših delovanj sil v bok med vetrom.

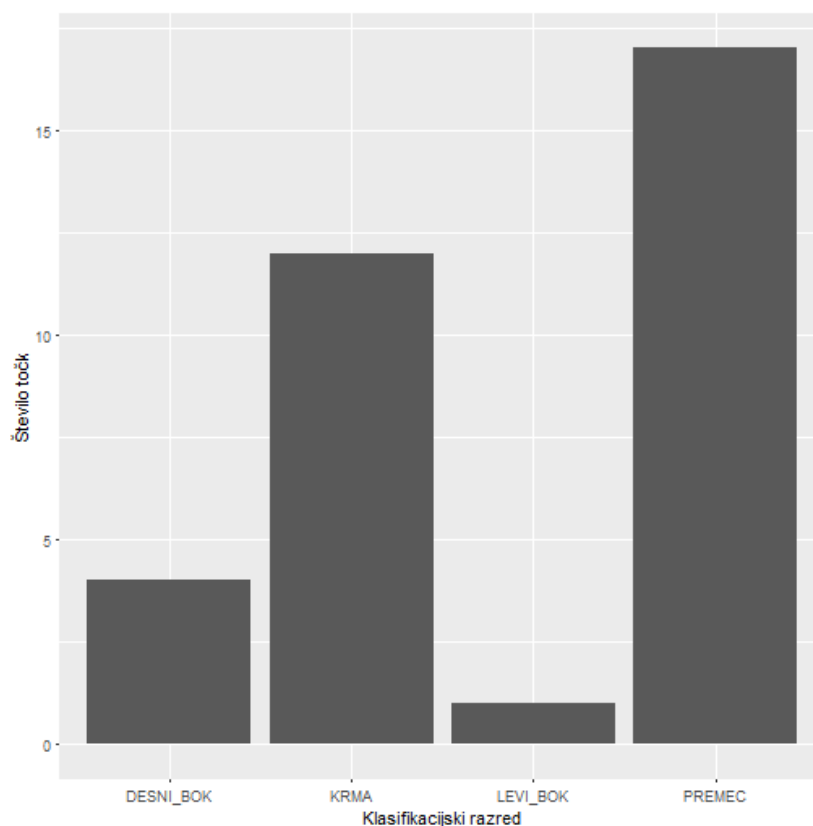
### Sprememba smeri pri vetru v krmo

V kolikor jadrnica pri vetru v krmo spremeni smer, pogosto pride do zanihanja jadra z ene strani na drugo, kar povzroči sunek in ga lahko zaznamo kot osamelec. Med vsemi osamelci, pri katerih je bila vrednost atributa *Sailpoint* „KRMA“, smo našli primer, kjer lahko z zagotovostjo trdimo, da je prišlo do tega pojava. Prikazan je na sliki 4.5. Točka, ki je obarvana rdeče, predstavlja osamelec in korelacija med osamelcem in spremembo smeri plovbe je očitna.

### 4.1.4 Hitrost vetra

Pričakovali bi da je število osamelcev in jakost sunkov premo sorazmerna s hitrostjo vetra. Višja hitrost pomeni višje valove in druge dejavnike, ki vplivajo na povečanje velikosti sil, ki vplivajo na plovilo. Vendar že prvi graf, ki prikazuje število točk pri posamezni hitrosti vetra, pravi drugačno zgodbo. Porazdelitev točk je približno normalna (rdeča navpična črta predstavlja povprečno hitrost vetra), kar odstopa od pričakovanj in kaže na to, da hitrost vetra nima večjega vpliva na pojavljanje osamelcev (slika 4.6).





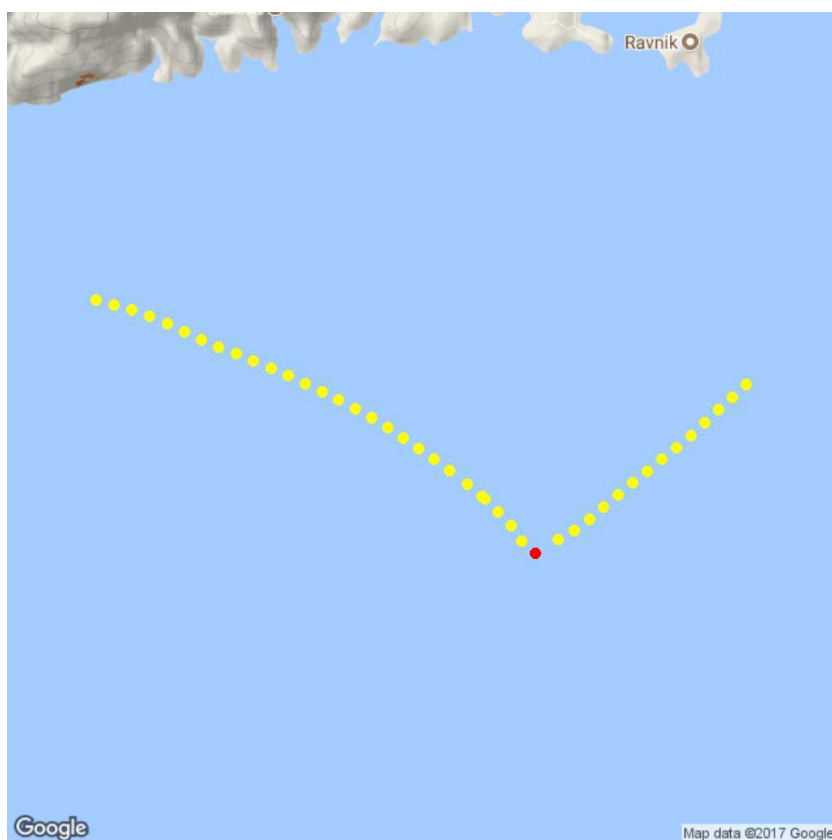
Slika 4.4: Osamelci klasificirani glede na Sailpoint.

### Normalizacija

Ker smo pri zgornjih grafih upoštevali zgolj zabeleženo hitrosti vetra pri osamelcih, nismo upoštevali celotne slike, brez katere težko izpeljemo ustrezne zaključke. Zato smo pred izločanjem osamelcev hitrost vetra normalizirali:

$$Wind\ speed = (Wind\ speed - \mu(Wind\ speed)) / \sigma(Wind\ speed) \quad (4.1)$$

Za potrditev zgornjih rezultatov, bi pričakovali, da bo večina osamelcev skoncentrirana okoli nič, ki zaradi normalizacije predstavlja povprečno hitrost vetra (rdeča navpična črta). Izkaže se, da le obstaja korelacija med hitrostjo vetra in osamelci, saj je povprečna hitrost vetra pri osamelcih precej višja (zelena navpična črta), kar pomeni, da osamelci v povprečju nastajajo pri



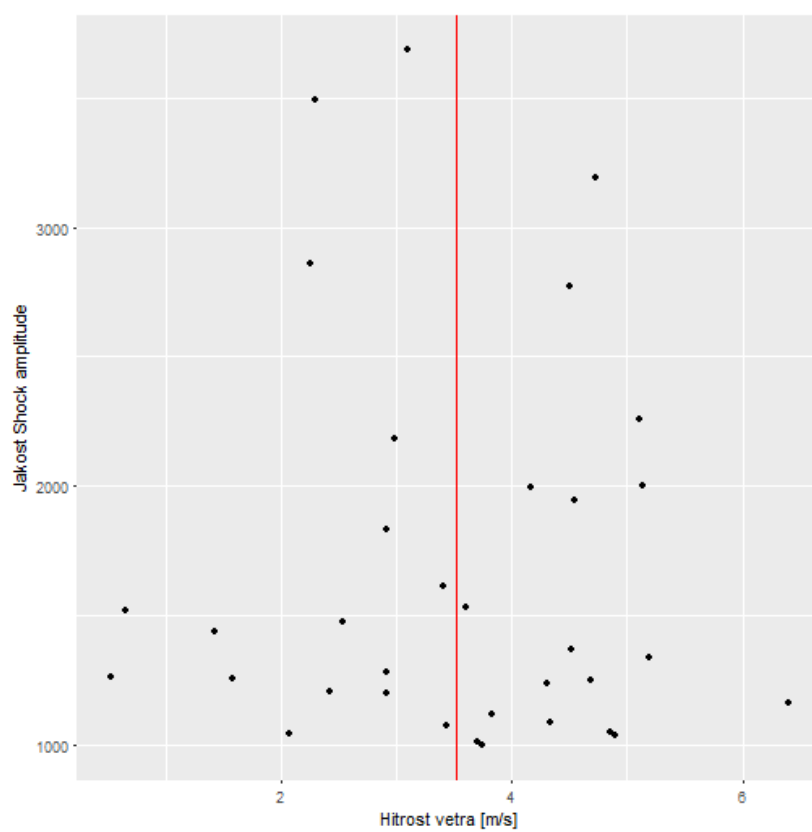
Slika 4.5: Osamelec med spremembo smeri pri vetru v krmo.

višji hitrosti vetra od povprečne. Graf je prikazan na sliki 4.7.

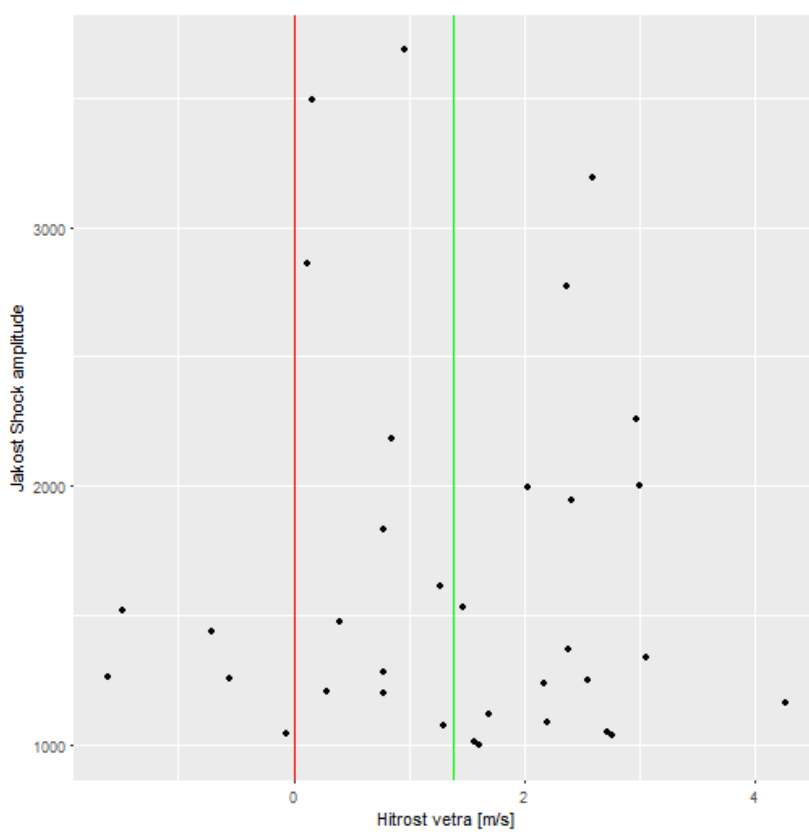
#### 4.1.5 Hitrost plovbe

Naslednji dejavnik, ki vpliva na *Shock amplitude*, je hitrost plovbe. Ker je hitrost plovbe v mirovanju 0 m/s, tukaj opazujemo zgolj točke, ki so bile zabeležene med samo plovbo (slika 4.8). Izkaže se, da pri višjih hitrostih nastaja več sunkov, najbolj izstopa dejstvo, da so vsi sunki, višji od 2g, nastali pri visokih hitrostih.

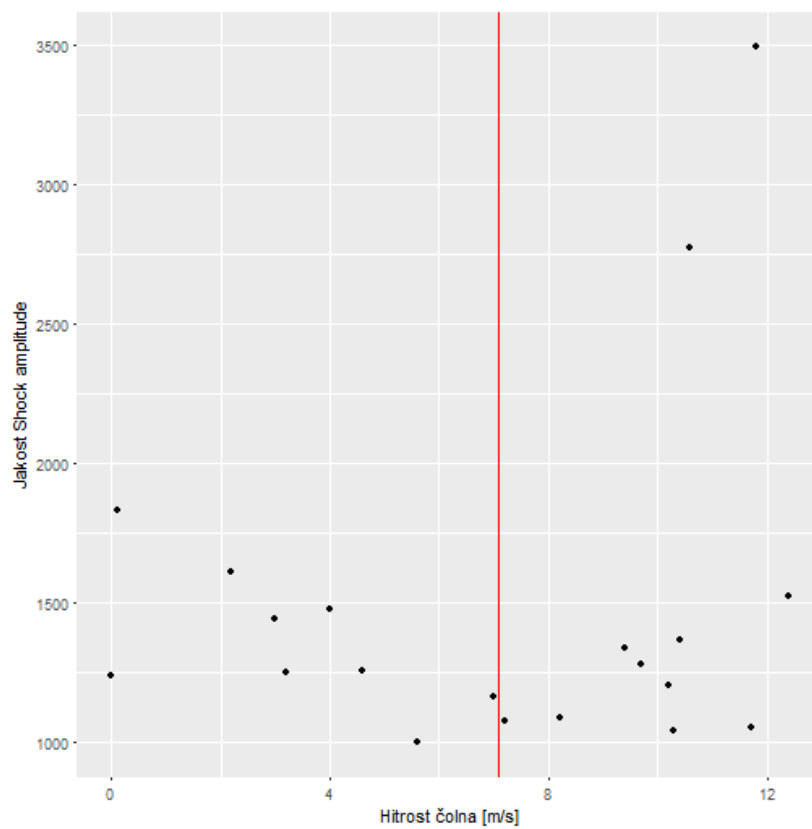
Vendar, ko zraven prikažemo še točke, ki so nastale v mirovanju, se naše domneve izrodijo, saj jakost sunkov med mirovanjem celo preseže jakost sunkov med plovbo, kar je prikazano na sliki 4.9.



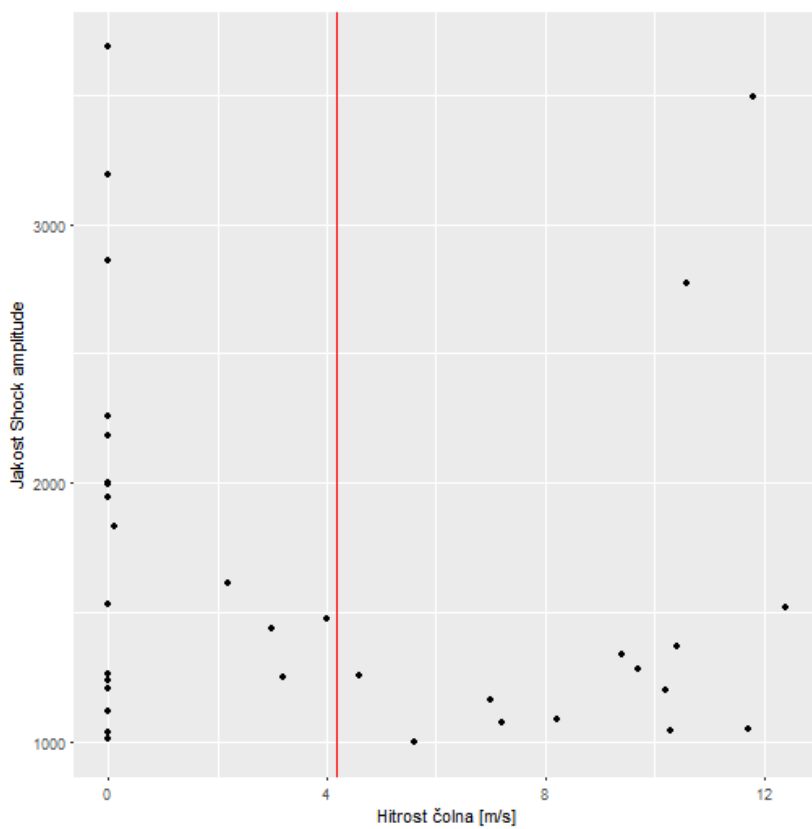
Slika 4.6: Osamelci glede na hitrost vetra.



Slika 4.7: Osamelci glede na normalizirano hitrost vetra.



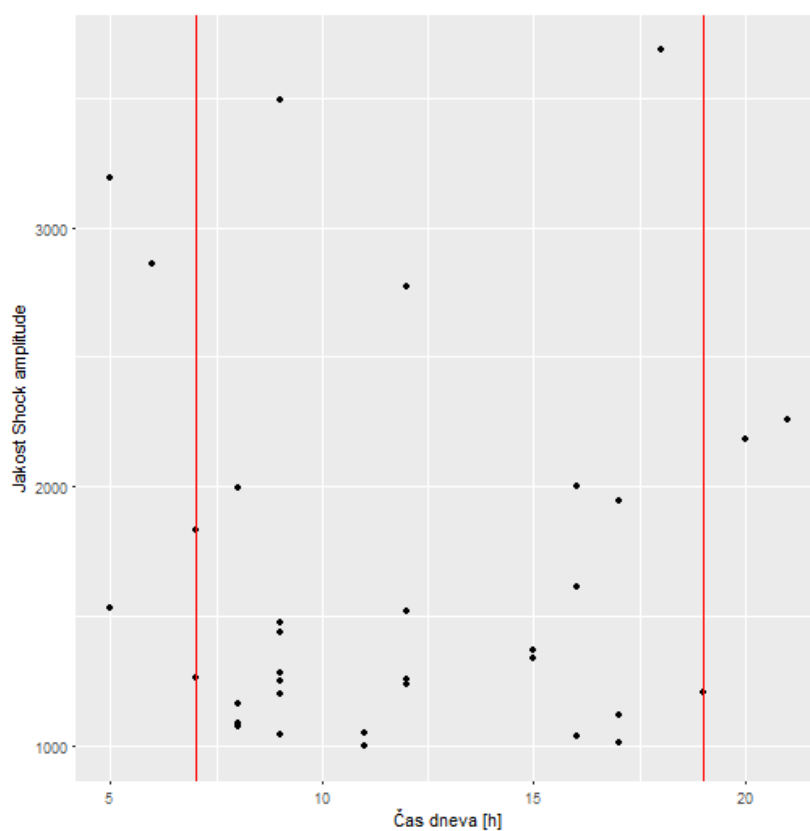
Slika 4.8: Osamelci glede na hitrost plovbe med plovbo.



Slika 4.9: Osamelci glede na hitrost plovbe.

### 4.1.6 Čas v dnevnu

Čas v dnevnu lahko vpliva na pojavitev posameznega osamelca, saj je morje zjutraj in zvečer mirnejše kot podnevi. Rdeči črti predstavljata dnevni interval med sedmo uro dopoldne in sedmo uro popoldne. Pokaže se trend, da veliko osamelcev nastane v času, ko se plovila odpravljajo na plovbo in prihajajo v pristanišče (slika 4.10).



Slika 4.10: Osamelci glede na čas v dnevnu.

### Izplutje in pristajanje

Vsa plovila, najsi gre za jadrnice ali motorna plovila, pri izhodu in vstopu v marino uporabljajo motor. Ob zagonu motorja pride pri vsakem plovilu

do močnega sunka, kar lahko povzroči osamelce. Spodnji grafi prikazujejo posamezna plovila skozi časovni interval več dni.

Najprej lahko opazimo, da je korelacija med vžigom motorja in začetkom ter koncem plovbe skoraj popolna, kar potrjuje, da vsa plovila v tej fazi plovbe plujejo na motor. Vžig motorja na grafu predstavlja dvig napetosti akumulatorja, saj se le-ta ob prižganem motorju polni. Tako lahko tudi določimo čas delovanja motorja, saj stanje visoke napetosti nastane zgolj v dveh primerih – v že zgoraj omenjenem in ko je plovilo privezano v marini in je akumulator priklopljen na zunanje napajanje.

Primeri, ko ob tem pride do nastanka osamelca ali vsaj višje vrednosti pospeška, niso tako pogosti in so tudi pri istem plovilu v razponu več dni precej nekonsistentni. Kljub temu je težko ovreči, da zagon motorja pri določenih plovilih občasno ne povzroči večjega odstopanja. Zavoljo jedrnatosti slika 4.11 predstavlja vrednosti enega plovila. Navpične črte predstavljajo začetek in konec plovbe, rdeče točke pa osamelce.

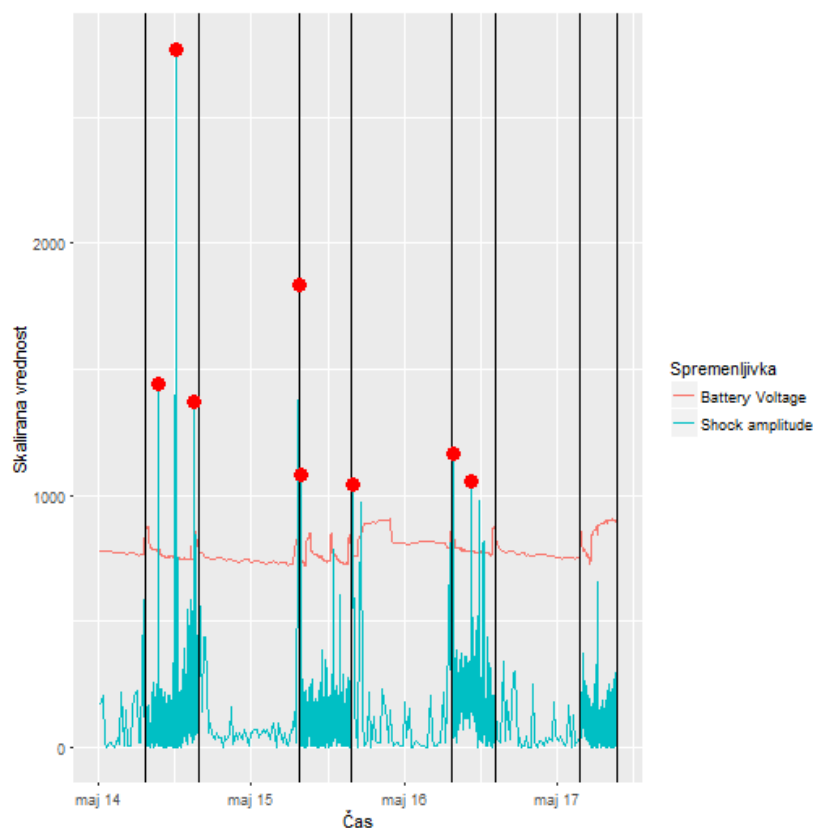
## 4.2 Analiza pospeškov po oseh

Podobno kot v podpoglavju 4.1, bomo tudi tu izločili attribute, ki sodijo v svoje poglavje. V tem primeru sta to atributa *Shock amplitude* in *Shock count*. Pospeški po oseh so tisti podatki, ki jih pospeškometer dejansko zabeleži in so temelj za atributa *Shock amplitude* in *Shock count*. Analiza teh pospeškov je kompleksnejša, saj operiramo s tremi ločenimi vrednostmi. Ker ugotovitve prejšnjega podpoglavja sovpadajo z ugotovitvami, ki smo jih odkrili tudi pri analizi vsake komponente posebej, bomo predstavili ugotovitve, ki so pri analizi *Shock amplitude* nemogoče.

### 4.2.1 Orientacija naprav

Pomembno dejstvo, ki se ga moramo zavedati je, da naprave na plovila nameščajo lastniki sami, oziroma zunanji izvajalci. Ti sicer prejmejo navodila kako napravo namestiti, vendar to ni zagotovilo, da je naprava nameščena



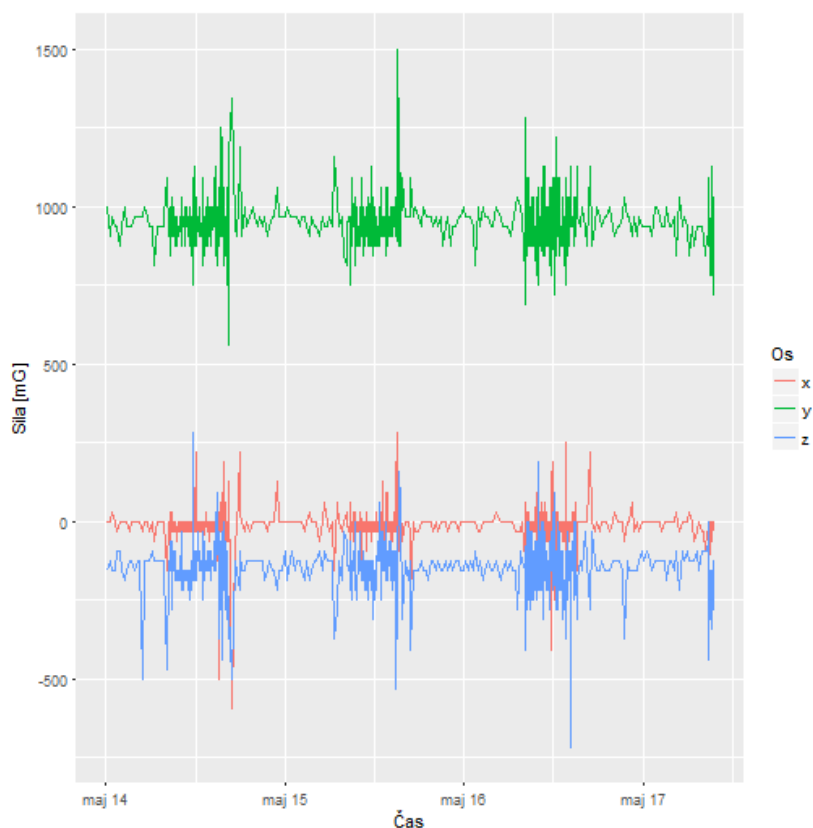


Slika 4.11: Osamelci glede na čas v dnevu, povezani z napetostjo baterije.

pravilno. Seveda obstaja nešteto različnih plovil in pričakovati, da bodo naprave povsod nameščene enako, je utopično. Tako se srečamo s prvo oviro pri analizi vsake komponente posebej – orientacija naprave.

Tu nam je v veliko pomoč gravitacija, saj pospeškometer ob mirovanju plovila beleži zgolj pospešek gravitacije, kar nam pove katera os, oziroma kombinacija osi, kaže navzdol. Poglejmo si delovanje pospeškov po oseh za dve različni napravi – slika 4.12 in slika 4.13.

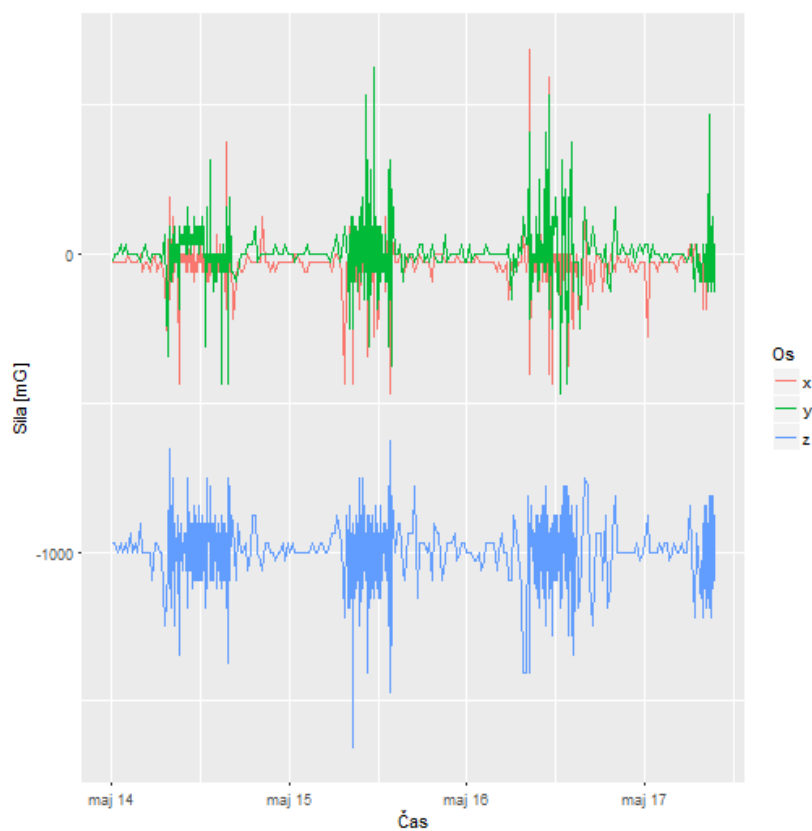
Po pregledu slik lahko hitro potrdimo naše domneve, kar pomeni da res naprave niso povsod nameščene enako. Če primerjamo dve različni napravi, lahko ugotovimo, da se vsaka komponenta giba okoli vrednosti 1000, 0 ali -1000. Iz tega se lahko naučimo katera komponenta je usmerjena navpično.



Slika 4.12: Delovanje pospeškov po oseh za napravo 1.

Pri sliki 4.12 je razvidno, da je navpična komponenta os  $y$  in da je usmerjena navzdol, saj njena vrednost niha okoli 1000 ( $1g$ ), do česar pride zaradi prej omenjenega gravitacijskega pospeška. Slika 4.13 prikazuje drugačno zgodbo. Komponenta, ki kaže navpično, je os  $z$  in je usmerjena navzgor, saj vrednost niha okoli  $-1000$  ( $-1g$ ).

Po natančnejšem pregledu vseh naprav, lahko ugotovimo, da sta zgoraj omenjeni orientaciji edini prisotni (za naprave, ki so zajete v analizi). Vendar nimamo nikakršnega podatka o orientaciji ostalih dveh osi, saj v mirovanju nihata okoli vrednosti 0.



Slika 4.13: Delovanje pospeškov po oseh za napravo 2.

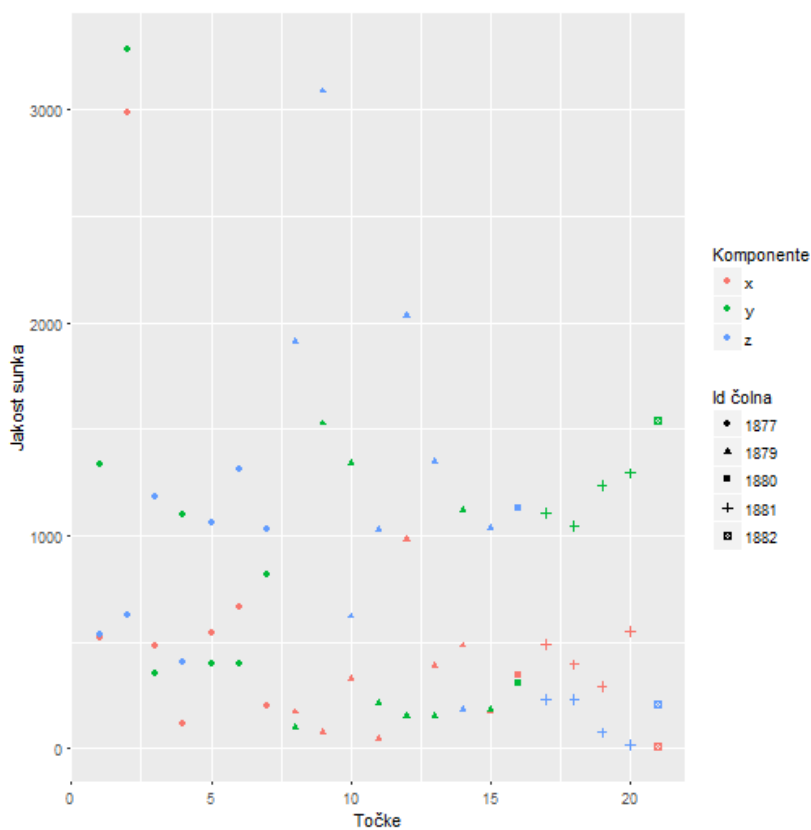
#### 4.2.2 Delitev na plovbo in mirovanje

Zavoljo lažje analize smo komponente obrnili tako, da je pri vseh napravah os  $z$  obrnjena v vertikalni smeri. To smo dosegli tako, da smo komponente, kjer je os  $y$  usmerjena vertikalno, zamaknili za eno mesto v desno:

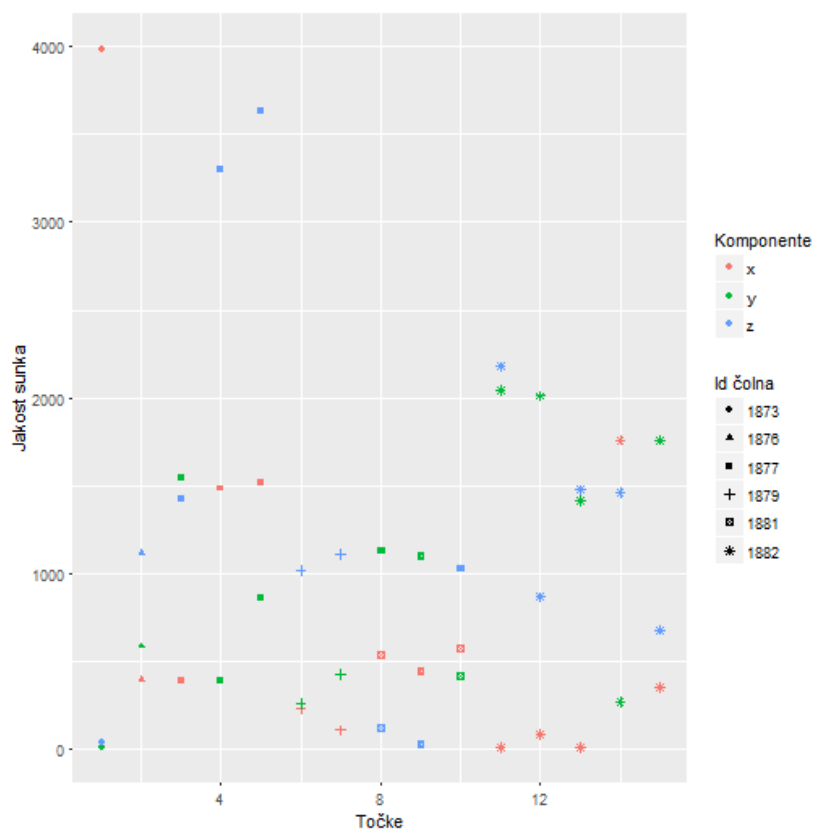
```
for(i in 1:nrow(osamelci)) {  
  if(osamelci[i, "navpicna_komponenta"] == "y") {  
    tmp1 <- osamelci[i, "x"]  
    tmp2 <- osamelci[i, "y"]  
    osamelci[i, "x"] <- osamelci[i, "z"]  
    osamelci[i, "y"] <- tmp1  
    osamelci[i, "z"] <- tmp2  
  }  
}
```

}  
}

Opazimo lahko, da med plovbo (slika 4.14), os  $x$  praktično nikoli ne preseže vrednosti  $1g$  in skoraj vse osamelce sprožita komponenti  $y$  in  $z$ . Medtem ko so vrednosti v mirovanju bolj naključne (slika 4.15). Poleg tega je razvidno tudi, da nikoli vse tri komponente ne presežejo meje  $1g$  naenkrat.



Slika 4.14: Osamelci po komponentah med plovbo.



Slika 4.15: Osamelci po komponentah med mirovanjem.

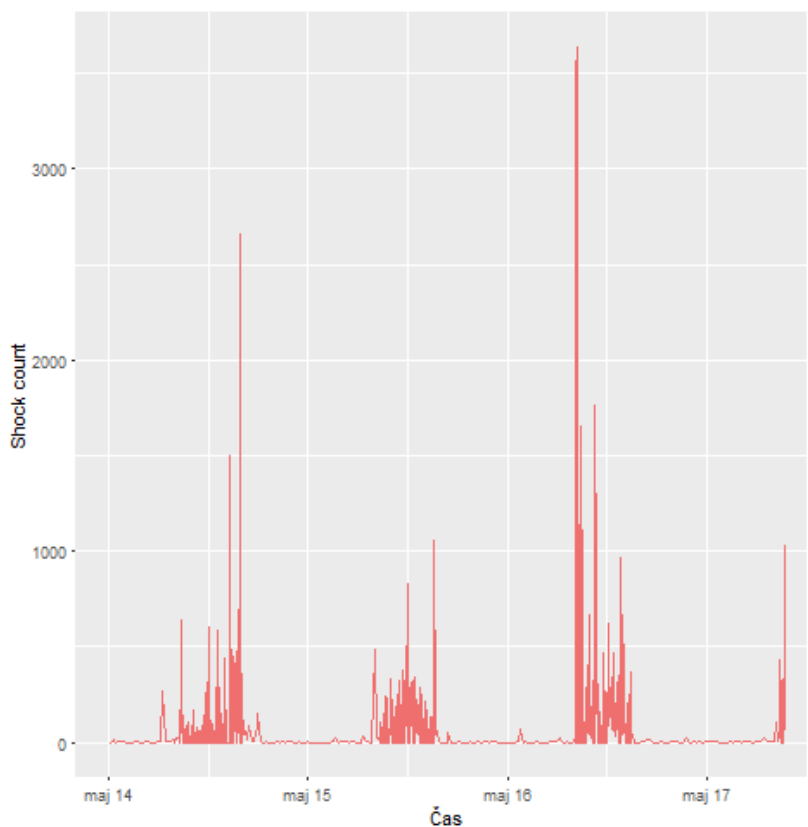
### 4.3 Analiza *Shock count*

Kot opisano v podpoglavju 3.1, podatek *Shock count* merimo tako, da beležimo kolikokrat na dano časovno enoto *Shock amplitude* preseže neko vrednost. Ta vrednost je vnaprej določena, statična in enaka pri vseh napravah. Vrednost, ki jo lahko zavzame *Shock count* leži na intervalu med 0 in 6000.

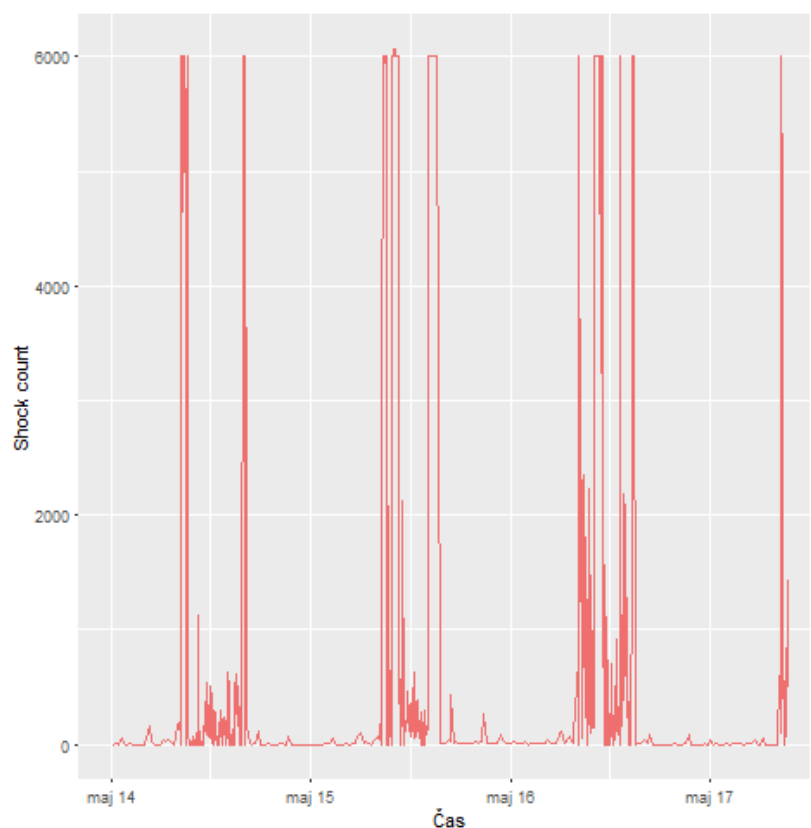
Za razliko od predhodnih dveh poglavij, je namen tukajšnje analize pregledati celotno zbirko podatkov in ne zgolj osamelcev.

### 4.3.1 Razlikovanje med napravami

Ker so pospeški, ki jih naprava zabeleži, odvisne od veliko dejavnikov, je pričakovano, da bodo vrednosti *Shock count* od naprave do naprave precej različne, glede na to, da je prag za vse naprave enak. Če pogledamo grafa (slika 4.16 in slika 4.17), lahko vidimo da je temu res tako, nekatere naprave nikoli ne dosežejo najvišje vrednosti, medtem ko druge med plovo redno zadevajo v zgornjo mejo, tako da bi bilo bolj smotrno prag prilagoditi vsaki posamezni napravi.



Slika 4.16: Vrednosti *Shock count* za napravo 1.



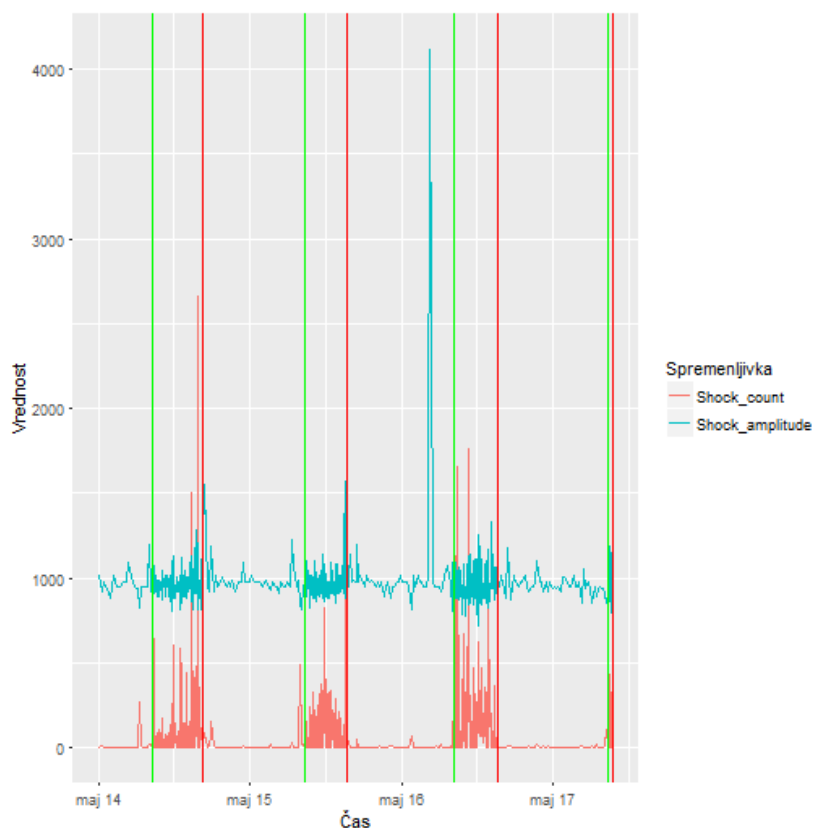
Slika 4.17: Vrednosti *Shock count* za napravo 2.

### 4.3.2 Primerjava plovbe in mirovanja

Ker vgrajena GPS naprava vsake toliko v mirovanju „pobezlja“ – beleži premikanje, kljub temu, da plovilo stoji na mestu, je za zmanjšanje beleženja neresničnih podatkov koristno, da zagotovimo alternativno možnost za preverjanje ali se plovilo dejansko premika ali je zasidrano v marini.

*Shock count* je primeren kandidat za takšno klasifikacijo, saj je v mirovanju vibracij bistveno manj, kot med plovbo, kar zmanjša število odzivov, višjih od postavljenega praga.

Graf na sliki 4.18 prikazuje vrednosti *Shock count* in *Shock amplitude*, zelena in rdeča navpična črta predstavljata začetek in konec plovbe, vmes so obdobja, ko je plovilo mirovalo.



Slika 4.18: Prikaz *Shock amplitude* in *Shock count* skozi čas.

Izpeljemo lahko več rezultatov:

- Potrdimo lahko korelacijo med *Shock count* in plovbo plovila, saj so vrednosti v mirovanju večinoma 0 ali zelo blizu te vrednosti, med plovbo pa se vrednosti precej dvignejo.
- *Shock count* in *Shock amplitude* nista neposredno povezana. Najlepši prikaz tega je pri sliki 4.18, kjer *Shock amplitude* vmes skoči na vrednost približno 4000, *Shock count* pa ostane nespremenjen.
- Težave povzročata le dva primera, ki se občasno pojavita:
  - vrednost *Shock count* se med plovbo umiri,



- občasni skok vrednosti *Shock count-a* med mirovanjem.

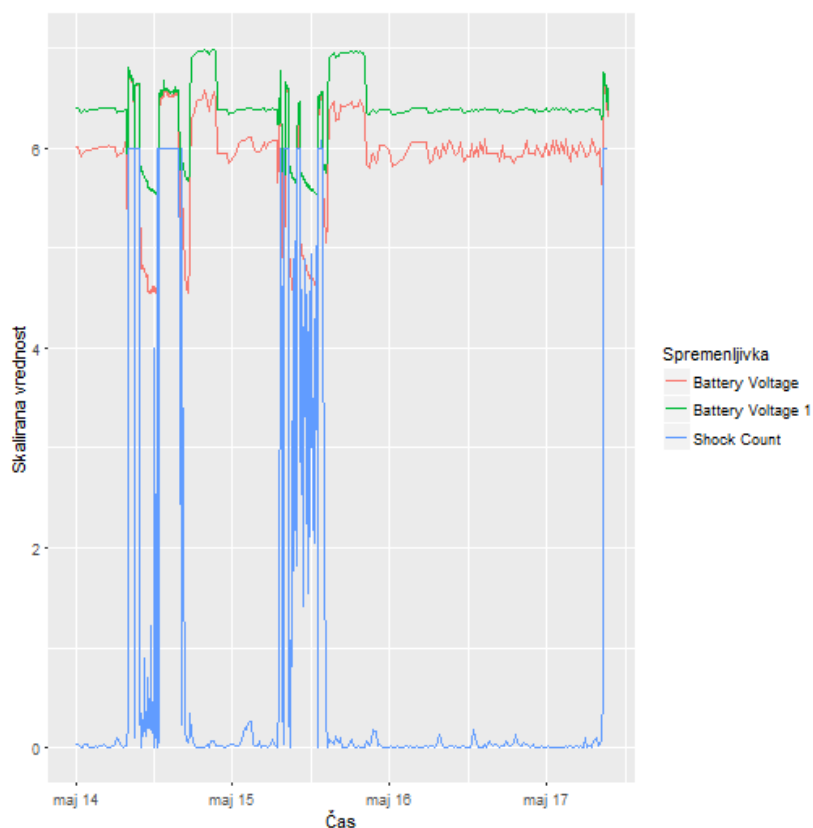
Če zgoraj omenjena problema povežemo, lahko pride do situacije, ko je vrednost *Shock count-a* v mirovanju večja kot med plovbo. Ker je za klasifikacijo plovbe potrebno nastaviti nek prag, je v takem primeru le-tega precej težko določiti.

### 4.3.3 Korelacija med *Shock count* in *Battery voltage*

Med iskanjem korelacij z ostalimi senzorji, ki jih imamo še na voljo, se je pojavila povezava med vrednostmi *Shock count* in *Battery voltage*. Lepo je namreč razvidno, da vsakič, ko *Shock count* doseže visoko vrednost, je stanje *Battery voltage* visoko. To je najbolj očitno pri napravah, ki pogosto zasedejo najvišjo možno vrednost pri *Shock count*. Zaradi lepšega prikaza na sliki 4.19 so vrednosti skalirane, in sicer:

$$Shock\_count = Shock\_count/1000 \quad (4.2)$$

$$Battery\_voltage = Battery\_voltage - 7.5 \quad (4.3)$$

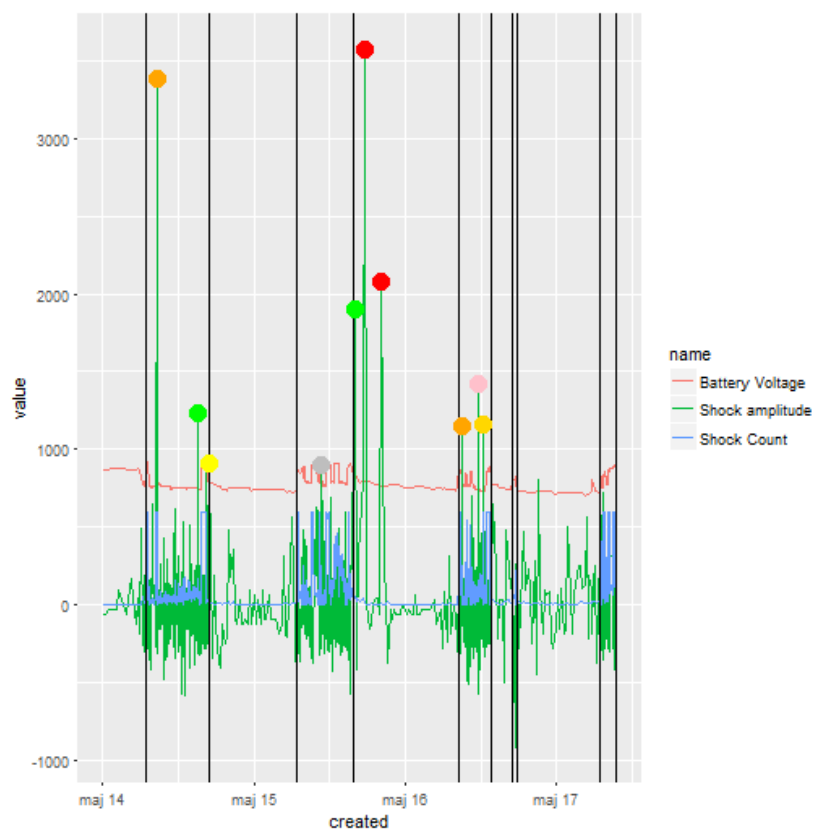


Slika 4.19: Korelacija med *Shock count* in *Battery voltage*.

## 4.4 Rezultati

Cilj naloge je razložiti in osmisliti večje odklone pospeškov – osamelce. To smo dosegli s pomočjo analize, ki je bila do sedaj opisana v tem poglavju. Osamelce smo klasificirali v različne razrede, glede na vrednosti atributov posameznega osamelca v primerjavi z vrednostmi pridobljenimi v analizi. Dopustili smo tudi klasifikacijo v več razredov, kar pomeni, da lahko za take točke z večjo gotovostjo trdimo, da v danem primeru ni prišlo do trka. Za lažjo predstavbo smo izrisali tudi grafe, kjer je vsak osamelec predstavljen kot barvna točka, barva točke pa je odvisna od razreda osamelca, primer je prikazan na sliki 4.20.

Prikažimo celotne rezultate še v tabeli, tako da naštejemo vse razrede in



Slika 4.20: Klasifikacija točk za določeno napravo.

preverimo uspešnost naše klasifikacije. Na tem mestu bomo za vsak osamelec omenili samo najbolj verjeten razred, da lahko izračunamo končno uspešnost.

V tabeli 4.1 lahko vidimo, da smo uspeli klasificirati 48 izmed 66 osamelcev, kar nam prinese 72,73% uspešnost. Seveda je možnosti za napredek še kar precej, ampak glede na razpoložljive podatke so rezultati zadovoljivi.

---

razred	število zadetkov
Pred izplutjem	4
Pred pristankom	1
Sprememba <i>Shock count</i>	4
Reset naprave	3
Visoka napetost	2
Visoka hitrost	12
Sprememba napetosti	10
Visoka hitrost vetra	12
Neklasificirani	18
Skupno	66

Tabela 4.1: Klasifikacija osamelcev.

# Poglavje 5

## Zaključek

Naloga nam je podala nova odkritja in mnogo možnosti za nadaljnje delo. Dva najpomembnejša zaključka sta: vsako plovilo bi morali obravnavati posebej, pridobiti bi morali podatke o potrjenih trkih.

**Obravnava vsakega plovila posebej.** Kot najpomembnejšo ugotovitev naloge, lahko izpostavimo dejstvo, da vrednosti atributov podatkov posameznih naprave in posledično posameznih plovil precej nihajo. V kolikor želimo v končnem produktu zagotoviti visoko stopnjo natančnosti zaznave, se morajo vrednosti, ki določajo ali gre za trk, visoke valove ali kaj drugega, prilagoditi vsaki napravi posebej. Dejavnikov, ki na to vplivajo je več:

- lokacija montaže naprave – vrednosti pospeškov se bodo bistveno razlikovale, če je naprava nameščena nekje v bližini motorja, na palubi ali v kabini.
- vrsta plovila – prav tako nastajajo razlike med vrstami plovil. Na primer jadrnica se bo v vetru odzivala bistveno drugače kot motorno plovilo.
- velikost plovila – tudi v primeru dveh plovil iste vrste, nastajajo razlike v vrednosti pospeškov zaradi dimenzij, geometrije in mase plovila.
- orientacija naprave – v kolikor želimo delati s pospeški po posameznih oseh, je ključnega pomena, da poznamo usmerjenost posameznih osi.

**Pridobitev podatkov o potrjenih trkih.** Ker je morjeplovstvo denarno potrošna panoga, je namensko zaletavanje v druga plovila, obalo ali kaj drugega, nesmotrno početje. Zato je na tem mestu potrebna potrpežljivost in dokler ne ugotovimo ustreznega načina simulacije trkov ali pa preprosto počakamo, da dobimo zadostno količino meritev, pri katerih z gotovostjo vemo, da je prišlo do trka, je nadaljnje delo precej oteženo, saj težko primerjamo podatke pravih trkov z lažnimi, če prvih sploh nimamo.

Glede na to, da je od začetka dela na diplomski pretekla skoraj celotna poletna sezona plutja po Jadranu – čas ko je v uporabi daleč največje število naprav in tudi dejstvo da je v obtoku vse več naprav z vgrajenim pospeškom, bomo do konca leta zbrali dovolj podatkov za nadaljnje analize.

## 5.1 Nadaljnje delo

Vsako plovilo je potrebno obravnavati posebej. Ker je ročna analiza podatkov posameznih plovil, določanje in sprotno popravljanje mejnih vrednosti zelo zamudno opravilo, je naslednji korak vpeljava algoritmov strojnega učenja, ki bi se neodvisno naučili v kakšnih območjih vrednosti operira posamezna naprava. Tako bi prilagajali mejne vrednosti in agent bi lahko samostojno detektiral trke v realnem času.

Ko tak agent doseže zadovoljivo stopnjo uspešnosti, sledi še integracija v končni produkt. Uporabnik bi bil v realnem času obveščen o trku preko mobilne aplikacije, SMS sporočila ali kako drugače.

# Literatura

- [1] Matt Dowle and Arun Srinivasan. *data.table: Extension of 'data.frame'*, 2017. R verzija paketa 1.10.4.
- [2] University Corporation for Atmospheric Research. Network common data form. <http://www.unidata.ucar.edu/software/netcdf/>, 2016. Prvi dostop: 09.02.2017, verzija 4.4.1.1.
- [3] Universtiy Corporation for Atmospheric Research. Netcdf java. <http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/>, 2017. Prvi dostop: 27.04.2017, verzija 4.6.10.
- [4] David Kahle in Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013.
- [5] Sentinel marine solutions. Boat monitor. <http://www.sentinelmarine.net/boat-monitor>, 2015. Prvi dostop: 25.10.2016.
- [6] Earth observing laboratory. Wind direction quick reference. <https://www.eol.ucar.edu/content/wind-direction-quick-reference>, 2017. Prvi dostop: 17.04.2017.
- [7] World Meteorological Organization. A guide to the code form fm 92-ix ext. grib. <https://www.wmo.int/pages/prog/www/WDM/Guides/Guide-binary-2.html>, 1994. Prvi dostop: 09.02.2017.
- [8] David Pierce. *ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files*, 2017. R verzija paketa 1.16.

- 
- [9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.
- [10] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015.
- [11] Nacionalna središnjica za usklađivanje traganjai spašavanja na moru MRCC Rijeka. Statistički podaci sar akcija. [http://www.mppi.hr/UserDocsImages/MPPI%20SAR%201.I-31.XII-16%20STATISTIKA%2020-1\\_17.pdf](http://www.mppi.hr/UserDocsImages/MPPI%20SAR%201.I-31.XII-16%20STATISTIKA%2020-1_17.pdf), 2016. Dostopano: september, 2017.
- [12] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.
- [13] Wikipedia. Sailing – wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Sailing&oldid=798861767>, 2017. Prvi dostop: 07.06.2017.