

Pronalaženje *Blazhko* zvijezda izvan skupova zvijezda

Natjecanje iz astronomije 2024.

2. razred SŠ, jedan natjecatelj

Sažetak Rada.....	3
1 UVOD.....	4
1.1 Motivacija i cilj.....	4
1.2 Promjenjive zvijezde	5
1.2.1 Tipovi promjenjivih zvijezda	5
1.2.2 Pulsirajuće promjenjive zvijezde	5
1.2.3 Povezanost perioda i oscilacije slojeva promjenjivih zvijezda.....	7
1.2.4 Svjetlosne krivulje	8
1.2.5 RR Lire zvijezde	9
1.3 Blazhko efekt.....	10
2 METODOLOGIJA.....	13
2.1 Dohvat i priprema podataka	13
2.1.1 LINEAR podaci	13
2.1.2 Selekcija RR Lira iz LINEAR podataka	14
2.1.2 ZTF podaci.....	16
2.4 Analiza RR Lira	18
2.4.1 Računanje preciznog perioda za RR Lire.....	18
2.4.2 Algoritam analize periodograma	19
2.4.3 Računanje najboljeg fita za svjetlosnu krivulju	21
2.5 Selekcija kandidata za Blazhko zvijezde.....	21
2.6 Analiza Blazhko kandidata.....	24
3 REZULTATI.....	28
3.1 Analiza RR Lira	28
3.2 Analiza RR Lira i pronađak Blazhko zvijezda.....	30
3.2.1 Detaljan opis 3 Blazhko zvijezde: 4101289, 6819457, 7048826, 10260828	31
4 RASPRAVA I ZAKLJUČAK	38
5 LITERATURA I IZVORI.....	43
5.1 Popis slika	44
5.2 Popis grafova	45
5.3 Popis tablica	45
6 PRILOG – programi	46

Sažetak Rada

U ovom praktičnom radu bavili smo se potragom za posebnom vrstom promjenjivih pulsirajućih zvijezda, **Blazhko zvjezdama**. Te zvijezde su RR Lire (RRab ili RRc tipa) koje prolaze kroz modulaciju faze ili amplitude zbog dosada nepoznatih razloga. Možemo pronaći naznake Blazhko efekta u zvjezdama kroz promjenu perioda, amplitude te sporednih frekvencija koje utječu na period u periodogramu, što je graf različitih mogućih frekvencija za izračun perioda naspram koliko dobro period odgovara krivulji sjaja.

Glavni cilj ovog praktičnog rada je analiza podataka svjetlosnih krivulja za 7010 zvijezda te među njima pronaći one koje pokazuju naznake Blazhko efekta.

Kako bismo postigli ovaj cilj, prvo smo pripremili podatke, gdje su podaci dolazili od LINEAR pregleda neba, a podaci za usporedbu i potvrdu modulacije krivulje sjaja je ZTF pregled neba. Jedinstvenost ovoga rada jest mogućnost proučavanja dosada neistraženog dijela neba gdje se nalaze RR Lire, s 2 skupa podataka istih zvijezda s otprilike 10 godina razlike u vremenu promatranja. Od početnih 7010 LINEAR zvijezda, odabrali smo 2914 RR Lira te preko njihovih ekvatorijalnih koordinata pronašli dotične ZTF parove LINEAR zvjezdama.

Zatim smo analizirali svjetlosne krivulje tako što smo računali period pomoću engl. Lomb-Scargle periodograma te primijenili algoritam za analizu periodograma koji traži znakove sporedne Blazhko frekvencije. Također smo računali vrijednost hi-kvadrat statistike koja opisuje koliko dobro izračunat period odgovara promatranoj svjetlosnoj krivulji. Dodatnom selekcijom zvijezda s dovoljnom količinom podataka (minimalno 250 promatranja za LINEAR i 40 promatranja za ZTF pregled neba) ostalo nam je 2857 zvijezda analizirane dotičnim metodama.

Zatim smo napravili algoritam koji prolazi kroz svih 2857 zvijezda te određuje jesu li Blazhko kandidati po postojanju Blazhko frekvencije, dovoljno značajnoj promjeni perioda, amplitude ili vrijednosti hi-kvadrata. Posljednji korak je bila vizualna analiza fazne krivulje pulsiranja zvijezde, njen periodogram i svjetlosne krivulje u svakoj sezoni promatranja kako bismo potvrdili konačne Blazhko zvijezde.

Pronašli smo 136 do sada nepoznatih Blazhko zvijezda, od kojih je 71.32 % RRab tipa, 28.68 % RRc tipa što je u skladu sa dosadašnjim istraživanjima Blazhko zvijezda. Od početnog skupa podataka kojeg smo analizirali, oko 4.67 % RR Lira zvijezda su Blazhko zvijezde. Postotak Blazhko zvijezda kod drugih istraživanja varira od 30 % do 47 % s malom količinom početnih podataka i 5.6 % za istraživanja s velikom količinom podataka.

1 UVOD

1.1 Motivacija i cilj

Zvijezdin sjaj ovisi o njenoj temperaturi i radijusu, no oni nisu konstantni. Sjaj zvijezde stalno se mijenja, uzrokovano termodinamičkim fluktuacijama zvijezde koja se stalno širi i skuplja. Iako je promjena svjetlosti tipična za sve zvijezde, one su često minimalne i teško primjetne. Ali, neke zvijezde drastično mijenjaju svoj sjaj, fluktuiraju po amplitudi od 1 do 2 magnitude (tj. od 2.5 do 6 puta sjajnije u maksimumu sjaja nego u minimumu) te ih zovemo promjenjive zvijezde. Promjenjive zvijezde su iznimno zanimljive te nam mogu mnogo informacija pridonijeti o strukturi galaksija, udaljenosti svemirskih objekata i dr. Poseban tip promjenjivih zvijezda su RR lire. RR Lire pokazuju jedan misteriozan efekt u svojim krivuljama sjaja koji još nije objašnjen: *Blazhko efekt*. Upravo zbog našeg minimalnog razumijevanja efekta te malog broja RR Lira koje ga prikazuju, odlučili smo istražiti postojanje Blazhko efekta kod RR Lira. Pronalaženjem novih Blazhko zvijezda s drugačijim svojstvima od dosadašnjih Blazhko zvijezda pridodaje znanju o Blazhko efektu kako bi znanstvenici mogli otkriti nove informacije i možda dokučiti zašto je on prisutan i kako funkcioniра.

Ciljevi ovog praktičnog rada su:

- ⇒ Istražiti promjenjive zvijezde te kako funkcioniрају,
- ⇒ Razumjeti dosadašnje spoznaje o Blazhko efektu i kako ga prepoznati,
- ⇒ Dohvatiti i pripremiti svjetlosne krivulje RR Lira zvijezda iz dviju različitih baza podataka s ~ 10 godina razlike u vremenu promatranja,
- ⇒ Analizirati periode, fitove i periodograme RR Lira te odabrati kandidate za Blazhko zvijezde,
- ⇒ Vizualno analizirati kandidate, zatim odrediti konačni katalog novih Blazhko zvijezda.

1.2 Promjenjive zvijezde

1.2.1 Tipovi promjenjivih zvijezda

Promjenjive zvijezde su zvijezde čiji se sjaj, odnosno magnituda mijenja. Postoji više razloga zašto zvijezde mijenjaju svoj sjaj, stoga ih dijelimo na *intrinzične i ekstrinzične promjenjive zvijezde*. Intrinzično promjenjive zvijezde su one čiji mehanizam za mijenjanje sjaja funkcioniра zbog fizičkih procesa unutar same zvijezde. Ekstrinzične promjenjive zvijezde mijenjaju svoj sjaj zbog vanjskih utjecaja, kao npr. ekliptične binarne zvijezde, zvezdane pjege te promjena oblika zvijezde zbog gravitacijskih ili magnetskih utjecaja.

Sukladno tome, dijelimo intrinzične promjenjive zvijezde na **pulsirajuće, kataklizmičke, eruptivne promjenjive zvijezde i na mlade zvezdane objekte**. U ovome radu se fokusiramo na *pulsirajuće promjenjive zvijezde*.

Ima mnogo vrsta pulsirajućih promjenjivih zvijezda te su samo neke od njih **RR Lire**, cefeide, **W Virginis**, **RV Tauri**, **Delta Scuti**, **SX Phoenicis** zvijezde, a od zvijezda s dugim periodom su **Mira** i **neperiodično-promjenjive zvijezde**. [1,2,6,14]

1.2.2 Pulsirajuće promjenjive zvijezde

Život jedne zvijezde ovisi o hidrostatskoj ravnoteži gdje se tlak plinova unutar zvijezde stalno sukobljava sa gravitacijom prouzročenom masom zvijezde. Dokle god su ove dvije sile izjednačene, zvijezda ostaje stabilna. Kod pulsirajućih promjenjivih zvijezda, granica ravnoteže se ciklično pomiče u korist tlaka ili gravitacije, no nikada previše da ne bi došlo do urušavanja zvijezde. Pulsirajuća zvijezda ciklično mijenja svoj volumen, što mijenja boju, temperaturu i magnitudu zvijezde. [3,4,5,14]

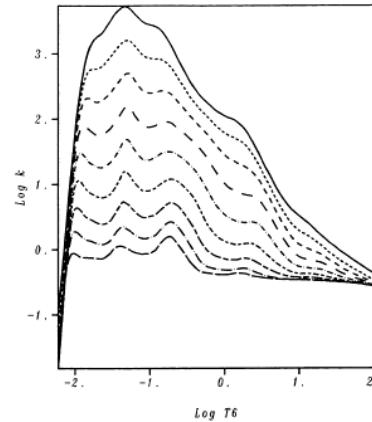
Zvijezda je većinski napravljena od 2 plina, vodika i helija. Unutar zvijezde, temperature su izrazito visoke što omogućuje različite konfiguracije elektrona u atomima vodika i helija, ovisno gdje se sami atomi nalaze. Budući da zvijezda mora povećati ili smanjiti svoj volumen, neki sloj zvijezde mora povećati ili smanjiti svoj volumen, što gura ostale slojeve unutrašnjosti zvijezde da mijenjaju svoju veličinu. Možemo pretpostaviti da se u ovom sloju nalazi plin koji se ponaša kao *idealni plin*. Idealni plinovi ponašaju se po idućem zakonu:

$$P = nkT \quad 1)$$

Gdje je P tlak plina, n je gustoća čestica odnosno broj čestica po jedinici volumena, k je Boltzmannova konstanta, a T je temperatura u Kelvinima. Plin u našem sloju je ograničen ostalim slojevima zvijezda, stoga kada se tlak plina povisuje, gustoća čestica plina i temperatura se također povećavaju. [4,5,6]

Unutar jezgre, termonuklearni procesi spajaju jezgre vodikovih atoma, oslobađajući energiju u obliku fotona, koji onda putuju kroz slojeve zvijezde sve do površine. Količina emitiranih fotona određuje magnitudu, odnosno sjaj zvijezde. Fotoni će ujedno morati proći kroz dotičan sloj zvezdane unutrašnjosti, a parametar koji određuje koliki udio fotona će proći je **neprozirnost plina**. Što je veća neprozirnost, to će fotoni teže prolaziti kroz sloj plina. Također, što je veća temperatura to je manja

neprozirnost. Neprozirnost plina teško je modelirati, no idući model prikazuje trend smanjenja neprozirnosti s povećanjem temperature te se može aproksimirati *Kramerovim zakonom*, opisanim jednadžbom 2. [4,5]

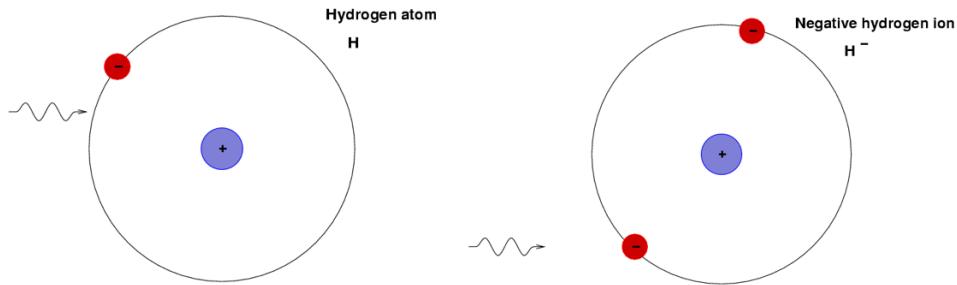


Slika 1: ovisnost neprozirnosti plina o temperaturu (aproksimativan model) [8]

$$\kappa \propto \frac{\rho}{T^{3.5}} \quad 2)$$

Uočavamo kako prema aproksimativnom *Kramerovom zakonu o neprozirnosti plina*, neprozirnost ovisi mnogo više o temperaturi nego li o gustoći plina. Zapažanje ukazuje da mora postojati poseban sloj unutar promjenjivih zvijezda gdje se znatno mijenja gustoća, a ne temperatura kako bi se neprozirnost dovoljno promijenila za oscilaciju sloja te pulsiranje zvijezde. [4]

No, unutar zvijezde ne postoje samo neutralni atomi vodika i helija. Jedinstvena okolina unutrašnjosti zvijezde omogućava negativnu ionizaciju vodika, koji mnogo više međudjeluje sa fotonima, zaustavljajući prođor fotona kroz sloj zvijezde.



Slika 2: prikaz interakcije fotona i atoma H i H^- [6]

Zbog svih razloga i procesa navedenih iznad, oscilacija volumena promjenjivih zvijezda, a s time i magnitude odvija se na sljedeći način:

- 01 Gornji slojevi zvijezde zbog utjecaja gravitacije djeluju velikom silom usmjerenom prema središtu, volumen zvijezde se smanjuje,

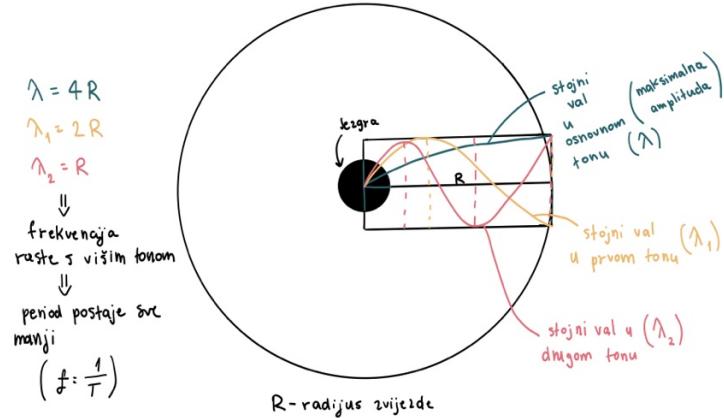
- 02 Zbog ovog pritiska na dotičan sloj, vodikovi (i helijevi) atomi prolaze kroz proces ionizacije, što znatno povećava gustoću sloja, ali ne temperaturu budući da ionizacija koristi energiju te se neprozirnost plina povećava,
- 03 Povećanjem neprozirnosti fotoni iz jezgre ne mogu jednako efikasno putovati do površine zvijezde, smanjujući ukupnu svjetlost zvijezde, odnosno sjaj je najmanji u ciklusu,
- 04 Velika količina fotona koji ne prolaze kroz dotičan sloj su energični te pritisak i temperatura počinju se povećavati od dotičnog, ali i slojeva ispod,
- 05 Povećanje pritiska i temperature pobjeđuje silu gravitacije i povećava ukupan volumen zvijezde, gurajući dotičan sloj prema površini,
- 06 Plin dolazi do novog područja zvijezde i deionizira se, što smanji neprozirnost na standardnu vrijednost te zbog povećanja i volumena i emitirane svjetlosti sjaj je u maksimalan u ciklusu,
- 07 Sloj se ponovno vraća na staru lokaciju zbog nedostatka pritiska te gravitacija opet kreće djelovati na dotičan sloj,
- 08 Proces se ponavlja od koraka 1. [4,5,6]

1.2.3 Povezanost perioda i oscilacije slojeva promjenjivih zvijezda

Najvažniji dio analize promjenjivih zvijezda je **period oscilacije njihovih unutrašnjih slojeva**. On određuje oblik svjetlosne krivulje, tip zvijezde te prisutnost efekata vezanih za promjenjive zvijezde.

Pulsiranje zvijezda ciklična je pojava, gdje zvijezda stalno povećava i smanjuje svoj volumen, u redovitom periodu. Možemo interpretirati pulsiranje zvijezde kao **val** s određenim periodom, frekvencijom i amplitudom. Val koji modelira pulsiranje promjenjivih zvijezda ne kreće se po prostoru već je **stojni val**. U jednostavnom modelu, možemo stojni val smjestiti unutar cijevi gdje je jedna strana zatvorena, a druga otvorena. Najjednostavniji stojni val jest val u **osnovnom tonu**, gdje poprima maksimalnu valnu duljinu. Osnovan ton možemo prikazati tako što jedan kraj stojnog vala učvrstimo za zatvoren kraj cijevi te on titra tako što mu vrh dotakne sam rub otvorenog dijela cijevi. [4]

Najjednostavniji način pulsiranja ili titranja promjenjive zvijezde je također u **osnovnom tonu titranja**, gdje čitava zvijezda povećava i smanjuje svoj volumen. Model cijevi možemo aplicirati na zvijezdu, gdje je radius zvijezde jednak duljini cijevi, zatvoren kraj je jezgra, a otvoren kraj je sam rub zvijezde. Slika ispod prikazuje model cijevi i titranja zvijezde.



Slika 3: model cijevi sa stojnim valovima unutar promjenjive zvijezde (ilustracija autora)

Gledajući osnovan ton, možemo vidjeti da je njegova valna duljina zapravo 4 puta veća od samog radijusa zvijezde, što će rezultirati malom frekvencijom i velikim periodom. No, zvijezda može i komplikiranije pulsirati, tako što postoji samo neki slojevi unutar zvijezde koji pulsiraju, stvarajući stojne valove u prvom, drugom, trećem itd. tonovima. Što zvijezda ima viši ton, valna duljina postaje sve kraća te period postaje sve kraći, što uočavamo u stvarnim promjenjivim zvijezdama. U ovom praktičnom radu proces izračunavanja perioda frekvencije tonova pulsiranja zvijezde, a iz frekvencije se lako izračuna period.

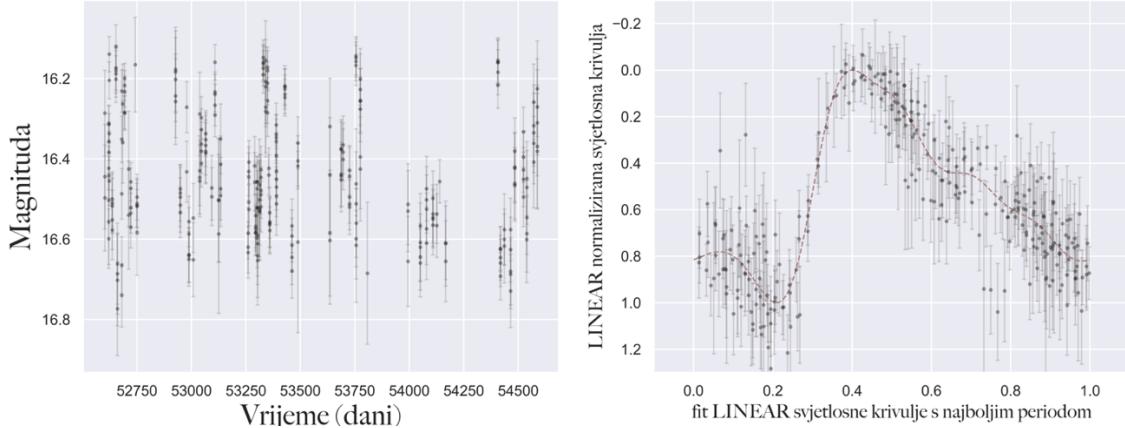
1.2.4 Svjetlosne krivulje

Promjenjive zvijezde promatramo optičkim teleskopima koji se mogu nalaziti na zemlji ili u svemiru te oni mjeri svjetlost odnosno magnitudu zvijezde u svakom promatranju. Ne mogu promatrati danju te ne mogu promatrati istu zvijezdu neprestano, stoga postoje određene sezone promatranja koje se ponavljaju godinama. Vrstu podatka koju prikupljamo za promjenjive zvijezde zove se **svjetlosna krivulja**, jer ona prikazuje promjenu svjetlosti zvijezde u odnosu na proteklo vrijeme.

Oblik svjetlosne krivulje ovisi o periodu te o vrsti zvijezde, stoga će svaka vrsta zvijezde imati različitu krivulju ovisno o njenim obilježjima, ali i načinu pulsiranja. [1,2,14]

Izračunom perioda možemo prikazati jedan ciklus promjenjive zvijezde. Slika 4 prikazuje ukupnu svjetlosnu krivulju za jednu zvijezdu iz podataka korištenih u ovome radu te prikaz jednog ciklusa pulsiranja.

LINEAR objekt 50402



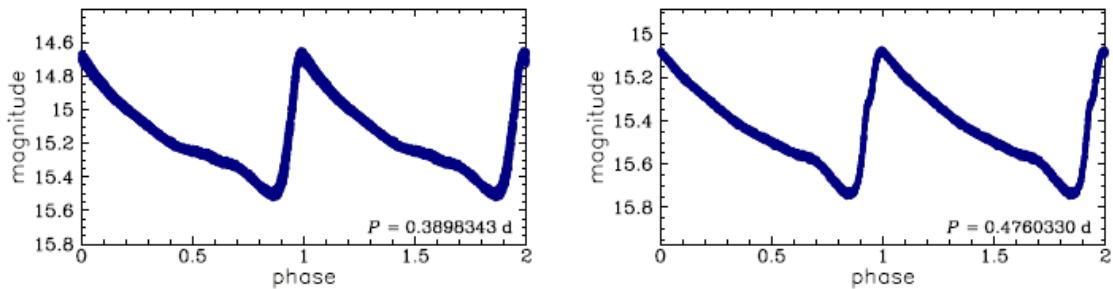
Slika 4: primjer ukupne svjetlosne krivulje i jednog ciklusa

1.2.5 RR Lire zvijezde

RR Lire su vrsta starih pulsirajućih promjenjivih zvijezda, koje najčešće pronalazimo u kuglastim skupovima zvijezda. Imaju vrlo kratke periode, od 0.05 do 1.2 dana te ovisno o tipu RR Lire, opseg perioda varira te promjena magnitude od 0.3 do 2 magnitude. Kada bismo pogledali HR dijagram, RR Lire zauzimaju vrlo mali i uzak prostor dijagrama, što ih čini lakšom za selektirati po boji, temperaturi i magnitudi. [7, 9] One su bijeli divovi spektralnog tipa A. Mali opseg magnituda ih čini odličnim za određivanje udaljenosti u galaksiji, stoga ih zovemo „standardne svijeće“ za računanje udaljenosti. RR Lire mogu pulsirati u jednom od 3 načina: osnovnim tonom, prvim tonom, i osnovnim i prvim tonom. [14, 15, 16, 17, 18, 19]

1.2.4.1 RRab tip

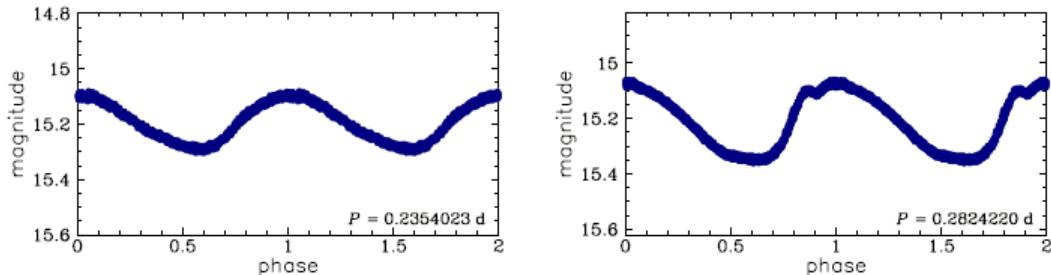
RRab tip zvijezda je najčešća varijanta RR Lira zvijezda. One pulsiraju osnovnim tonom, što znači da imaju dulje periode od ostalih tipova, od 0.3 do 1.2 dana. Karakteristike njihovih svjetlosnih krivulja jest nagli rast i sporiji pad sjaja (magnitude), kao što je prikazano na slici ispod. [14, 15, 16, 17, 18, 19]



Slika 5: svjetlosna krivulja RRab tipa [18]

1.2.4.2 RRc tip

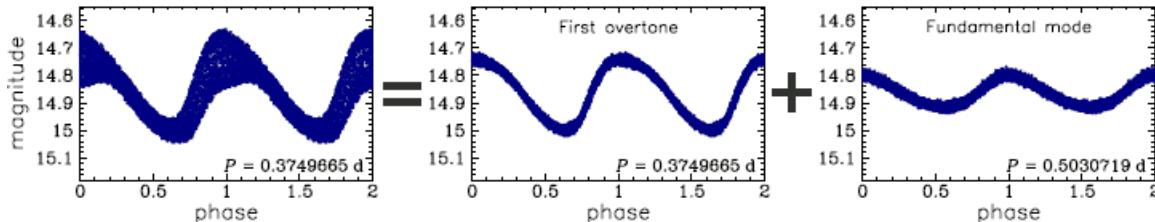
RRc zvijezde imaju simetričnu svjetlosnu krivulju koja nalikuje na sinusoidu. Imaju vrlo kratak period, od 0.2 do 0.5 dana, zato što pulsiraju u prvoj tonu. Također, imaju manju promjenu magnitude od RRab tipa. [14, 15, 16, 17, 18, 19]



Slika 6: svjetlosna krivulja RRc tipa [18]

1.2.4.3 RRd tip

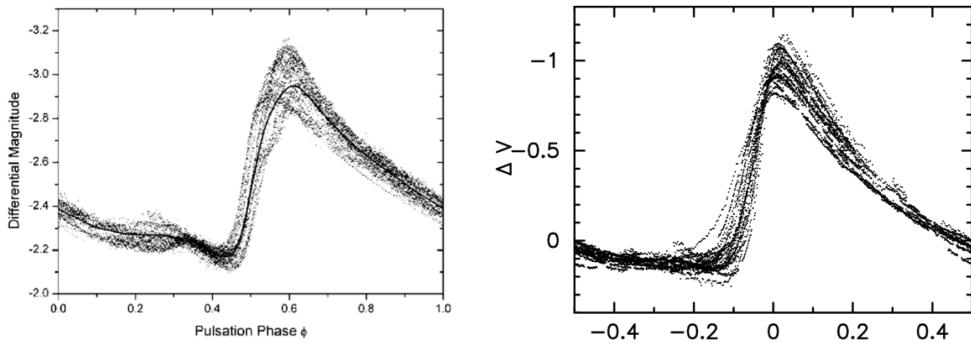
RRd su posebne zato što pulsiraju i u osnovnom i u prvom tonu te imaju zato 2 perioda, prvi period osnovnog tona koji je najčešće 0.7 dana, dok za prvi ton je otprilike 0.5 dana. Svjetlosna krivulja nastaje spajanjem pulsiranja u oba tona. [14, 15, 16, 17, 18, 19]



Slika 7: spajanje svjetlosnih krivulja RRd tipa [18]

1.3 Blazhko efekt

Blazhko efekt se pojavljuje kod promjenjivih zvijezda, posebno kod RR Lira. Zvijezde koje su Blazhko zvijezde pokazuju **modulaciju svjetlosne krivulje**, odnosno periodično mijenjanje oblika svjetlosti, a mijenjanjem oblika mijenja se i amplituda i period. Uzrok Blazhko efekta i kako točno funkcioniра nije još poznato te postoje mnoge potencijalne teorije koje pokušavaju objasniti efekt. Slike ispod prikazuju svjetlosne krivulje Blazhko efekta. [22, 23, 24, 25, 26, 27, 29]



Slika 4: prikaz Blazhko efekta kod RR Lira [25, 38]

No, iako je Blazhko efekt neobjašnjiv dosada, postoji jedan efekt u fizici s kojim možemo usporediti Blazhko efekt teorijski. [10,11]

Standardan val u fizici možemo opisati idućom formulom:

$$y(t) = A \sin(2\pi ft) \quad 3)$$

gdje je A amplituda, f frekvencija vala i t je vrijeme. Ako bismo uzeli 2 vala sa malom razlikom u frekvenciji i zbrojili njihove vrijednosti te aplicirali trigonometrijsku jednakost zbrajanja 2 sinusa 2 kuta:

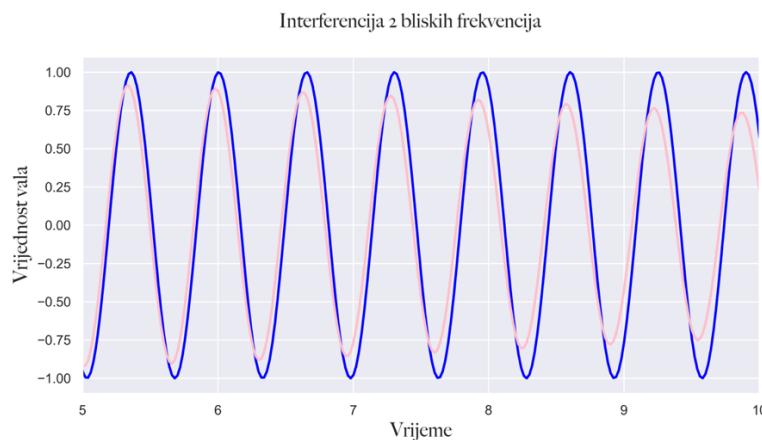
$$y(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) \quad 4)$$

$$\sin(a) + \sin(b) = 2 \cos\left(\frac{a-b}{2}\right) \sin\left(\frac{a+b}{2}\right) \quad 5)$$

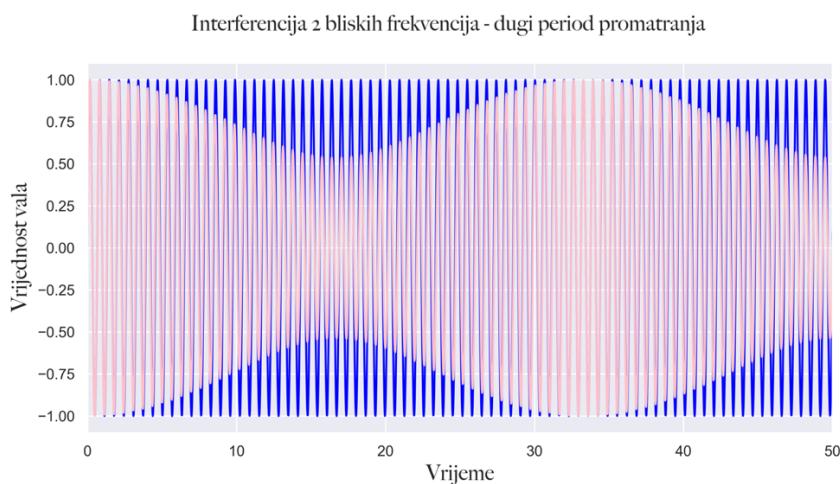
Dobili bismo iduću jednadžbu:

$$y(t) = 2 \cos\left(2\pi \frac{f_1 - f_2}{2} t\right) \sin\left(2\pi \frac{f_1 + f_2}{2} t\right) \quad 6)$$

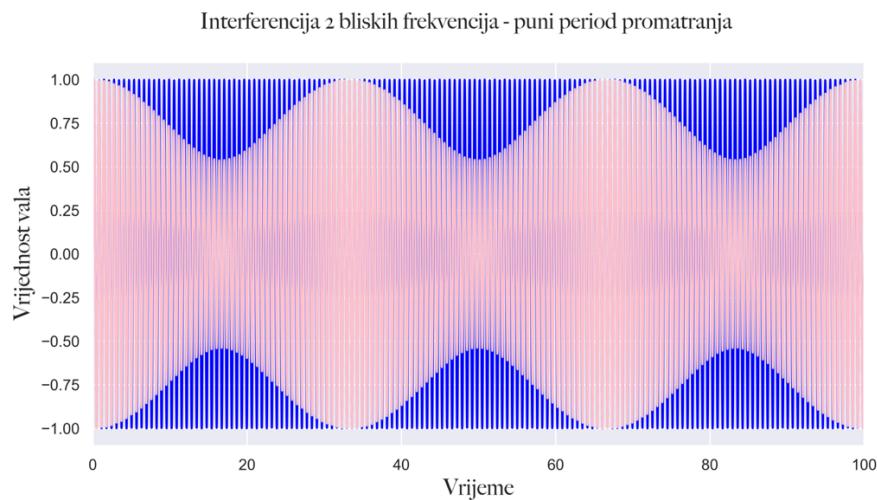
Budući da su obje frekvencije vrlo slične u vrijednosti, možemo prvu zgradu interpretirati kao Δf , a drugu kao f_0 , odnosno prosječnu frekvenciju. Budući da je razlika u frekvenciji Δf mnogo manja od prosječne frekvencije f_0 , ona će stvarati oscilaciju amplitude sinusoide. Razlog zašto se stvara modulacija amplitude jest kada zbrajamo valove, njihove vrijednosti će se eventualno poklapati (iako imaju različite frekvencije) te međusobno poništiti budući da je jedna vrijednost negativna, a druga pozitivna. [10,11] Slike ispod prikazuju modulaciju valova preko različitih perioda vremena te grafički prikaz promjene amplitude uočili su drugi istraživački radovi o Blazhko efektu. [23]



Slika 5: interferencija valova pri malom period promatranja



Slika 6: interferencija valova pri duljem period promatranja, modulacija amplitude je vidljiva



Slika 7: modulacija amplitude preko punog perioda promatranja, dolazi do izražaja

2 METODOLOGIJA

2.1 Dohvat i priprema podataka

Za uspješnu analizu RR Lira u potrazi za Blazhko efektom, potrebno je razmatrati svjetlosne krivulje iste zvijezde u drugačijem vremenskom razdoblju. Zato u ovom praktičnom radu koristimo 2 baze podataka: **LINEAR** i **ZTF**, kako bismo mogli uspoređivati parametre zvijezda s vremenskim razmakom od otprilike 10 godina. U ovom praktičnom radu koristili smo sljedeće materijale:

- ⇒ Računalo s pristupom internetu
- ⇒ Web stranicu GitHub¹
- ⇒ Programski jezik Python²
- ⇒ VsCode³ – programsko sučelje
- ⇒ Jupyter notebooks – Anaconda⁴ okolina

2.1.1 LINEAR podaci

LINEAR set podataka (Lincon Near-Earth Asteroid Research) sadrži 7010 svjetlosnih krivulja periodičnih promjenjivih zvijezda. Iako je izvorna misija ovog istraživanja bila pronaći asteroide, naknadno se promatrao značajan dio neba, uključujući sjaj raznih promjenjivih zvijezda. **LINEAR** istraživanje provedeno je s 2 teleskopa na lokaciji u Novom Meksiku. Promatrali su široki dio vidljivog spektra i bliski infracrveni spektar.

Proces čišćenja podataka prije analize sadržavao je sljedeće postupke:

1. Uklanjanje šuma koje stvara kamera
2. Uklanjanje umjetnih linija i gradijenata zbog različite osjetljivosti piksela
3. Oduzimanje pozadinskog svjetla pomoću Poissonove statistike

Nakon čišćenja podataka, morali su biti astrometrijski ponovno baždareni, odnosno znanstvenici su morali osigurati da koordinate objekata u podacima odgovaraju drugom istraživanju, poput SDSSa. Od svih podataka, 7 % je imalo lošu astrometriju (koordinate nisu odgovarale) pa su ih znanstvenici uklonili iz skupa podataka. Ostali podaci imali su 1-2 % pogreške u astrometriji. [30,31]

Idući korak bio je ponovno kalibriranje fotometrije, odnosno filteri i svjetlosni izvori morali su biti ponovno kalibrirani kako bi odgovarali SDSS podacima. Ispravci su se odnosili na pogreške CCD kamere, ispravljanje magnituda objekata i ispravljanje podcijenjenih pogrešaka magnitude. Pošto je kalibracija održena, podaci su grupirani u različite kategorije, uključujući kategoriju promjenjivih zvijezda. Iako izvorni skup podataka ima 5 milijuna objekata, od njih samo 7010 su potvrđene periodične promjenjive zvijezde. Znanstvenici su naknadno analizirali promjenjivi dio skupa podataka

¹ <https://github.com>

² <https://www.python.org>

³ <https://code.visualstudio.com>

⁴ <https://www.anaconda.com>

kategorizirajući krivulje svjetlosti, izračunavajući njihove periode te izvodeći zaključke iz prethodnih faktora. [30,31]

LINEAR podaci nalaze se unutar Python modula *AstroML*⁵ kao već obrađen i pripremljen skup podataka od 7010 periodičnih promjenjivih zvijezda. **LINEAR** podatke učitavamo na sljedeći način:

```
from astroML.datasets import fetch_LINEAR_sample
dataL = fetch_LINEAR_sample(data_home='..../inputs')
```

Podatke spremamo u lokalnu mapu zvanu `inputs`. **LINEAR** podaci strukturirani su unutar Python `array` objekta, odnosno multi-dimenzionalne liste. Struktura **LINEAR** podataka je sljedeća:

1. `dataL` je objekt `AstroML.datasets` i sadrži sljedeće:
 - a. `dataL.ids` popis je svih ID-ova zvijezda, gdje je ID identifikacijski broj zvijezde, svaka ima jedinstveni broj
 - b. `dataL.get_light_curve()` je funkcija koja pristupa podacima za određeni ID krivulje svjetlosti. Svaka krivulja svjetlosti je `array` objekt gdje je svaki stupac vrijeme, magnituda i pogreška magnitudo. Kako bi se dobio niz za svaki stupac, originalni niz mora biti transponiran – svaki stupac podataka pretvara se u zasebnu listu
2. `dataL.targets` sadrži sve metapodatke za svaku svjetlosnu krivulju, npr. rektascenziju, deklinaciju, itd.

2.1.2 Selekcija RR Lira iz **LINEAR** podataka

Kako bismo preuzeli ZTF parove LINEAR zvijezdama, prvo je potrebno selektirati sve RR Lire iz skupa LINEAR podataka. `dataL.targets` ne sadrži dovoljno podataka kako bi se LINEAR zvijezde selektirale, stoga je potrebno koristiti drugu bazu podataka, `LINEAR geneva`, koja je zapravo tablica informacija o LINEAR zvijezdama koje imaju klasifikaciju svjetlosne krivulje.

Prvi korak je preuzimanje Geneva baze podataka te selekcija LINEAR ID-ova iz tablice za koje imamo podatke u LINEAR skupu zvijezda:

```
def select_good_LINEAR(LDATA):
    """
    Ova funkcija selektira LINEAR IDove iz Geneva baze podataka koji su
    prisutni u LINEAR skupu podataka.

    Argumenti:
        LDATA = LINEAR podaci
```

⁵ AstroML: <https://www.astroml.org>

```

"""
# -----
IDs = [x for x in LDATA.ids] # stvaranje liste IDova

# PREUZIMANJE GENEVA BAZE PODATAKA
#-----
dataPeriods = fetch_LINEAR_geneva()
dataPeriods = pd.DataFrame(dataPeriods) # prebacivanje u DataFrame oblik
podataka za lakše korištenje
ID_orig = list(dataPeriods['LINEARobjectID'].to_numpy()) # stvaranje
liste Geneva IDova

LINEAR_data = pd.DataFrame()# priprema prazne tablice za selektirane
LINEAR zvijezde

for id in ID_orig: # za svaku GENEVA zvijezdu
    if id in IDs:# ako je ID prisutan i u LINEAR skupu
        index = ID_orig.index(id) # spremi indeks gdje je pronađen ID
        row = pd.DataFrame(dataPeriods.iloc[[int(index)]]) # selektiraj
red GENEVA tablice
        # spoji red sa početnom LINEAR_data tablicom
        LINEAR_data = pd.concat([LINEAR_data,
row.reset_index(drop=True)], ignore_index=True, axis=0)

return LINEAR_data # spremi gotovu tablicu sa svim selektiranim GENEVA
IDovima

LINEAR_periods = select_good_LINEAR(dataL)
print(LINEAR_periods.shape)
LINEAR_periods.head()

```

Nakon izvršenja programa, tablica izgleda ovako:

Tablica 1: Metapodaci LINEAR zvijezda koji imaju klasifikaciju tipa zvijezde

(5204, 14)

	ra	dec	ug	gi	iK	JK	logP	Ampl	skew	kurt	magMed	nObs	LCtype	LINEARobjectID
0	119.526443	46.962120	1.166	0.369	1.018	0.269	-0.254138	0.619	-0.313	-0.567	16.37	301	1	29848
1	119.324013	47.095505	1.356	0.521	1.167	0.311	-0.244691	0.709	-0.493	-0.997	15.02	289	1	32086
2	119.712975	52.149574	1.175	0.386	1.100	0.205	-0.191591	0.487	-0.291	-0.879	16.46	284	1	50402
3	118.491257	53.168125	0.805	0.554	1.618	0.198	-0.178900	0.695	-0.026	-1.059	14.08	274	1	61011
4	119.187241	53.379295	1.123	0.207	1.074	0.210	-0.275092	0.624	-0.547	-0.322	16.54	276	1	62892

Gdje su značenja stupaca sljedeća:

- 01 ra – rektascencija,
- 02 dec – deklinacija,
- 03 ug – UG indeks boje (Ultraviolet – Green ili Ultraljubičasto – Zeleno),
- 04 gi – GI indeks boje (Green – Infrared ili Zeleno – Infracrveno),
- 05 iK – dodatni infracrveni indeks boje,

- 06 JK - dodatni infracrveni indeks boje,
- 07 logP – logaritam perioda (u danima) zvijezde,
- 08 Ampl – amplituda promjene sjaja (u magnitudama),
- 09 Skew – asimetričnost vrijednosti magnituda svjetlosnih krivulja [12],
- 10 Kurt – kurtozija vrijednosti magnituda svjetlosnih krivulja [12],
- 11 magMed – medijan vrijednosti magnituda svjetlosnih krivulja [12],
- 12 nObs – broj promatranja po krivulji,
- 13 Lctype – klasifikacija svjetlosne krivulje,
- 14 LINEARobjectID – ID LINEAR objekta.

Kako bismo uspješno selektirali RR Lire, iz tablice iznad potrebni su nam stupci `gi`, `Lctype` i `LINEARobjectID`. Budući da RR Lire zauzimaju vrlo mali dio HR dijagrama, mogu se uspješno odabrati po boji, pogotovo preko `gi` filtera. `Lctype` 1 ili 2 ukazuju na RRab i RRc tip RR Lire, stoga je i klasifikacija krivulje potrebna. [34]

```
Lrrlyr = LINEAR_periods[(LINEAR_periods['gi'] > -0.5) & (LINEAR_periods['gi'] < 0.4) & (LINEAR_periods['Lctype'] > 0) & (LINEAR_periods['Lctype'] < 3)] # selekcija RR Lira
Lrrlyr = Lrrlyr.reset_index(drop=True) # resetiranje indeksa
Lrrlyr.to_csv('../outputs/Lrrlyr_unprocessed.csv', index=False)
```

Pomoću prethodno objašnjene analize, selektirano je sveukupno 2941 RR Lira. Idući korak je pronalaženje ZTF parova RR Lirama.

2.1.2 ZTF podaci

Zwicky Transient Facility ili ZTF je optičko istraživanje koje koristi Palomar Schmidtov teleskop dijametra objektiva 1.22 m za promatranje čitavog neba sjeverne hemisfere u 3 različita filtera: „*g*“ odnosno zeleno (engl. green), „*r*“ odnosno crveno (engl. red) i „*i*“ odnosno infracrveno (engl. infrared). Promatra nebo od listopada 2017. godine. Teleskop ima vrlo veliko vidno polje te onda ima sposobnost primanja mnogo podataka. Svi podaci koje ZTF prikupi pohranjuju se u IPAC⁶ institutu u Kaliforniji, kojem se može pristupiti putem IRSA⁷ sučelja. [32]

ZTF je uspio promatrati **prijelazne (nepostojane) objekte**, npr. supernove koje se nalaze u drugim galaksijama. Uz njih, promatrao je i promjenjive objekte u našoj galaksiji što uključuje promjenjive zvijezde i asteroide. Svjetlosne krivulje u ZTF skupu podataka nadograđene su svakih nekoliko mjeseci, a pohranjene su u HDF5 datotekama za svako polje promatranja. [32]

⁶ IPAC: <https://www.ipac.caltech.edu>

⁷ IRSA: <https://irsa.ipac.caltech.edu/frontpage/>

2.1.2.1 Dohvat LINEAR koordinata

ZTF parove LINEAR zvijezdama tražimo pomoću **ra** i **dec** koordinata (rektascencije i deklinacije), tako što pretražujemo bazu podataka ZTFa i tražimo podatke za zvijezde koje se nalaze kod dotičnih koordinata unutar malog radijusa. Iduća funkcija opisuje proces:

```
def getZTFlightcurve(ra, dec, radius=3.0):
    """
    Ova funkcija koristi se koordinatama LINEAR zvijezde kako bi pronašla tu istu zvijezdu u ZTF podacima.

    Argumenti:
        ra(float): rektascenzija
        da(float): deklinacija
        radius(float): radijus područja potrage na nebu
    """
    try:
        lcq = lightcurve.LCQuery() # stvaranje ztfquery objekta za potragu neba
        res = lcq.from_position(ra, dec, radius) # traženje u podacima prema rektascenziji, deklinaciji i radijusu

        # Selekcija samo podataka za vrijeme, magnitude, greške u magnitudi, filter te catflag
        ZTFdata = res.data[['mjd', 'mag', 'magerr', 'catflags',
        'filtercode']]

        # Brisanje očitih lažnih točaka
        ZTFdata = ZTFdata.loc[ZTFdata['catflags'] < 32768]
    except:
        # Ako nema podataka, spremi praznu tablicu
        ZTFdata = pd.DataFrame()
    return ZTFdata
```

Funkcija stvara `ztfquery` objekt, koji je dio Python modula `ztfquery`, poseban modul ZTFa. Preko ovog objekta, traži se područje neba s određenim radijusom koje odgovara unesenim koordinatama. Pošto je pronađena zvijezda, preuzeti su stupci podataka s vremenom, magnitudom, greškom magnitude, catflagom (oznakom kvalitete mjerjenja) te filterom. Zatim se brišu očite lažne točke ovisno o vrijednosti catflaga. Ako program nije uspio pronaći par LINEAR IDu, spremi se prazna tablica podataka.

Zatim, pomoću iduće funkcije, tražimo parove za sve LINEAR RR Lire zvijezde:

```
if os.path.isfile(NAME): # ako dotična datoteka postoji
    ZTF_data_lc = np.load(NAME, allow_pickle=True) # učitavanje podataka
else:
    Lids = Lrrlyr['ID'].to_numpy()
    rectascension = Lrrlyr['ra'].to_numpy() # lista svih rektascenzija
    declination = Lrrlyr['dec'].to_numpy() # lista svih deklinacija
    ZTF_data_lc = [] # postavljanje prazne liste koja će se ispuniti podacima
```

```

for i in tqdm(range(len(Lids))):
    Lid = Lids[i] # određujemo svaki LINEAR ID
    ra = rectascension[i] # pronalazimo dotičnu rektascenziju
    dec = declination[i] # pronalazimo dotičnu deklinaciju
    ZTFdata = getZTFlightcurve(ra, dec) # koristeći prethodnu funkciju,
    tražimo ZTF podatke

    ZTF_data_lc.append((Lid, ZTFdata)) # spremamo ZTF podatke za dotičnu
    zvijezdu

    # spremamo podatke na računalo
    if i%10==0:
        ZTF_data_lca = np.array(ZTF_data_lc, dtype=object)
        np.save("../outputs/ZTF_dataset_"+end+".npy", ZTF_data_lca,
allow_pickle=True)

    ZTF_data_lca = np.array(ZTF_data_lc, dtype=object)
    np.save("../outputs/ZTF_dataset_"+end+".npy", ZTF_data_lca,
allow_pickle=True)

```

Pomoću petlje iznad, prolazimo kroz sve selektirane RR Lire i tražim njihove ZTF parove. Nakon pripreme i dohvata podataka, možemo provesti osnovnu statističku analizu podataka.

2.4 Analiza RR Lira

Nakon što smo pripremili sve LINEAR i ZTF parove zvijezda, potrebno ih je detaljno analizirati za iduću fazu praktičnog rada. Analiza uključuje precizno računanje perioda i LINEAR i ZTF podataka za dotičnu zvijezdu, analizirati produkt računanja perioda odnosno periodogram i računanje vrijednosti chi2 te fit za svaku svjetlosnu krivulju. [35,36] Čitav kod za analizu RR Lira možete pronaći u [prilogu](#), ovdje je kvalitativno objašnjen postupak i teorija.

2.4.1 Računanje preciznog perioda za RR Lire

U matematičkom smislu, svjetlosna krivulja je skup točaka u koordinatnom sustavu. Period je veličina koja određuje oblik ovih točaka u skupu, odnosno što je period veći izduženja je krivulja, a što je manji uža je krivulja. Pod „krivulja“ misli se na liniju koju bismo nacrtali povezujući sve točke svjetlosne krivulje kao aproksimacija njenog oblika, odnosno **fit** svjetlosne krivulje.

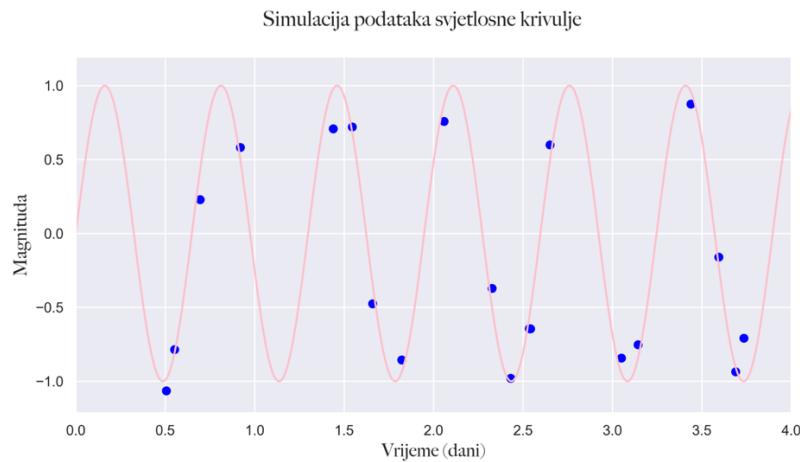
Kada bismo imali neke točke pozicionirane unutar koordinatnog sustava i htjeli bismo povući liniju koja najbolje spaja sve točke, prvo bismo nacrtali ravnu liniju. Linearna funkcija je najgrublja aproksimacija podataka te ih najčešće ne opisuje na najprecizniji način. Koristeći idući polinom po veličini, kvadratnu funkciju, možda bismo mogli bolje opisati točke. Ako koristimo kubnu funkciju ili polinom 4., 5. ili čak 6. stupnja, možemo još bolje opisati podatke i imati točniji *fit*.

Svetlosnu krivulju također možemo prikazati kao set podataka ovisnim o vremenu, odnosno promjenu sjaja o vremenu možemo opisati valovima. Najjednostavniji matematički val jest *sinusoida*. Što više sinusoida spajamo zajedno, možemo postići kompleksniji val što je analogno procesu povećanja stupnja polinoma za što bolji *fit*.

Zato za računanje perioda promjenjivih zvijezda koristimo metodu koja se zove engl. *Lomb-Scargle* metoda. Ona se bazira na matematičkom *Fourier* nizu te uspješno radi za podatke koji nisu ravnomjerno raspoređeni po x-osi (razlike između svake 2 točke nisu jednake) [33, 37]. Lomb-Scargle metoda uzima svjetlosnu krivulju, testira različite frekvencije sinusoida koja je građena od jednostavnijih sinusoida i određuje je li dotična frekvencija zbilja najbolji fit za podatke. Kvantificiranjem je li *fit* dobar postižemo veličinu zvanom *uspješnost fita ili hi-kvadrat*. [12] Kada bismo nacrtali graf svake frekvencije i koliko se uspješno slaže sa dotičnom svjetlosnom krivuljom, dobili bismo njen **periodogram**, koji je važan za daljnju analizu. Frekvencija sa najvećom vrijednošću periodograma bira se kao najbolja frekvencija za dotičnu svjetlosnu krivulju te je najbolji period recipročna vrijednost frekvencije.

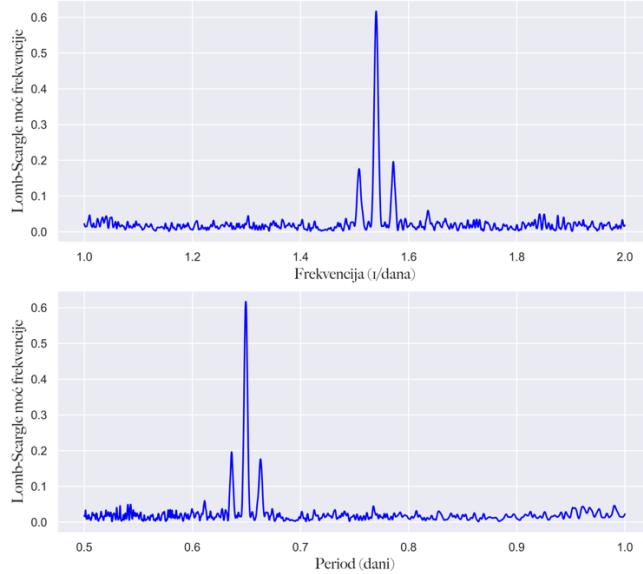
2.4.2 Algoritam analize periodograma

Izračunom perioda, produkt je periodogram koji je izrazito važan za analizu Blazhko efekta. [23,24,25,26,27] Primjer idealnog periodograma Blazhko zvijezde možemo prikazati pomoću simulirane svjetlosne krivulje, koja je izgrađena pomoću interferencije glavne i blazhko frekvencije iz [dijela 1.2.3](#).



Slika 8: simulacija svjetlosne krivulje za analizu periodograma

Podaci su plave točke na slici 8, koje imaju i pogreške u magnitudi te nisu ravnomjerno raspoređene vremenski, kao i prave svjetlosne krivulje. Koristeći Lomb-Scargle periodogram, možemo analizirati frekvencije koje grade svjetlosnu krivulju te odrediti period.



Slika 9: Lomb-Scargle periodogram frekvencija i perioda simulacije svjetlosne krivulje

Graf prikazuje jedan glavni vrh (najviši vrh) koji reprezentira glavnu frekvenciju te 2 manja lokalna vrha koji reprezentiraju **Blazhko frekvenciju**. Razlika po x-osi prema frekvenciji između lokalnih vrhova i glavnog vrha je Δf iz [dijela 1.2.3](#). Upravo te lokalne vrhove oko glavnog vrha koji reprezentira period svjetlosne krivulje tražimo u periodogramima svjetlosnih krivulja LINEAR i ZTF zvijezda.

Za potrebe ovog rada napravili smo algoritam koji traži Blazhko vrhove unutar periodograma te on funkcioniра na sljedeći način:

- 01 Dijeli periodogram na 2 dijela: lijevi dio, sve frekvencije prije glavnog maksimuma te desni dio, sve frekvencije nakon glavnog maksimuma,
- 02 Zatim, množi sve frekvencije s lijeve strane s svim frekvencijama s desne strane periodograma. Množenjem obje strane povećavamo odnos vrhova i pozadinskog šuma, gdje će se korisni vrhovi naglasiti, a utjecaj šuma smanjiti,
- 03 For petlja prolazi kroz svaku vrijednost pomnoženog periodograma te traži najveći vrh koji nije godišnji alias promatranja (tj. lažni vrh zbog utjecaja godišnjeg ciklusa promatranja), koji bi trebao označavati mjesto gdje su se dvije blazhko frekvencije pomnožile
 - a. Ako petlja pronađe ovaj vrh koji je u usporedbi sa pozadinskim šumom viši, smatra se da periodogram prikazuje Blazhko zvijezdu te se računa Blazhko period, značajnost Blazhko vrha te omjer glavnog i Blazhko vrha
 - b. Ako petlja ne pronađe ovaj vrh, smatra se da periodogram ne prikazuje Blazhko zvijezdu te se spremaju prazne vrijednosti za blazhko parametre.

Naravno, algoritam nije savršen te nekada pronađe lažne Blazhko frekvencije ako jedan vrh je vrlo značajan, a drugi lokalni vrh s kojim je pomnožen nije značajan ili ako je pomnožena vrijednost minimalno viša od pozadinskog šuma. No ipak, korisnost ovog algoritma u identificiranju potencijalnih Blazhko zvijezda i odstranjivanje zvijezda koje definitivno ne prikazuju efekt je velika.

2.4.3 Računanje najboljeg *fit* za svjetlosnu krivulju

Osim analize perioda i periodograma, potrebno je analizirati koliko dobro taj period opisuje svjetlosnu krivulju. Statistička vrijednost koja govori koliko dobro neki *fit* odgovara skupu podataka zove se hi-kvadrat te je opisana idućom formulom: [12]

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{\sigma^2} \quad 7)$$

Gdje je χ^2 hi-kvadrat, O_i promatrana varijabla odnosno vrijednost (magnituda svjetlosne krivulje), σ je greška mjerena, te E_i je očekivana vrijednost varijable, odnosno *fit* svjetlosne krivulje. Što je veći hi-kvadrat, to je period lošija aproksimacija oblika krivulje, a što je manji onda je period dobra aproksimacija oblika krivulje. Što se tiče analize Blazhko zvijezda, najbitnija je vrijednost hi-kvadrata za **prosječni period** i LINEAR i ZTF podataka za istu zvijezdu. Ako ne dolazi do Blazhko efekta, periodi će ostati vrlo bliski te će prosječan period vrlo dobro opisivati i LINEAR i ZTF podatke. Ali, ako dolazi do Blazhko efekta onda periodi neće ostati slični te prosječan period neće dobro opisivati i LINEAR i ZTF podatke, odnosno hi-kvadrat će biti velik.

Koristeći se prethodno objašnjениm metodama, analiziramo svih 2914 parova LINEAR i ZTF zvijezda, no ako ne postoji ZTF podaci za dotičnu LINEAR zvijezdu, ona se preskače i ne analizira se LINEAR zvijezda. Nakon analize, ostalo nam je 2857 zvijezda za daljnju analizu.

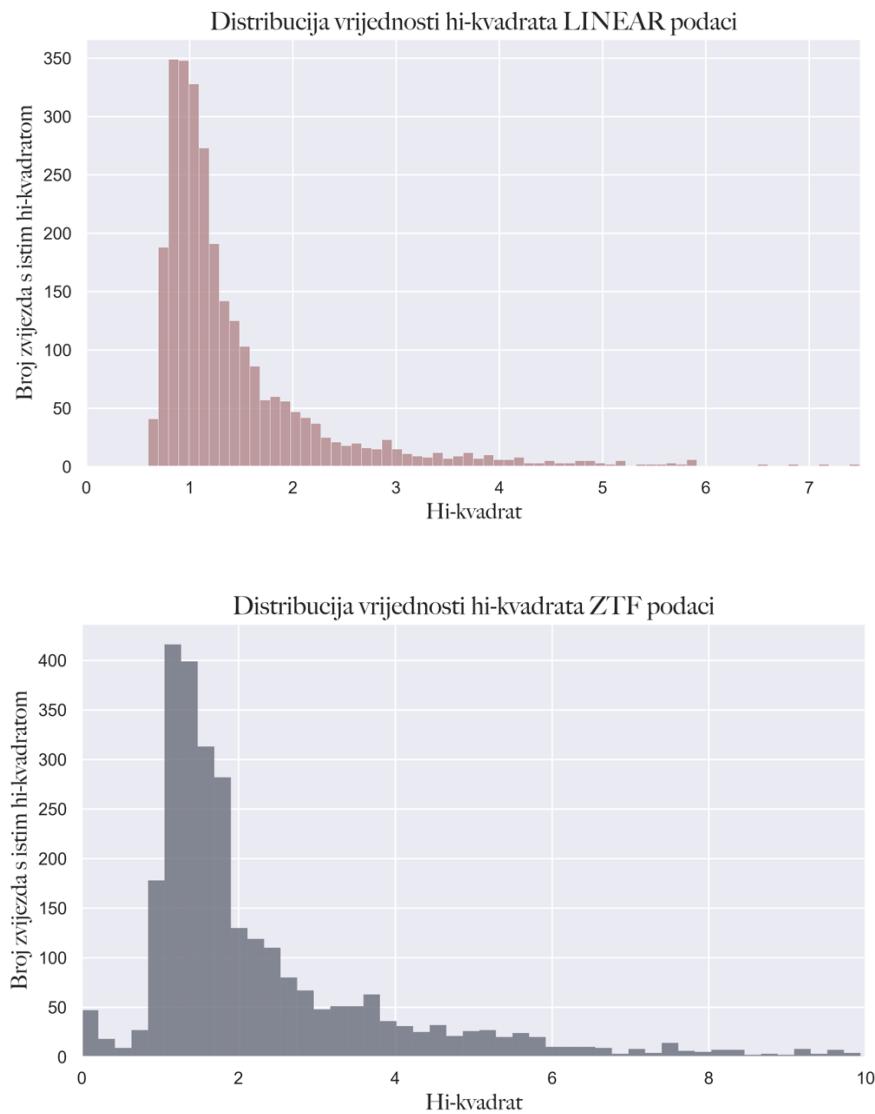
2.5 Selekcija kandidata za Blazhko zvijezde

Nakon analize RR Lira, potrebno je birati zvijezde koje su potencijalno Blazhko zvijezde. Stvaramo algoritam koji sistematično određuje Blazhko kandidate na sljedeći način:

- 01 **Odbacivanje loših podataka:** postoje zvijezde koje imaju izrazito visoku amplitudu (koja je nemoguća) ili vrlo visoke vrijednosti hi-kvadrata koje ukazuju na vrlo loše podatke. Stoga, zvijezde koje imaju amplitudu veću od 2 magnitude, hi-kvadrat vrijednost veću od 9 ili nerealni period koji je veći od 4, zvijezde s manje od 250 promatranja za LINEAR i 40 za ZTF podatke i zvijezde čiji omjer perioda je između 0.8 i 1.2 (veća ili manja vrijednost ukazuje na krivu klasifikaciju zvijezda ili krivo određen ZTF par),
- 02 **Periodogram zvijezde:** ako su svi kriteriji zadovoljeni vezani za pokazatelje Blazhko efekta preko periodograma, zvijezda je automatski Blazhko kandidat. Kriteriji su:
 - a. ne smije postojati dnevni alias perioda (tj. lažni vrh zbog utjecaja dnevnog ciklusa promatranja),
 - b. Blazhko vrh mora imati relativnu snagu veću od 0.05, odnosno mora biti viši od pozadinskog šuma drugih frekvencija,
 - c. statistička značajnost Blazhko vrha mora biti veća od 5,
 - d. Blazhko period odnosno period ponavljanja Blazhko modulacije mora biti između 35 i 325 dana [22,18].
- 03 **Analiza amplitude, perioda i vrijednosti hi-kvadrata:** ako zvijezda nije zadovoljila kriterij analize periodograma, slijedi analiza drugih vrijednosti po kojima možemo prepoznati Blazhko efekt. Velika promjena amplitude ukazuje na **modulaciju amplitude**, dok velika vrijednost hi-kvadrata ili razlika u periodima između LINEAR i ZTF zvijezda ukazuje na **modulaciju perioda**

ili **faze**. Algoritam je građen po principu bodovanja, gdje svaka vrijednost amplitude, hi-kvadrata ili razlike perioda dobiva određeni broj bodova te se kandidat za Blazhko efekt bira po konačnom zbroju bodova.

Kako bismo odredili vrijednosti amplitude, razlike perioda i vrijednosti hi-kvadrata za dobro bodovanje, analizirali smo distribuciju dotičnih vrijednosti u podacima.



Graf 1: distribucija vrijednosti hi-kvadrata

Iz grafa iznad, možemo vidjeti da većinu zvijezda ima vrijednost hi-kvadrata od otprilike 1, što znači da ako je zvijezda Blazhko, mora imati malo veću vrijednost nego većinu podataka. Iduće bodovne granice su birane za vrijednosti hi-kvadrata LINEAR i ZTF podataka:

- 01 Oba hi-kvadrata su unutar svojih granica, vrijedno 4 boda,
- 02 Oba hi-kvadrata su iznad svojih granica, vrijedno 6 bodova,

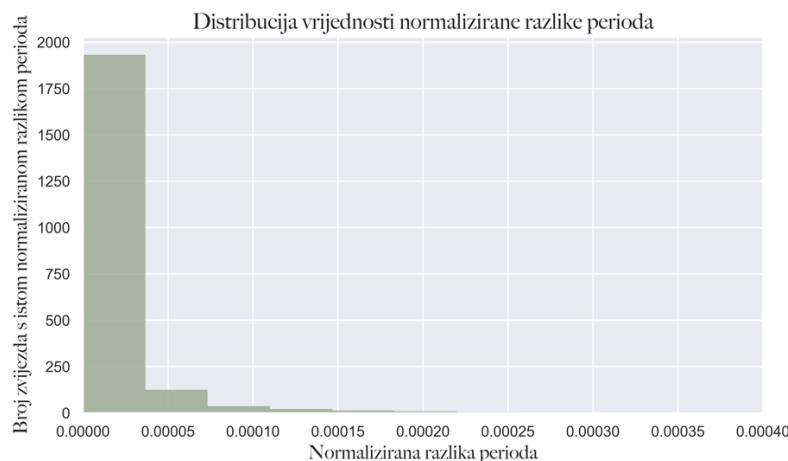
03 LINEAR:

- a. $1.8 < \chi^2 < 3.0$ – vrijedno 2 boda,
- b. $\chi^2 > 3.0$ - vrijedno 3 boda.

04 ZTF:

- a. $2.0 < \chi^2 < 4.0$ – vrijedno 2 boda,
- b. $\chi^2 > 4.0$ – vrijedno 3 boda.

Iduća vrijednost koju je potrebno analizirati je razlika perioda, koja se računa tako da ZTF period oduzmemo od LINEAR perioda te podijelimo sa prosjekom perioda.



Graf 2: distribucija normalizirane razlike perioda

Možemo vidjeti kako su vrijednosti izrazito male te da većinu zvijezda ima gotovo neprimjetnu razliku u periodu. Pomoću grafa iznad, odredili smo iduće granice bodovanja:

- 01** $0.00002 < razlika\ perioda < 0.00005$, vrijedno 2 boda,
- 02** $0.00005 < razlika\ perioda$, vrijedno 4 boda.

Zadnje, odredili smo iduću bodovnu skalu za promjenu amplitude, koju možemo odrediti i bez grafa:

- 01** $0.05 < amplituda < 0.15$, vrijedno 1 bod,
- 02** $0.15 < amplituda < 2.00$, vrijedno 2 boda.

Konačna bodovna skala imala je maksimum od 12 mogućih postignutih bodova za svaku svjetlosnu krivulju te su birane sve zvijezde s više od 4 bodova (od 5 nadalje). Pomoću algoritma birano je 239 potencijalnih Blazhko zvijezda. Važno je napomenuti da algoritam nije savršen te da neke Blazhko zvijezde algoritam neće moći prepoznati jer se vrijednosti biranih parametara ne razlikuju od zvijezda bez Blazhko efekta. Algoritam služi kako bi efikasno odredili podskup zvijezda s velikim postotkom Blazhko zvijezda tako da što manje zvijezda treba izrazito detaljno pregledavati, što je i cilj ovog algoritma.

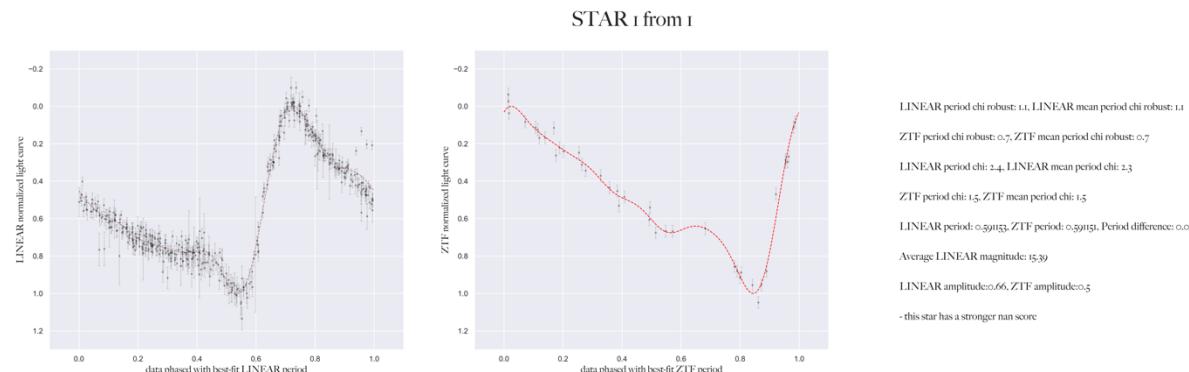
2.6 Analiza Blazhko kandidata

Nakon biranje Blazhko kandidata pomoću algoritma objašnjenog iznad, potrebno je vizualno analizirati 239 Blazhko kandidata određenih pomoću prethodno objašnjenog algoritma i odrediti konačnu listu Blazhko zvijezda. Kako bismo uspješno analizirali Blazhko kandidate, potrebno je analizirati nekoliko grafova. Vizualna analiza sastoji se od jednog Python Class objekta gdje za svaku zvijezdu crta se 4 grafa pomoću kojih provodim vizualnu analizu. Ako je zvijezda Blazhko zvijezda, može se pritisnuti gumb „KEEP“ za nastavak na iduću zvijezdu te se podaci dotične zvijezde spremaju u drugu tablicu ili se može pritisnuti gumb „CONTINUE“ koji nastavlja crtanje grafova za iduću zvijezdu.

Kako bi analiza zvijezda bila što rigoroznija, podijelili smo Blazhko kandidate u 5 grupa:

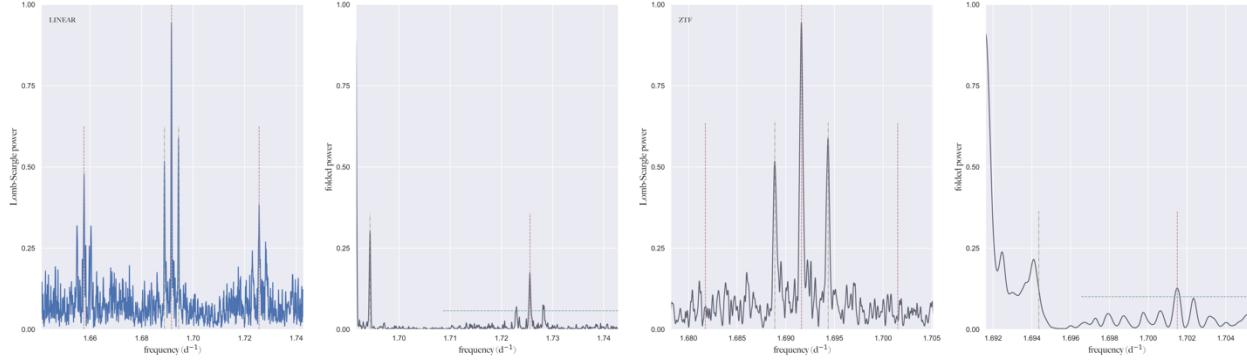
- 01 **LINEAR periodogrami:** svi Blazhko kandidati koji su prema LINEAR periodogramu bili izabrani kao kandidat,
- 02 **ZTF periodogrami:** svi Blazhko kandidati koji su prema ZTF periodogramu bili izabrani kao kandidat,
- 03 **LINEAR hi-kvadrat:** svi Blazhko kandidati koji su prema vrijednosti LINEAR hi-kvadrata bili izabrani kao kandidat,
- 04 **ZTF hi-kvadrat:** svi Blazhko kandidati koji su prema vrijednosti ZTF hi-kvadrata bili izabrani kao kandidat,
- 05 **Ostalo:** svi kandidati koji nisu zadovoljili nijedan kriterij za grupe iznad.

Vizualna analiza provodi se kroz 4 faza promatranja 4 različitih grafova prikazani u sučelju. Prva faza je analiza faze RR Lira LINEAR i ZTF podataka te pregledavanja razlike u periodu.



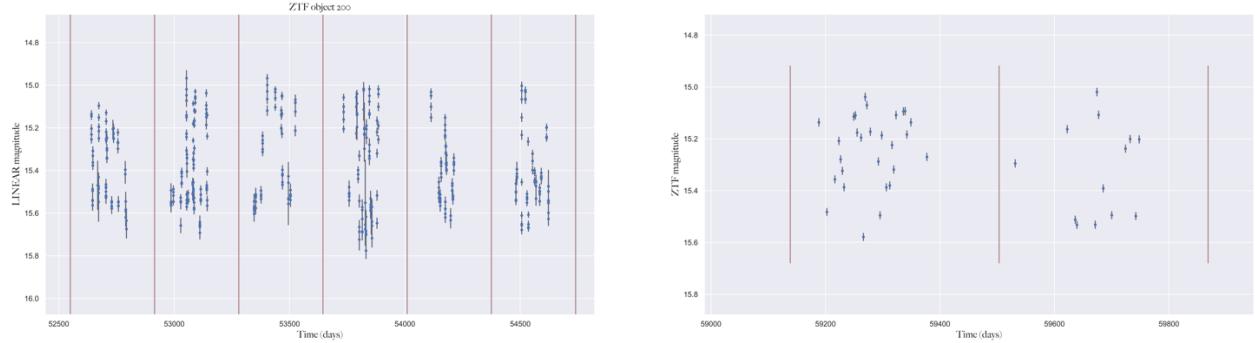
Slika 10: prva faza vizualne analize Blazhko kandidata

U prvoj fazi provjerava se ispravnost svjetlosne krivulje, je li oblik sličan obliku RR Lira, imaju li podaci previše grešaka u promatranju te je li razlika u periodu prevelika. Ako su svi prethodni parametri zadovoljeni, kao što je prikazano u slici iznad, prijelazi se na drugu fazu.



Slika 11: druga faza vizualne analize Blazhko kandidata

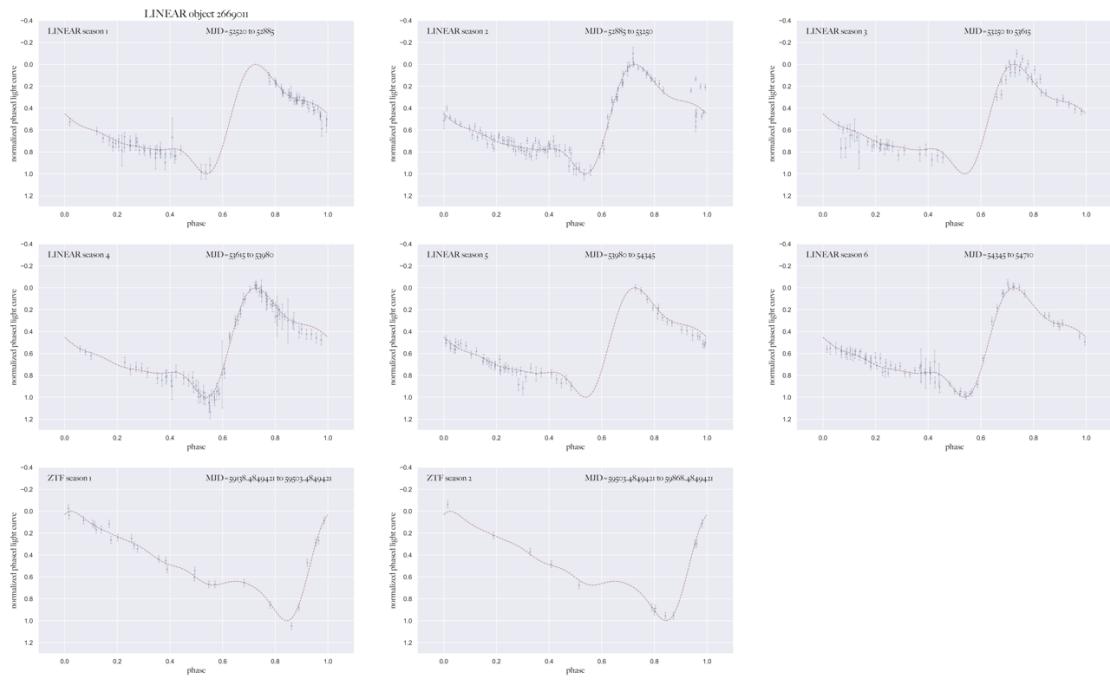
Druga faza uključuje provjeravanja točnosti algoritma za periodogram. Na slici iznad vidimo da LINEAR periodogram (lijevo) je lijep primjer Blazhko efekta, s 2 blazhko vrha jednako udaljenih od glavne frekvencije označeni crvenom linijom, što je vidljivo i na grafu odmah do koji prikazuje pomnožene strane periodograma ([§2.4.2](#)). Također provjeravamo je li visina vrhova statistički značajna u odnosu na pozadinu, što je istina za LINEAR primjer iznad. Kod ZTF periodograma vidimo 2 lijepa vrha, no oni se nalaze na frekvencijama koje su *frekvencije godišnjeg aliasa*, stoga nisu pravi signali. ZTF periodogram ne prikazuje nikakav značajni vrh, što je vidljivo na oba grafa. Ako jedan ili oba periodograma ne prikazuju Blazhko efekt, ne znači da mala modulacija amplitude nije prisutna, stoga je uvijek potrebno do kraja pogledati sve grafove na sučelju te nastaviti s fazom 3.



Slika 12: treća faza vizualne analize Blazhko kandidata

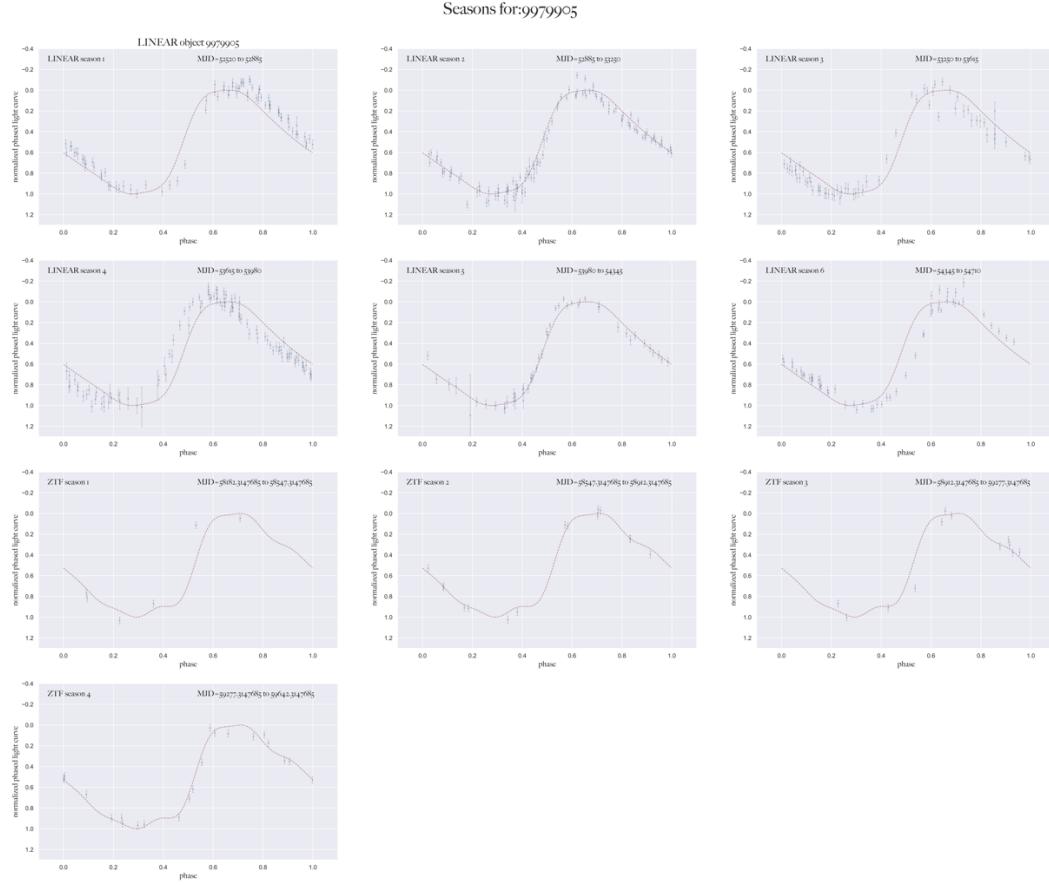
Treća faza uključuje promatranje ukupne sezone promatranja LINEAR i ZTF podataka za pojedinu zvijezdu. Potrebno je vidjeti je li magnituda konstantna tokom čitavog vremenskog perioda promatranja ili postoji modulacija amplitude. U primjeru iznad, za LINEAR podatke možemo vidjeti da postoji blaga razlika u amplitudi između sezona te da **ne** formiraju pravokutnik. Kod ZTF podataka ne vidimo značajne razlike u magnitudi te možemo aproksimirati oblik svih promatranja kao pravokutnik.

Seasons for:2669011



Slika 13: četvrta faza vizualne analize Blazhko kandidata, modulacija amplitude

U zadnjoj fazi vizualne analize potrebno je pogledati kako se svjetlosna krivulja mijenjala tokom vremenskog razdoblja po sezonomu. Gleda se postoji li **modulacija amplitude** ili **fazna modulacija**. U primjeru iznad vidljiva je blaga modulacija amplitute, gdje su točke pomaknute u jednom trenutku na najboljem *fitu*, u drugom ispod, a u trećem iznad *fita*. Translacija podataka po y-osi je znak modulacije amplitute.



Slika 14: četvrta faza vizualne analize Blazhko kandidata, fazna modulacija

Fazna modulacija je prikazana kao promjena u periodu, odnosno translacija grafa po x-osi u odnosu na najbolji *fit*. Na slici iznad možemo vidjeti odličan primjer fazne modulacije gdje se mijenja pozicija svih točaka svjetlosne krivulje u odnosu na *fit* kako prolaze sezone promatranja.

Ako zvijezda koju promatramo preko sučelja nema oblik RR Lire ili ima 2 pada svjetlosti, odmah se odbacuje kao Blazhko kandidat budući da nije tip zvijezda koji može imati Blazhko efekt. Iako smo na početku izabrali samo promjenjive zvijezde klasificirane kao RR Lire, moglo je doći do krive klasifikacije zvijezda u Geneva setu podataka.

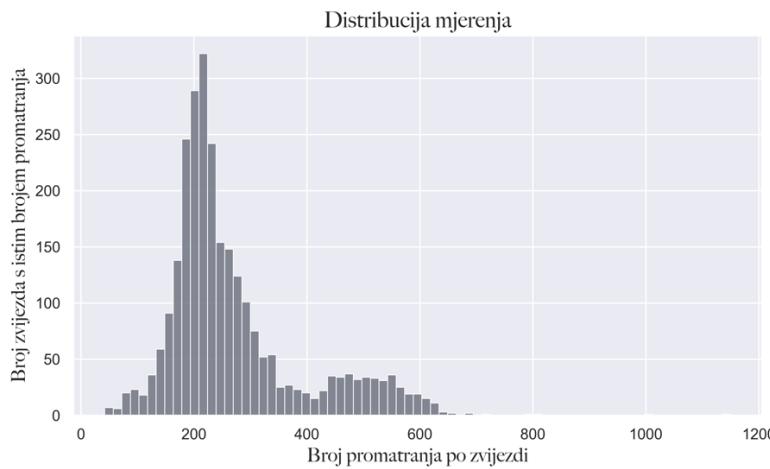
Zvijezda se može priznati kao Blazhko zvijezda ako prikazuje ili modulaciju preko grafa sezona ili ako ima jako lijep primjer periodograma. Ako periodogram prikazuje dobre blazhko vrhove s pozadinskim šumom te je blaga modulacija prisutna u grafu sezona, također se zvijezda može priznati kao Blazhko zvijezda.

Nakon prvotne analize svih Blazhko kandidata provodi se još jedna vizualna analiza izabranih kandidata istim postupkom kako bismo došli do konačne liste Blazhko zvijezda pronađene izvan skupova zvijezda.

3 REZULTATI

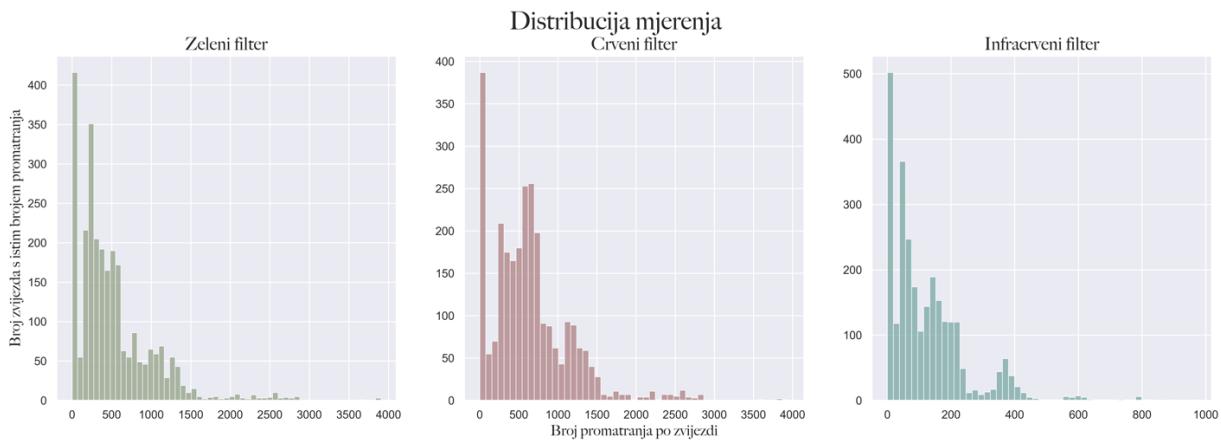
3.1 Analiza RR Lira

Osnovnom analizom LINEAR i ZTF podataka uočavamo da LINEAR ima, kada bismo zbrojili broj mjerena u svakoj svjetlosnoj krivulji RR Lira, 733,550 mjerena, dok ZTF parovi imaju 1,423,997 mjerena. Mnogo veći broj mjerena kod ZTF podataka je prisutan zbog strukture ZTF podataka: sadrži mjerena u više filtera, točnije 3 filtera. Stoga, crveni filter ima 1,772,818 mjerena, 1,423,997 mjerena u zelenom i 337,132 mjerena u infracrvenom filteru. Iz idućeg grafa možemo vidjeti distribuciju mjerena po svjetlosnoj krivulji za LINEAR zvijezde.



Graf 3: raspodjela mjerena kod LINEAR RR Lira

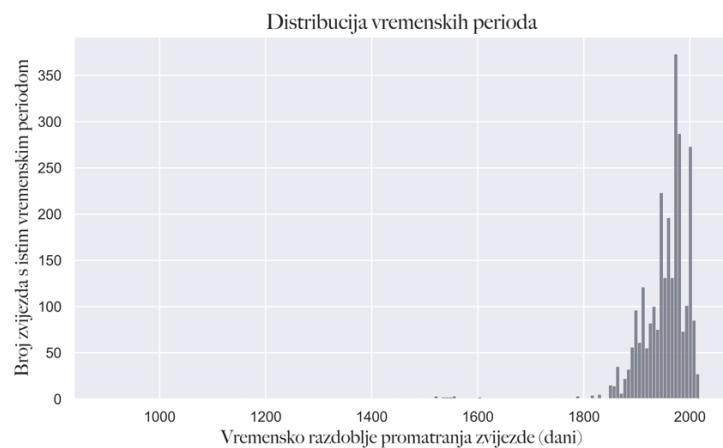
Uočavamo da većina zvijezda ima malo iznad 200 mjerena po krivulji, s uzastopnim porastom u brojnosti za otprilike 500 mjerena po krivulji. Zapažanje je vrlo pozitivno, budući da su za analizu potrebne LINEAR zvijezde s više od 250 mjerena po krivulji za pouzdanu analizu. Idući dijagram prikazuje distribuciju mjerena za sve ZTF filtere:



Graf 4: raspodjela mjerena kod ZTF RR Lira

Vidimo da distribucije svih filtera imaju sličan oblik, međutim infracrveni grafikon ima manji raspon x-osi. Zeleni filter najviše zvijezda ima broj mjerena manje od otprilike 100 promatranja te otprilike 400-500 promatranja. Vrlo slična situacija je s crvenim filtrom, s nešto višim brojem opažanja, otprilike 500-600. Infracrveni filter ima najmanji broj točaka, s većinom zvijezda ispod otprilike 50 promatranja te 100 točaka.

Drugi osnovan aspekt samih svjetlosnih krivulja je duljina promatranja dotične zvijezde. Zvijezda nije promatrana čitavo vrijeme, nego u kraćim sezonomama ili u intervalima od jedne noći, budući da se ne promatra tokom dana. Iduća 2 dijagrama prikazuju razliku u vremenskom razdoblju promatranja kod LINEAR i ZTF parova:

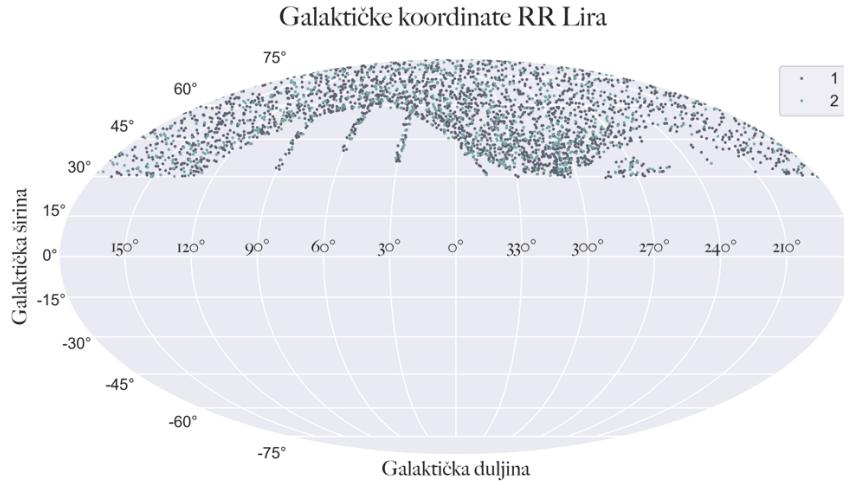


Graf 5: distribucija vremenskih perioda LINEAR zvijezda



Graf 6: distribucija vremenskih perioda ZTF zvijezda po filterima

Uočavamo kako kod LINEAR zvijezda, većina zvijezda ima od 1900 do 2000 dana vremenski period promatranja. Slična je situacija i kod ZTF zvijezda, no vidimo da postoji mnogo zvijezda sa izrazito kratkim periodom promatranja ili gdje je period jednak 0. Zapažanje perioda jednakim 0 objašnjavamo činjenicom da ne postoji ZTF par za svaku LINEAR zvijezdu, stoga ni neće imati vrijeme promatranja.



Graf 7: prostorna distribucija RR Lira po galaksiji

Možemo vidjeti kako RRab i RRc nisu grupirane u posebna područja nego su ravnomjerno raspršene po nebu te da se sve zvijezde nalaze na galaktičkoj širini višoj od 30° . Zapažanje pokazuje da zvijezde se ne nalaze u nekom posebnom skupu ni prema središtu galaksije.

3.2 Analiza RR Lira i pronalazak Blazhko zvijezda

Nakon osnovne analize RR Lira, bilo ih je potrebno analizirati metodama objašnjenim u §2 ovog rada. Od početnih 2914 RR Lira zvijezda, analizom njihovih svjetlosnih krivulja preko izračuna perioda, analizom periodograma te izračunom vrijednosti hi-kvadrata, selektirano je 2857 zvijezda.

Pomoću algoritma za prepoznavanje Blazhko kandidata, selektirano je 239 zvijezda. Zatim su one bile analizirane vizualno, procesom objašnjenim u §2.6. Nakon vizualne analize, bilo je selektirano sveukupno 136 Blazhko zvijezda kao konačne zvijezde iz početnog skupa 2914 RR Lira, od koji smo mogli obraditi samo 2857 zbog dostupnosti podataka. Tablica ispod prikazuje 10 primjera Blazhko zvijezda pronađenih u LINEAR i ZTF podacima te neke njihove osnovne podatke:

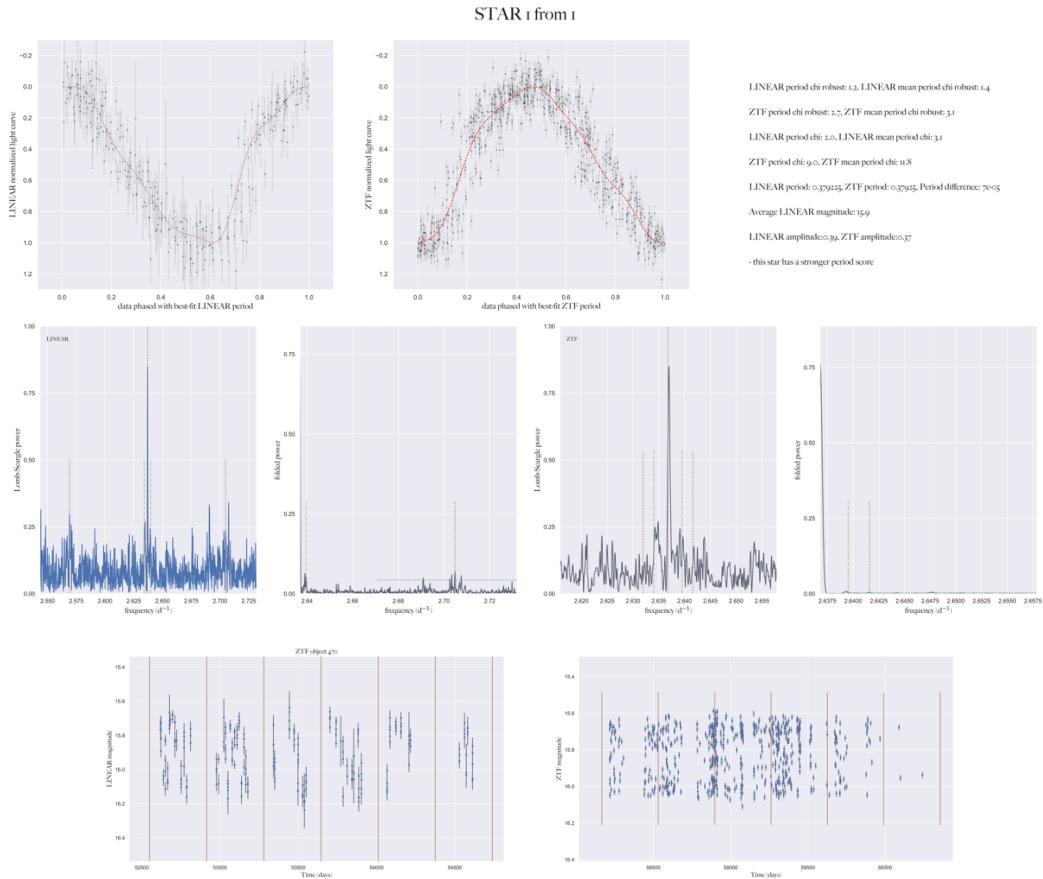
Tablica 2: 10 Blazhko zvijezda i njihovi osnovni podaci

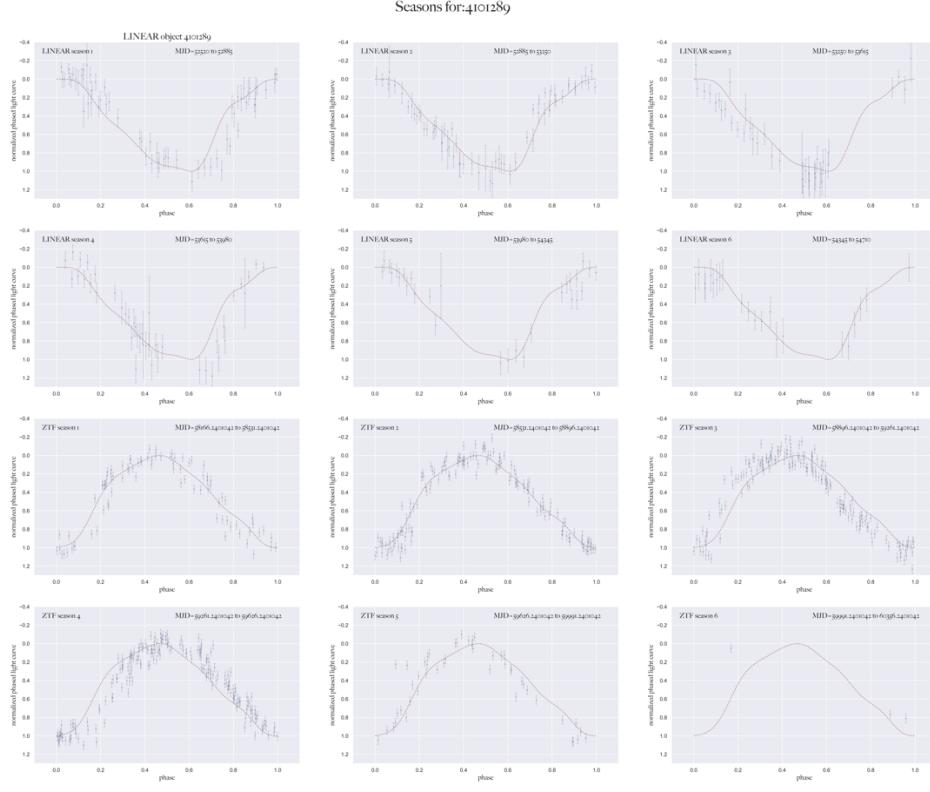
	LINEAR id	LINEAR period	ZTF period	LINEAR hi-kvadrat	ZTF hi-kvadrat	LINEAR amplituda	ZTF amplituda	LINEAR glavna frekvencija	LINEAR blazhko frekvencija	ZTF glavna frekvencija	ZTF blazhko frekvencija	LINEAR blazhko period	ZTF blazhko period
0	523832	0.372376	0.372384	1.2	1.1	0.42	0.59	2.6855	2.8040	2.6854	2.7122	8.4370	37.3413
1	1240665	0.632528	0.632522	3.0	1.1	0.33	0.33	1.5810	1.6149	1.5810	1.5865	29.4942	182.3154
2	1736308	0.555848	0.555843	1.3	1.0	0.70	0.59	1.7991	1.8330	1.7991	1.8264	29.4377	36.6367
3	2669011	0.591153	0.591151	1.1	0.7	0.66	0.50	1.6916	1.7256	1.6916	1.7015	29.4464	101.2146
4	2742032	0.629676	0.629692	0.9	1.4	0.55	0.89	1.5881	1.6221	1.5881	1.5956	29.4681	133.2445
5	2812086	0.646015	0.646000	3.0	3.2	0.61	0.58	1.5480	1.5781	1.5480	1.5519	33.1730	258.7322
6	3507643	0.801141	0.801132	1.6	0.9	0.37	0.43	1.2482	1.2821	1.2482	1.2538	29.5290	181.1594
7	5931160	0.664700	0.664708	0.8	1.1	0.35	0.33	1.5044	1.5383	1.5044	1.5078	29.5421	293.6858
8	6665721	0.643318	0.643325	1.0	1.7	0.46	0.45	1.5544	1.5883	1.5544	1.5669	29.5159	79.9680
9	17185566	0.614160	0.614169	1.5	1.9	0.40	0.42	1.6282	1.6620	1.6282	1.6365	29.5814	120.6273

3.2.1 Detaljan opis 3 Blazhko zvijezde: 4101289, 6819457, 7048826, 10260828

U ovom dijelu opisujemo 4 primjera Blazhko zvijezda za ilustraciju načina biranja Blazhko zvijezda te njihovih značajki.

3.2.1.1 Zvijezda 4101289



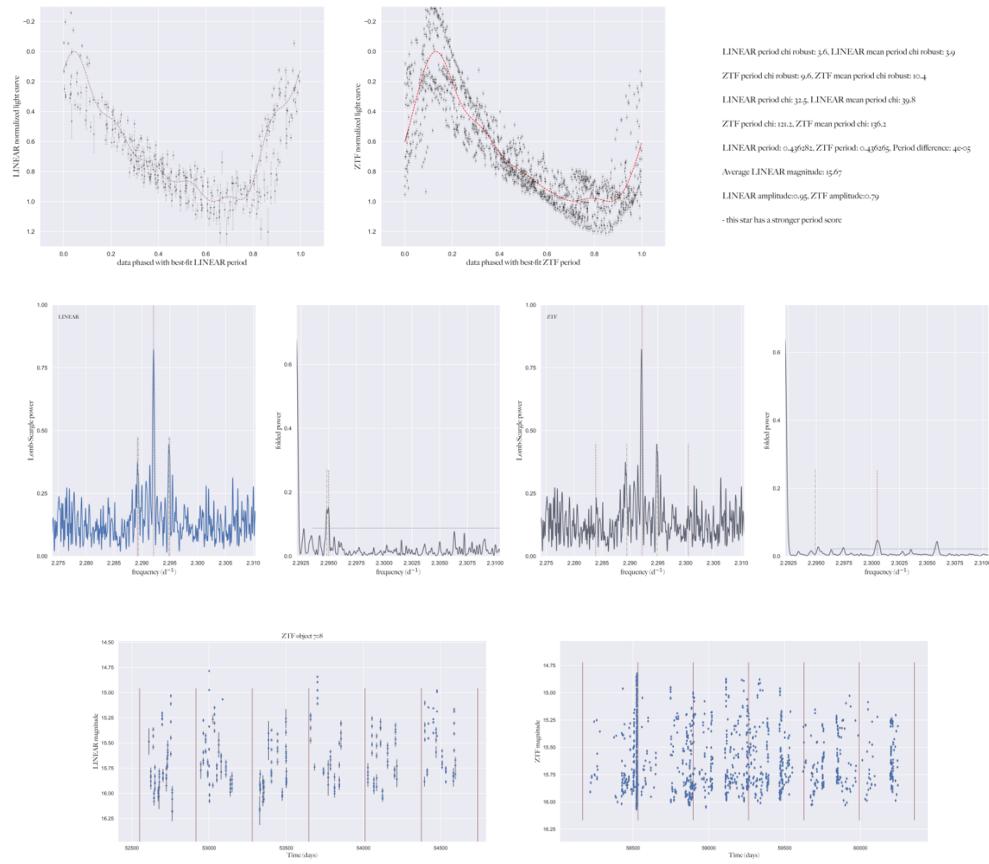


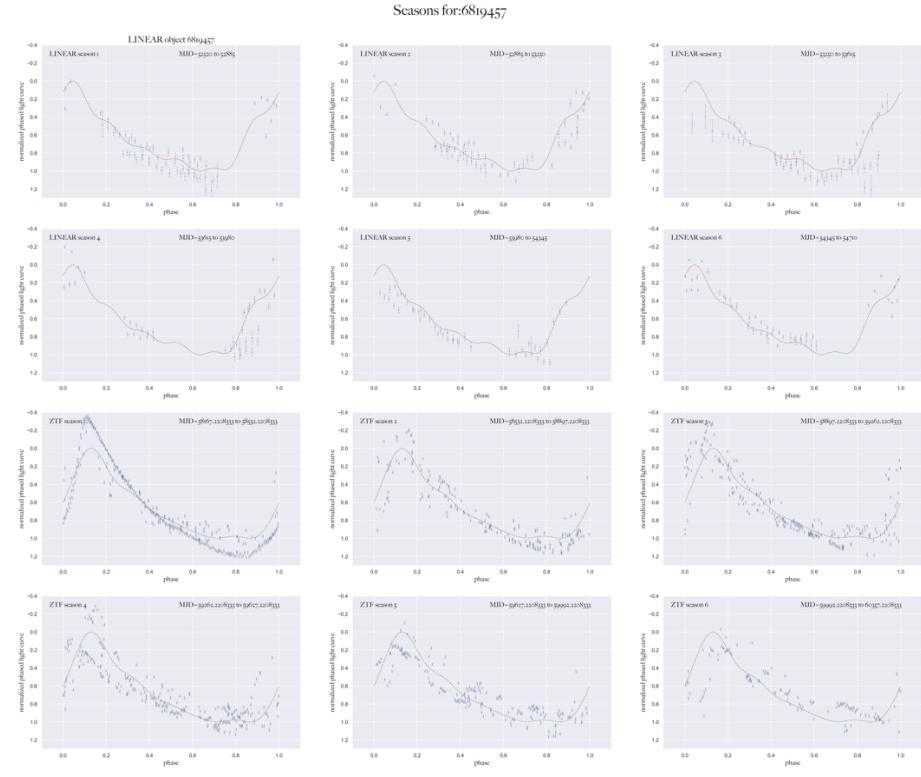
Slika 15: sučelje za vizualnu analizu Blazhko zvijezde 4101289

Iz slike iznad možemo primjetiti da periodogram i kod LINEAR i kod ZTF podataka ne prikazuje signifikantnu prisutnost Blazhko efekta, no to ne znači da zvijezda nije Blazhko. Iz dijagonala za sezone promatranja LINEAR i ZTF podataka sveukupno, vidimo da kod LINEAR podataka ima naznake promjene amplituda. Kada pobliže pogledamo zadnju sliku koja prikazuje pojedinačne sezone, očita je vrlo snažna modulacija perioda i u LINEAR i u ZTF podacima.

3.2.1.2 Zvijezda 6819457

STAR 1 from 1



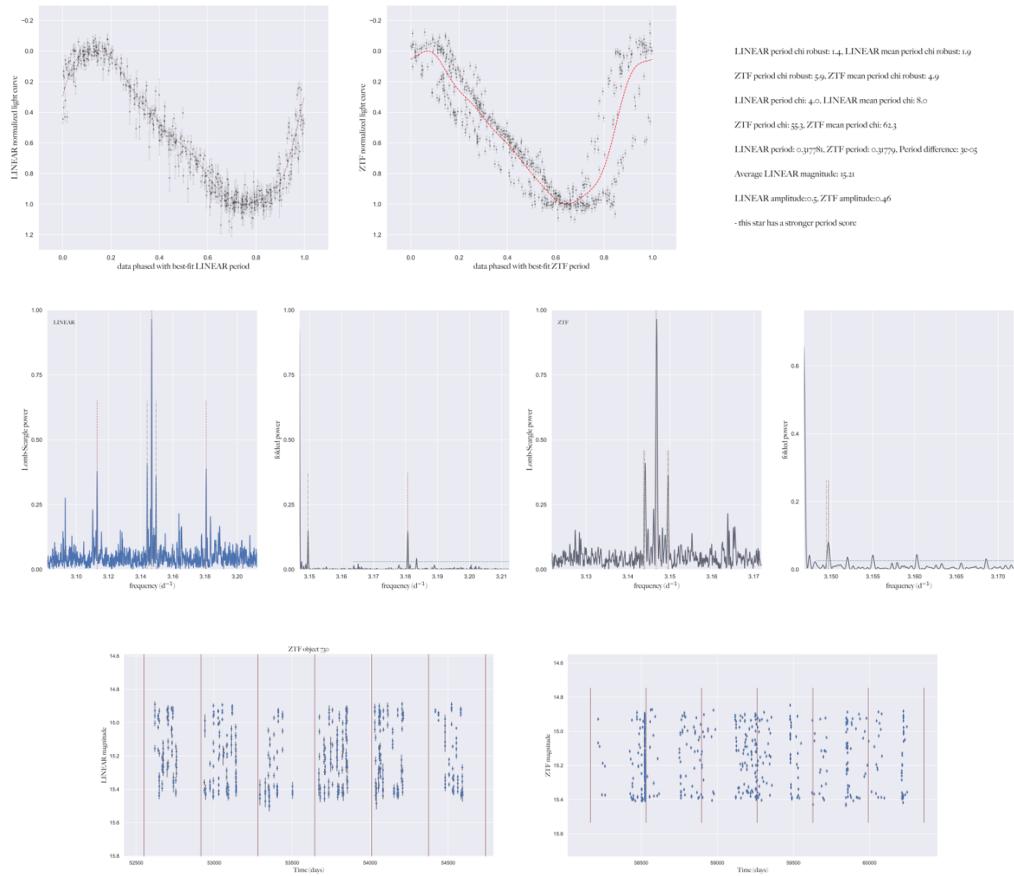


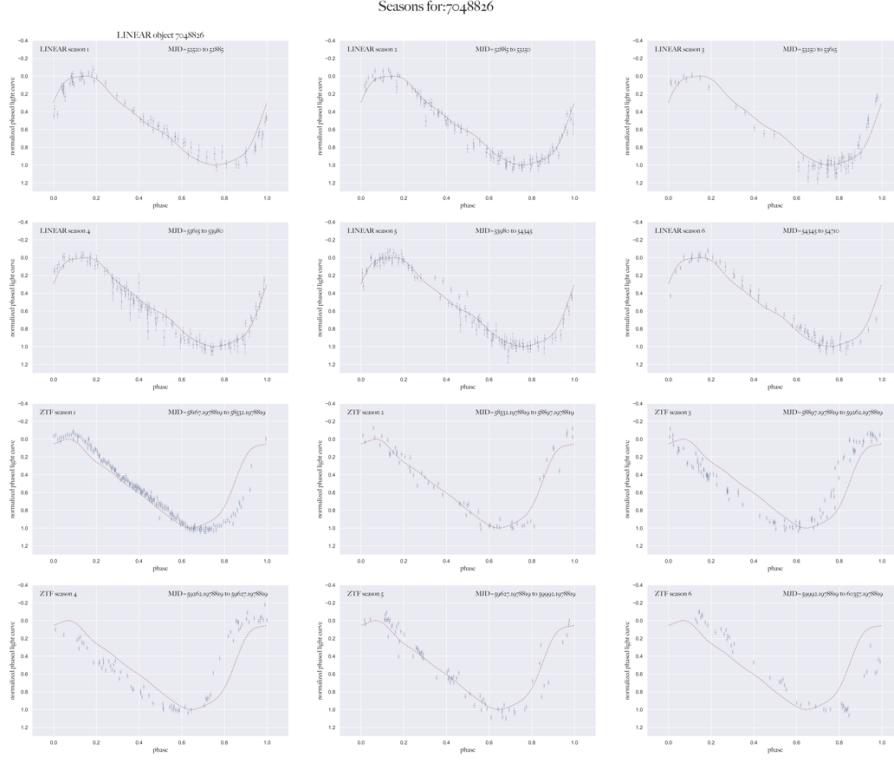
Slika 16: sučelje za vizualnu analizu Blazhko zvijezde 6819457

Samо gledajući prikaz jedne faze svjetlosne krivulje zvijezde možemo vidjeti modulaciju amplitude kod ZTF podataka. Iako periodogrami ne prikazuju Blazhko frekvencije, gledajući krivulje po sezona ma promatranja možemo uočiti izrazito snažnu modulaciju amplitude kod ZTF podataka.

3.2.1.3 Zvijezda 7048826

STAR 1 from 1

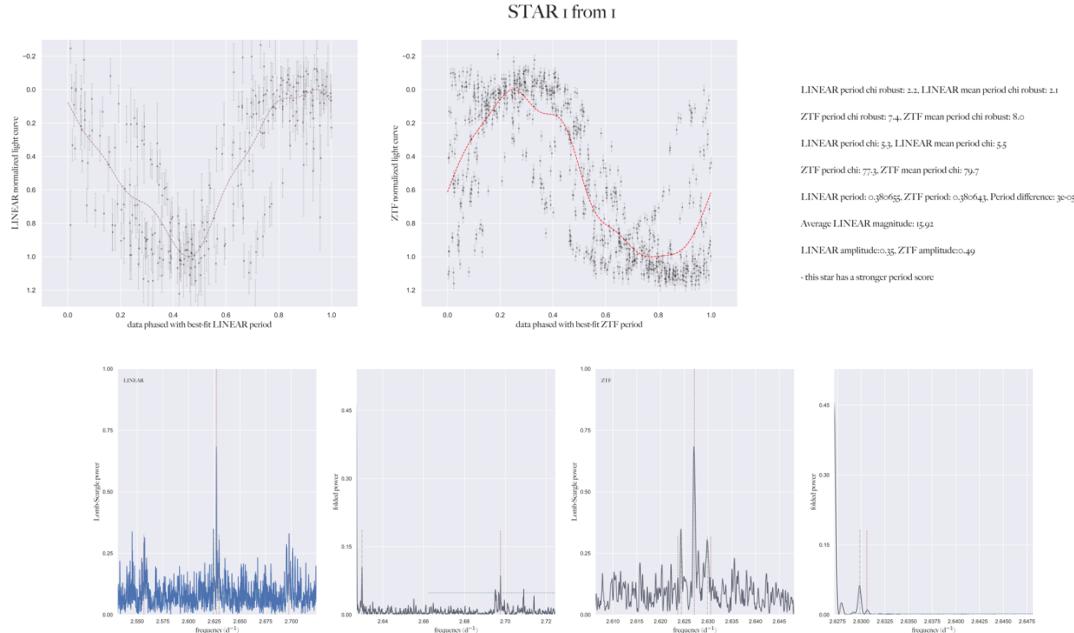


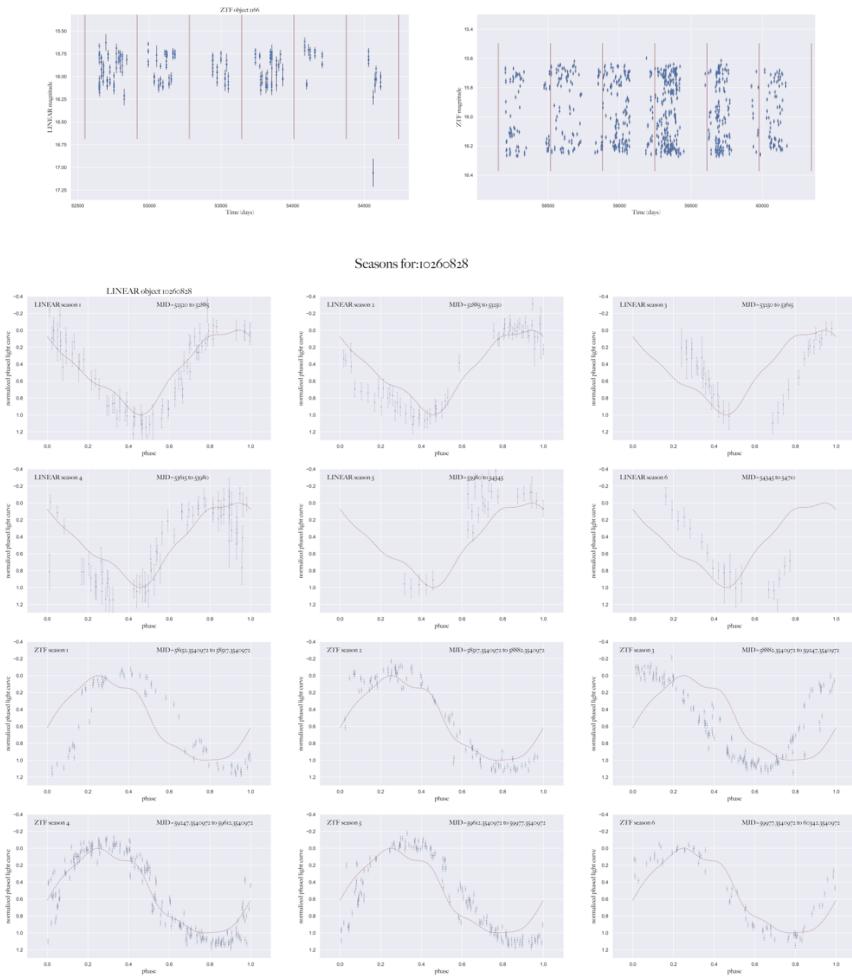


Slika 17: sučelje za vizualnu analizu Blazhko zvijezde 7048826

U ovom primjeru, LINEAR periodogram je vrlo lijep i prikazuje lokalne Blazhko frekvencije, dok ZTF periodogram ne prikazuje naznake Blazhko efekta. Gledajući sezonska promatranja zvijezde, uočavamo modulaciju amplitude kod LINEAR, a faznu modulaciju kod ZTF podataka.

3.2.1.4 Zvijezda 10260828



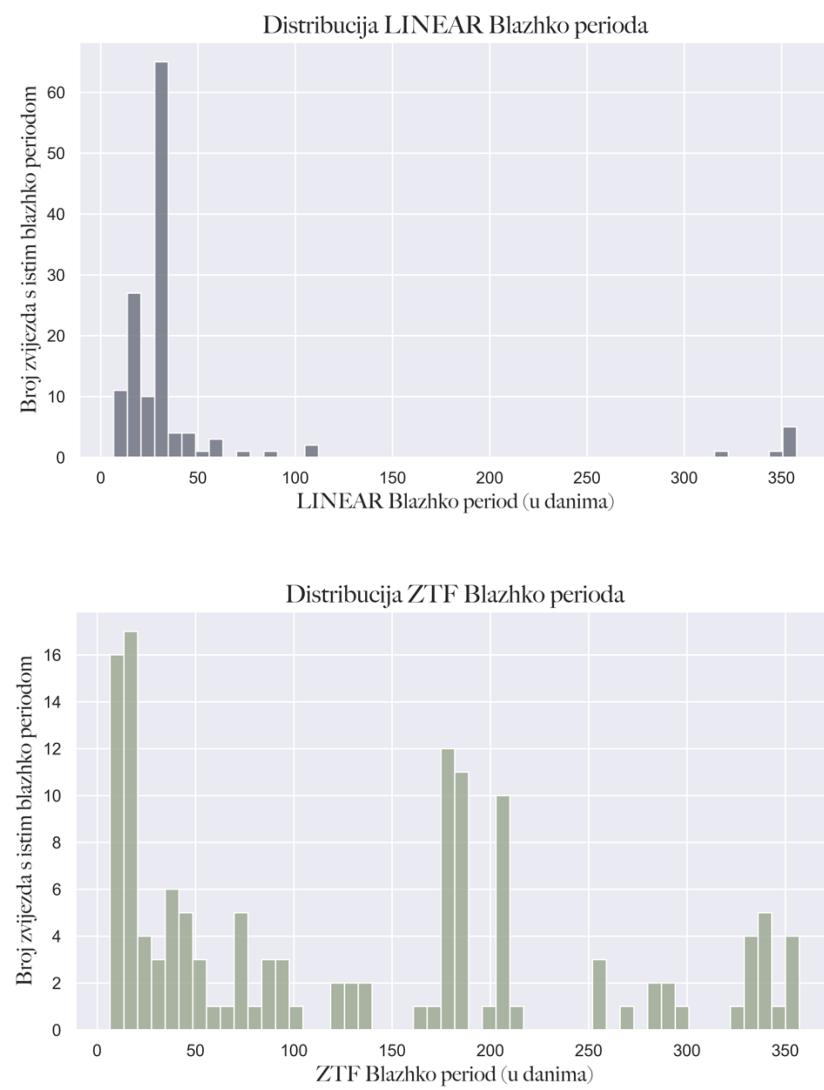


Slika 18: sučelje za vizualnu analizu Blazhko zvijezde 10260828

Zvijezda 10260828 nema naznaka Blazhko efekta kod ZTF periodograma, no ima naznake Blazhko frekvencijama kod LINEAR periodograma. Gledajući sezonske svjetlosne krivulje zvijezde, možemo vidjeti vrlo očitu modulaciju faze kod i kod LINEAR i kod ZTF podataka, prikazujući fantastičan primjer Blazhko zvijezde.

4 RASPRAVA I ZAKLJUČAK

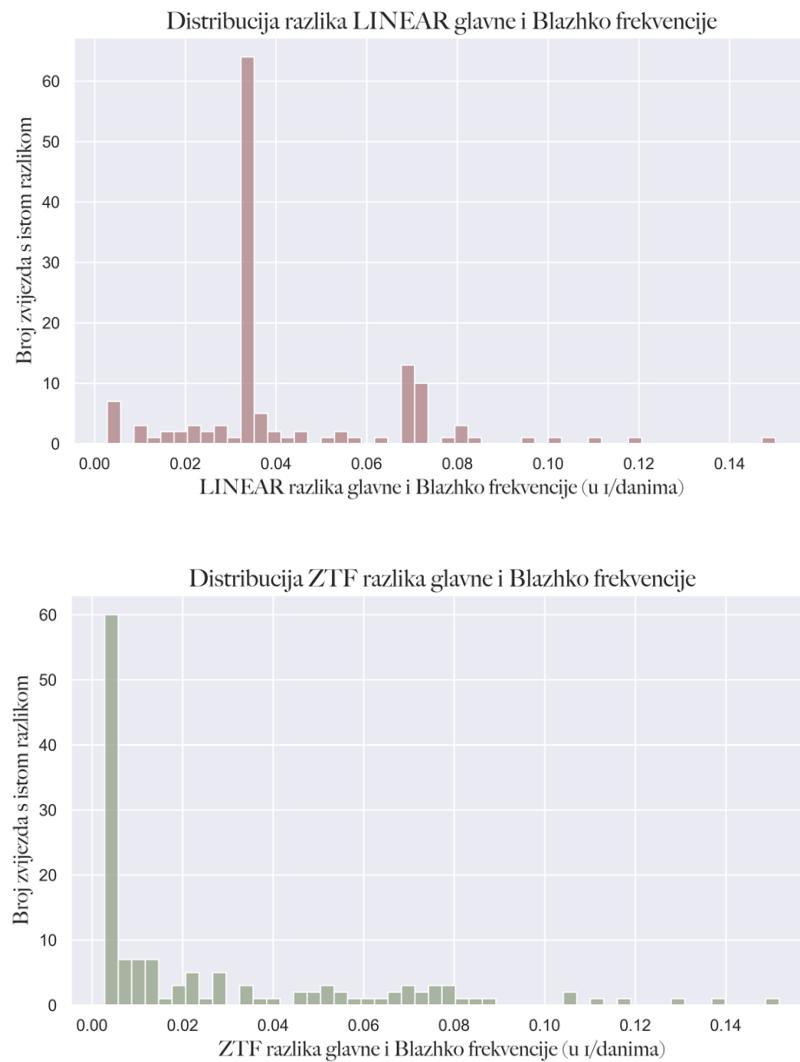
U ovom praktičnom radu promatrali smo svjetlosne krivulje RR Lira pomoću 2 različita skupa podataka: **LINEAR** i **ZTF** pregleda neba. Razlika u vremenu promatranja ova 2 pregleda neba je otprilike 10 godina, što nam daje uvid u dugoročnu promjenu svjetlosne krivulje pojedine zvijezde. Počeli smo s 7010 promjenjivih zvijezda iz LINEAR podataka [30, 31] te pomoću klasifikacije zvijezda odabrali smo 2914 RR Lira i napravili ZTF bazu podataka. Analizom podataka pomoću raznih algoritama te vizualnom analizom, odredili smo **136 Blazhko zvijezda**, odnosno oko 4.67 % od početnog skupa podataka za RR Lire.



Graf 8: distribucija LINEAR i ZTF Blazhko perioda

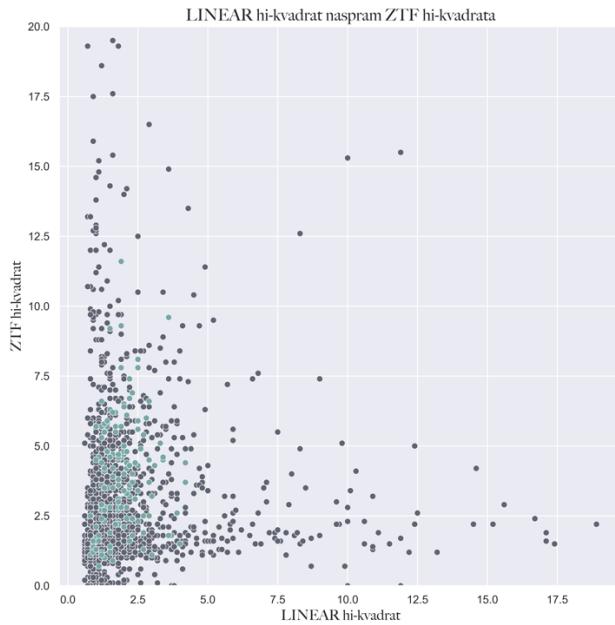
Većina Blazhko zvijezda ima mali Blazhko period, manji od 50 dana što je u skladu sa distribucijom perioda RR Lira jer što je manji period RR Lire i njen Blazhko period je manji. Vidljiv je drugi vrh u

distribuciji kod ZTF podataka od oko 150 dana, što je zanimljivo opažanje. Pobliže možemo analizirati razliku glavne frekvencije (period) i Blazhko frekvencije (Blazhko period) pomoću idućih grafova:



Graf 9: LINEAR i ZTF razlike glavnih i Blazhko frekvencija

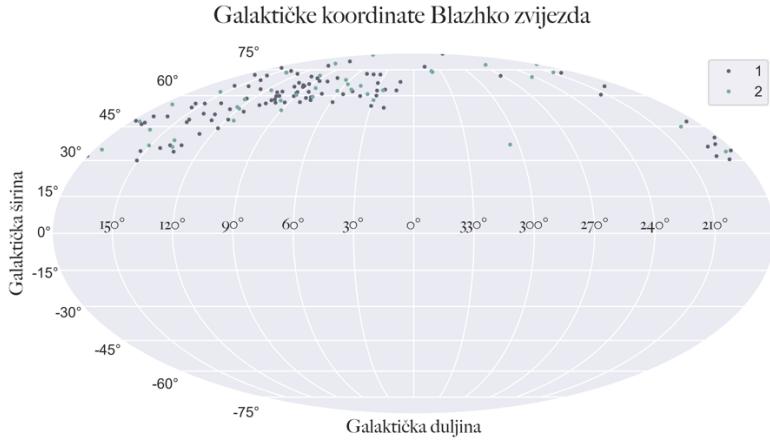
Možemo vidjeti da su razlike u frekvencijama izrazito male, gdje LINEAR ima najviše zvijezda s otprilike 0.035 razlike u frekvencijama, dok ZTF ima najviše zvijezda sa razlikom manjom od 0.02. Zanimljivo je koliko male razlike u frekvencijama uzrokuju vidljive modulacije svjetlosne krivulje.



Graf 10: točkasti dijagram LINEAR i ZTF vrijednosti hi-kvadrata

Još jedan prikaz kako male razlike u periodu pulsiranja same zvijezde uzrokuje promjene u svjetlosnoj krivulji prikazuje Graf 8. Tamnoplava boja prikazuje vrijednosti hi-kvadrata svih RR Lira, dok svjetloplava boja prikazuje Blazhko zvijezde. Većina razlika u periodu su manje od 0.0001 dana, a razlike u vrijednostima hi-kvadrata su primjetne i signifikantne. Također možemo uočiti kako većina Blazhko zvijezda imaju niske vrijednosti hi-kvadara za LINEAR podatke no više vrijednosti za ZTF podatke.

Od 136 novootkrivenih Blazhko zvijezda, 71.32 % su RRab tipa, a 28.68 % su RRc tipa. Udio RRab i RRc tipa nije se mnogo promjenio između Blazhko skupa zvijezda i početnog skupa zvijezda te je u skladu sa drugim istraživanjima [31, 38], udio RRab mnogo veći od udjela RRc tipa.



Graf 11: pozicije Blazhko zvijezda pomoću galaktičkih koordinata

Gledajući prostorni raspored Blazhko zvijezda po galaksiji, većina ih se nalazi između 0° i 150° , iznad 30° galaktičke širine. Mali dio zvijezda nalaze se nakon 240° galaktičke dužine, većinski RRab tipovi zvijezda.

Konačno, zaključujemo da je naš glavni cilj ovoga rada ispunjen te smo uspješno pronašli velik broj RR Lira s Blazhko efektom. Od početnog skupa podataka isključivo RR Lira, 4.67 % su Blazhko. Neka druga istraživanja pomoću Kepler teleskopa [23] s izrazito preciznim podacima koji se ne mogu reproducirati na Zemlji ili Konkoly pretraga neba [24] pokazuju da su 30-47 % početnih RR Lira Blazhko zvijezde. U usporedbi s ovim praktičnim radom, veličina početnog seta RR Lira tih istraživanja mnogo je manji, otprilike 50 zvijezda, dok u našem istraživanju početna količina RR Lira mnogo je veća, s 2914 zvijezda. Kada bismo gledali druge preglede neba s mnogo većom početnom količinom podataka, otprilike 10000 RR Lira [28], postotak Blazhko zvijezda je 5.6 %. Zaključujemo da je naš udio Blazhko zvijezda u skladu sa istraživanjima s velikim početnim brojem RR Lira. Neki drugi faktori su preciznost podataka, gdje su LINEAR podaci mnogo manje precizni od npr. Kepler podataka. Osim samih podataka, promatrali smo sasvim različit dio neba, zvijezde koje nisu dio nekog skupa zvijezda, dok su druga istraživanja gledala kuglaste skupove ili središte galaksije. Faktor opažanja različitog dijela neba isto može utjecati na postotak prisutnih Blazhko zvijezda. Također, moramo uzeti i u obzir nepregledane zvijezde koje algoritam nije selektirao tokom procesa odabira Blazhko kandidata koje su možda i Blazhko zvijezde. Neovisno o poteškoćama tokom potrage za Blazhko zvijezdama, smatramo da smo uspjeli pronaći veliki broj Blazhko zvijezda. Naš praktičan rad iskoristio je jedinstvenu mogućnost pregledavanja dosada nepregledanog dijela neba pomoću 2 seta podataka koji su razmaknuti vremenski 10 godina, dajući novu perspektivu u potrazi za Blazhko zvijezdama.

U budućnosti želimo nastaviti analizu ovih Blazhko zvijezda te gledati postoje li neki trendovi među pronađenih 136 zvijezda tako što analiziramo još jednu bazu podataka koji imaju podatke LINEAR

zvijezda koje smo analizirali. Također bismo mogli istražiti mogućnosti neuronske mreže koja bi tražila Blazhko zvijezde umjesto da vizualno moramo analizirati svaku zvijezdu zasebno ili neke druge metode određivanja fazne i amplitudne modulacije. Zasada, ovaj praktičan rad služi kao dobar primjer za automatizaciju traženje Blazhko zvijezda za buduće teleskope kao LSST i dr. koji bi mogli unaprijediti naše znanje o Blazhko efektu i RR Lirama s velikom količinom podataka koju će producirati.

5 LITERATURA I IZVORI

- 01 https://chandra.harvard.edu/graphics/edu/earth_scientist_stars.pdf (3.8.2023.)
- 02 <https://openstax.org/books/astronomy-2e/pages/19-3-variable-stars-one-key-to-cosmic-distances> (4.7.2023.)
- 03 <http://spiff.rit.edu/classes/phys370/lectures/mags/mags.html> (3.3.2024.)
- 04 http://spiff.rit.edu/classes/phys370/lectures/pulse_ii/pulse_ii.html (4.3.2024.)
- 05 http://spiff.rit.edu/classes/phys370/lectures/pulse_i/pulse_i.html (4.3.2024.)
- 06 <http://spiff.rit.edu/classes/ladder/lectures/varstars/varstars.html#pulse> (5.3.2024.)
- 07 https://web.physics.ucsb.edu/~jatila/LambdaLabs/Globulars/HRdiagramlab_JKV.pdf (7.3.2023.)
- 08 <https://articles.adsabs.harvard.edu/full/1992ApJS...79..507R/0000512.000.html> (5.3.2024.)
- 09 <https://astronomy.swin.edu.au/cosmos/h/hertzsprung-russell+diagram> (6.3.2024.)
- 10 <https://www.animations.physics.unsw.edu.au/jw/beats.htm> (21.2.2024.)
- 11 https://phys.libretexts.org/Courses/University_of_California_Davis/UCD%3A_Physics_7C_-_General_Physics/8%3A_Waves/8.6%3A_Beats (21.2.2024.)
- 12 <https://hrcak.srce.hr/file/175513> (12.2.2024.)
- 13 <https://home.ifa.hawaii.edu/users/mendez/ASTRO110LAB11/variables.html> (10.8.2023.)
- 14 Percy J. (2007) *Understanding Variable Stars* 1. izd. Cambridge - 3. i 6. poglavljje
- 15 <https://www.assa.org.au/resources/variable-stars/classifying-variable-stars/> (23.7.2023.)
- 16 https://www.aavso.org/vsots_rrlyr (20.7.2023.)
- 17 https://en.wikipedia.org/wiki/RR_Lyrae_variable (24.7.2023.)
- 18 https://ogle.astrouw.edu.pl/atlas/RR_Lyr.html (24.7.2023.)
- 19 <https://www.aavso.org/sites/default/files/Variable%20Star%20Classification%20and%20Light%20Curves%20Manual%202.1.pdf> (25.7.2023.)
- 20 <https://www.eso.org/sci/publications/messenger/archive/no.13-jun78/messenger-no13-15-17.pdf> (3.8.2023.)
- 21 <http://www.physics.smu.edu/sdalley/quarknet/Variable%20Star%20Astronomy.pdf> (6.8.2023.)
- 22 https://en.wikipedia.org/wiki/Blazhko_effect (6.8.2023.)
- 23 Sazbo R. et al. (2013) *Blazhko effect in Cepheids and RR Lyrae stars*, Cambridge University Press, Vol. 9, Simpozij S301 (<https://arxiv.org/pdf/1309.3969v1.pdf>)
- 24 Jurcsik, Sodor, et al. (2009) *The Konkoly Blazhko Survey: is light-curve modulation a common property of RRab stars?*, Monthly Notices of the Royal Astronomical Society, Vol. 400, Izdanak 2 (<https://ui.adsabs.harvard.edu/abs/2009MNRAS.400.1006J/abstract>)
- 25 Kolenberg K. (2008), 1907-2007: *What's new on the Blazhko front?*, J. Phys.: Conf. Ser. 118 012060 (<https://iopscience.iop.org/article/10.1088/1742-6596/118/1/012060/pdf>)
- 26 Szabo R., Kollath Z. et al. (2010), *Does Kepler unveil the mystery of the Blazhko effect? First detection of period doubling in Kepler Blazhko RR Lyrae stars*, Monthly Notices of the Royal Astronomical Society Vol. 409, Izdanak 3, (<https://arxiv.org/pdf/1007.3404.pdf>)
- 27 Stothers B. (2006), *A new explanation of the Blazhko effect in RR Lyrae stars*, The Astrophysical Journal, 652:643-649, (<https://iopscience.iop.org/article/10.1086/508135/pdf>)
- 28 Netzel, Smolec et al. (2018), *Blazhko effect in the first overtone RR Lyrae stars of the OGLE Galactic bulge collection*, Monthly Notices of the Royal Astronomical Society , Vol. 480, Izdanak 1, (<https://academic.oup.com/mnras/article/480/1/1229/5055626>)

- 29 Teays T. (2018), *The Blazhko effect in RR Lyrae*, IAU colloquium Vol 139., (<https://www.cambridge.org/core/journals/international-astronomical-union-colloquium/article/blazhko-effect-in-rr-lyrae/5A46E7907CCD38501CBB27028CDB0560>)
- 30 Sesar B., Stuart J., Ivezić Ž. et al. (2011), *Exploring the variable sky with LINEAR I. photometric recalibration with the Sloan Digital Sky Survey*, The Astronomical Journal, 142:190, (<http://faculty.washington.edu/ivezic/Publications/Sesar2011LINEAR.pdf>)
- 31 Palaversa L., Ivezić Ž., Eyer L. et al. (2013), *Exploring the variable sky with LINEAR III. Classification of periodic light curves*, The Astronomical Journal, 146:101, (<http://faculty.washington.edu/ivezic/Publications/Palaversa2013LINEAR.pdf>)
- 32 Bellm E., Kulkarni S. et al. (2019), *The Zwicky Transient Facility: System Overview, Performance and First results*, Publications of the Astronomical Society of the Pacific, 131:018002, (<https://iopscience.iop.org/article/10.1088/1538-3873/aaecbe/pdf>)
- 33 Deb and Singh (2009), *Light curve analysis of variable stars using Fourier decomposition and principal component analysis*, Astronomy and Astrophysics, 507, 1729-1737, (<https://www.aanda.org/articles/aa/pdf/2009/45/aa12851-09.pdf>)
- 34 Ivezić Ž. et al. (2004), *The selection of RR Lyrae stars using single-epoch data*, The Astronomical Journal, 129:1096-1108, (<http://faculty.washington.edu/ivezic/Publications/203458.web.pdf>)
- 35 <https://www.youtube.com/watch?v=J5smgeVN38I> (25.7.2023.)
- 36 <https://www.youtube.com/watch?v=NJIIwlVZPws> (3.8.2023.)
- 37 <https://www.youtube.com/watch?v=spUNpyF58BY> (7.8.2023.)
- 38 <http://sait.oat.ts.astro.it/MmSAI/77/PDF/492.pdf> (2.4.2024.)

5.1 Popis slika

Slika 1: ovisnost neprozirnosti plina o temperaturu (aproksimativan model) [8].....	6
Slika 2: prikaz interakcije fotona i atoma H i $H - [6]$	6
Slika 3: model cijevi sa stojnim valovima unutar promjenjive zvijezde (ilustracija autora)	8
Slika 4: prikaz Blazhko efekta kod RR Lira [25, 38].....	11
Slika 5: interferencija valova pri malom periodu promatranja	12
Slika 6: interferencija valova pri duljem periodu promatranja, modulacija amplitude je vidljiva.....	12
Slika 7: modulacija amplitude preko punog perioda promatranja, dolazi do izražaja.....	12
Slika 8: simulacija svjetlosne krivulje za analizu periodograma.....	19
Slika 9: Lomb-Scargle periodogram frekvencija i perioda simulacije svjetlosne krivulje	20
Slika 10: prva faza vizualne analize Blazhko kandidata	24
Slika 11: druga faza vizualne analize Blazhko kandidata.....	25
Slika 12: treća faza vizualne analize Blazhko kandidata	25
Slika 13: četvrta faza vizualne analize Blazhko kandidata, modulacija amplitude.....	26
Slika 14: četvrta faza vizualne analize Blazhko kandidata, fazna modulacija	27
Slika 15: sučelje za vizualnu analizu Blazhko zvijezde 4101289	32
Slika 16: sučelje za vizualnu analizu Blazhko zvijezde 6819457	34
Slika 17: sučelje za vizualnu analizu Blazhko zvijezde 7048826	36
Slika 18: sučelje za vizualnu analizu Blazhko zvijezde 10260828	37

5.2 Popis grafova

Graf 1: distribucija vrijednosti hi-kvadrata	22
Graf 2: distribucija normalizirane razlike perioda.....	23
Graf 3: raspodjela mjerena kod LINEAR RR Lira	28
Graf 4: raspodjela mjerena kod ZTF RR Lira	28
Graf 5: distribucija vremenskih perioda LINEAR zvijezda	29
Graf 6: distribucija vremenskih perioda ZTF zvijezda po filterima	29
Graf 7: prostorna distribucija RR Lira po galaksiji	30
Graf 8: distribucija LINEAR i ZTF Blazhko perioda	38
Graf 9: LINEAR i ZTF razlike glavnih i Blazhko frekvencija.....	39
Graf 10: točkasti dijagram LINEAR i ZTF vrijednosti hi-kvadrata	40
Graf 11: pozicije Blazhko zvijezda pomoću galaktičkih koordinata	41

5.3 Popis tablica

Tablica 1: Metapodaci LINEAR zvijezda koji imaju klasifikaciju tipa zvijezde	15
Tablica 2: 10 Blazhko zvijezda i njihovi osnovni podaci.....	30

6 PRILOG – programi

```
# IMPORTING LIBRARIES
# ----

# AstroML
from astroML.datasets import fetch_LINEAR_sample
from astropy.coordinates import SkyCoord
import astropy.units as u

# Basic libraries
import pickle
import os
import sys
from tqdm import tqdm

# Plotting
import seaborn as sns
from matplotlib import pyplot as plt
from matplotlib import ticker
import matplotlib.colors as mcolors
from matplotlib.font_manager import FontProperties

# DataFrame analysis
import pandas as pd

# Math libraries
import numpy as np
import scipy as sc
from scipy.stats import norm

# CONFIGURATION
# ----
sns.set_theme() # setting the theme for plotting
sys.path.insert(0,'../src/')
np.random.seed(42)

# configuring plotting colors
colors = ['#5F6372', '#79A8A4', '#B2AD8F', '#92A186', '#AD8082']
blue = '#5F6372'
turquoise = '#79A8A4'
light_green = '#B2AD8F'
green = '#92A186'
pink = '#AD8082'
muted_colors = sns.set_palette(sns.color_palette(colors))

# configuring fonts for plotting
font = FontProperties()
font.set_family('avenir')
font.set_name('Big Caslon')
font.set_style('normal')
font.set_size('xx-large')

%matplotlib inline

def select_good_LINEAR(LDATA):
    """
    This function selects correct LINEAR Ids and calculates the preliminary periods of light
    curves.

    Arguments:
        NAME(string): name to save with
        LDATA = LINEAR data
    """
    # -----
    # convert to dataframe for easier manipulation
    IDs = [x for x in LDATA.ids] # making a list of ids

    # ACCESSING THE OFFICIAL PERIODS
```

```

-----
dataPeriods = fetch_LINEAR_geneva() # accessing the good data
dataPeriods = pd.DataFrame(dataPeriods)
ID_orig = list(dataPeriods['LINEARObjectID'].to_numpy()) # accessing the good ID's

LINEAR_data = pd.DataFrame(())

for id in ID_orig: # for every star in the good dataset
    if id in IDs:# if the original ID is in our list of ID's
        index = ID_orig.index(id)
        row = pd.DataFrame(dataPeriods.iloc[[int(index)]]) # assign the current row we are
analyzing
        # concatenate that row with the save_data dataframe
        LINEAR_data = pd.concat([LINEAR_data, row.reset_index(drop=True)],
ignore_index=True, axis=0)

return LINEAR_data

LINEAR_periods = select_good_LINEAR(dataL)
print(LINEAR_periods.shape)
LINEAR_periods.head()
LINEAR_periods.to_csv("../outputs/LINEAR_RRLR.csv", index=False)

Lrrlyr = LINEAR_periods[(LINEAR_periods['gi']>-
0.5)&(LINEAR_periods['gi']<0.4)&(LINEAR_periods['LCtype']>0)&(LINEAR_periods['LCtype']<3)] # RR
lyrae type
Lrrlyr = Lrrlyr.reset_index(drop=True)
Lrrlyr.to_csv('../outputs/Lrrlyr_unprocessed.csv',index=False)
Lab = Lrrlyr[Lrrlyr['LCtype']==1] # RRab type
Lc = Lrrlyr[Lrrlyr['LCtype']==2] # RRc type
print(len(Lab), len(Lc))
Lrrlyr.head()

def getZTFlightcurve(ra, dec, radius=3.0):
    """
    This function uses the right ascension and declination coordinates to find LINEAR
    counterparts in ZTF data.

    Arguments:
        ra(float): right ascension values
        da(float): declination values
        radius(float): radius to search the sky with
    """
    # matching radius is given in arcsec
try:
    lcq = lightcurve.LCQuery()
    res = lcq.from_position(ra, dec, radius)
    ZTFdata = res.data[['mjd', 'mag', 'magerr', 'catflags', 'filtercode']]
    # M. Graham recommends to get rid of obvious spurious points
    ZTFdata = ZTFdata.loc[ZTFdata['catflags'] < 32768]
except:
    ZTFdata = pd.DataFrame(())
return ZTFdata

end = 'FINAL_FINAL_FINAL_complete'
NAME = '../outputs/ZTF_dataset'+end+'.npy'
if os.path.isfile('ZTF_dataset_FINAL_FINAL_COMPLETE.npy'): # if this file exists
    ZTF_data_lc = np.load(NAME, allow_pickle=True)
else:
    Lids = Lrrlyr['LINEARObjectID'].to_numpy()
    rectascension = Lrrlyr['ra'].to_numpy() # list of rectascension coordinates
    declination = Lrrlyr['dec'].to_numpy() # list of declination coordinates
    ZTF_data_lc = []

for i in tqdm(range(len(Lids))): #len(Lids)
    Lid = Lids[i]
    ra = rectascension[i]
    dec = declination[i]
    ZTFdata = getZTFlightcurve(ra, dec)
    ZTF_data_lc.append((Lid, ZTFdata))

```

```

    if i%10==0:
        ZTF_data_lca = np.array(ZTF_data_lc, dtype=object)
        np.save("../outputs/ZTF_dataset_"+end+".npy", ZTF_data_lca, allow_pickle=True)

    ZTF_data_lca = np.array(ZTF_data_lc, dtype=object)
    np.save("../outputs/ZTF_dataset_"+end+".npy", ZTF_data_lca, allow_pickle=True)
ZTF_data_lca= np.load("../outputs/ZTF_dataset_"+end+".npy", allow_pickle=True)

def doPeriods(time, mag, magErr, nterms, nyquist=100, freqFac=1.05, verbose=False):
    """
    This function calculates the best period for RR Lyrae stars using the Lomb-Scargle
    periodogram. It first tries with the auto
    frequency grid, then it zooms in around the highest Lomb-Scargle power peak and searches
    for the best period.

    Notes:
        - the freqFactor = 1.02 allows search for Blazhko periods longer than 50*basic period,
        so ~25 days and longer
        - the freqFactor = 1.05 allows search for Blazhko periods longer than 20*basic period,
        so ~10 days and longer

    Arguments:
        time(array): time data array for light curve
        mag(array): magnitude data array for light curve
        magErr(array): magnitude error data array for light curve
        nterms(int): number of Fourier terms with which to fit for best period
        lsPs(bool): decide if you want to save periodogram or not, default is True so yes
        nyquist(int): highest frequency of search
        freqFac(float): frequency for searching (defining the grid)
    """
    try:
        if verbose: print('Engaging in period calculation...')
        ls = LombScargle(time, mag, magErr, nterms=nterms) # set up a LombScargle object to
        model the frequency and power
        frequencyAuto, powerAuto = ls.autopower(nyquist_factor=nyquist) # calculate the
        frequency and power
        best_freq = frequencyAuto[np.argmax(powerAuto)]
        frequency = np.arange(best_freq/freqFac, best_freq*freqFac, 5e-6)
        power = ls.power(frequency) # compute LS periodogram again
        period = 1. / frequency
        best_period = period[np.argmax(power)] # choosing the period with the highest power
        return best_period, frequency, power
    except:
        if verbose: print('There is no data!')
        # if there is no data, assign everything to 0 or empty
        best_period = 0.0
        frequency = np.array(())
        power = np.array(())
        return best_period, frequency, power

def LINEARLS(dataL, Lid, nterms, verbose=False):
    """
    This function accesses the LINEAR data and calculates the period.

    Arguments:
        LINEARids(list): list of all LINEAR ids
        LINEARlightcurves(array): light curve data
        order(int): the order of light curve in the list
        verbose(bool): printing statements
    """
    if verbose:
        print('-----')
        print('Period and light curve analysis for LINEAR ID =', Lid)
    ### first prepare light curve data
    # LINEAR light curve for this star (specified by provided LINEARid)
    tL, mL, mLerr = dataL[Lid].T

    ### now compute periods (using LombScargle from astropy.timeseries)

```

```

# LINEAR-only period
if verbose:
    print('computing LINEAR period...')
Plinear, fL, pL = doPeriods(tL, mL, mLerr, nterms)

if verbose:
    print('LINEAR period = ', Plinear)

return Plinear, fL, pL, tL, mL, mLerr

def ZTFs(ZTFdata, Lid, nterms, verbose=False):
    """
    This function calculates the period of a ZTF light curve by taking the median of the
    periods of the 3 filters.

    Arguments:
        ZTFdata(array): datafram of ZTF data for a light curve
        nterms(int): number of terms for the Fourier fitting
        ZTFbands(list): list of filters, Default ["zg", "zr", "zi"]
        lsPS(Bool): flag, Default False
        verbose(bool): printing statements
    """

    # variables
    ZTFperiod_ograms = []
    ZTFbands=['zg', 'zr', 'zi']

    if verbose:
        print('And now for the ZTF counterpart -----')
        print(' computing ZTF period...')

    BandData, timeZ, magZ, magErrZ = None, None, None, None
    for b in ZTFbands:
        if verbose: print('Period calculation for ',b, 'filter.')

        BandData = ZTFdata.loc[ZTFdata['filtercode'] == b]
        timeZ = BandData['mjd'].to_numpy()
        magZ = BandData['mag'].to_numpy()
        magErrZ = BandData['magerr'].to_numpy()

        ZTFperiod, Zfreq, Zpow = doPeriods(timeZ, magZ, magErrZ, nterms)
        if ZTFperiod > 0:
            ZTFperiod_ograms.append((ZTFperiod, Zfreq, Zpow, timeZ, magZ, magErrZ))

    ZTFperiod_ograms.sort(key=lambda x: x[0], reverse=True)
    if len(ZTFperiod_ograms) < 3:
        ZTFbestPeriod, ZTFbestfreq, Zbestpow, timeZ, magZ, magErrZ = ZTFperiod_ograms[0]
    else:
        ZTFbestPeriod, ZTFbestfreq, Zbestpow, timeZ, magZ, magErrZ = ZTFperiod_ograms[1]

    if verbose:
        print('ZTF period = ', ZTFbestPeriod)

    return ZTFbestPeriod, ZTFbestfreq, Zbestpow, timeZ, magZ, magErrZ

def LCanalysisFromP(time, mag, magErr, P, ntermsModels):
    """
    This function fits light curve data with a sinusoidal wave using a certain number of terms
    and the period of
    the periodic light curve.

    Arguments:
        time(array): time data array of light curve
        mag(array): magnitude data array of light curve
        magErr(array): magnitude error data array of light curve
        P(float): the best fit period
        ntermsModels(int): the number of terms with which to fit the light curve
    """
    LCanalysisResults = {}
    # first compute best-fit models for given period

```

```

mtf = MultiTermFit(2*np.pi/P, ntermsModels)
mtf.fit(time, mag, magErr)
a, b, c = mtf.predict(1000, return_phased_times=True, adjust_offset=False)
LCanalysisResults['modelPhaseGrid'] = a
LCanalysisResults['modelFit'] = b
LCanalysisResults['dataPhasedTime'] = c
# light curve template normalization: mag = A * t(phi) + mmax, where
# phi is phase, t is template, A is amplitude and mmax is the magnitude at
# maximum light (note: numerically it is the minimum value of mag)
# also: we are using models for computing amplitude and mmax to avoid noise in data
A = np.max(b) - np.min(b)
mmax = np.min(b)
LCanalysisResults['A'] = A
LCanalysisResults['mmax'] = mmax
LCanalysisResults['modTemplate'] = (b - mmax)/A
LCanalysisResults['dataTemplate'] = (mag - mmax)/A
LCanalysisResults['dataTemplateErr'] = magErr/A
# for chi2, first interpolate model fit to phases of data values
modelFit2data = np.interp(c, a, LCanalysisResults['modTemplate'])
LCanalysisResults['modelFit2data'] = modelFit2data
delmag = LCanalysisResults['dataTemplate'] - modelFit2data
LCanalysisResults['chi'] = delmag/LCanalysisResults['dataTemplateErr']
LCanalysisResults['chi2dof'] =
np.sum(LCanalysisResults['chi']**2)/np.size(LCanalysisResults['chi'])
LCanalysisResults['chi2dofR'] = sigG(LCanalysisResults['chi'])
return LCanalysisResults

Lrrlyr = pd.read_csv('../outputs/Lrrlyr_unprocessed.csv')
end = 'ffinal'
dataZTF = np.load("../outputs/ZTF_dataset_"+end+".npy", allow_pickle=True)

Lids = Lrrlyr['LINEARobjectID'].to_numpy()
rectascension = Lrrlyr['ra'].to_numpy() # list of rectascension coordinates
declination = Lrrlyr['dec'].to_numpy() # list of declination coordinates

test = len(Lids)
ZTF_data_best = []
periodograms = []
lc_analysis = {}
fits = []

for i in tqdm(range(test)):
    # designated LINEAR id
    Lid = Lids[i]
    # SEARCHING FOR ZTF DATA
    for j in dataZTF:
        # If the ID's match
        if j[0]==Lid:
            # check if ZTF has data for this ID
            if j[1].shape[0] == 0:
                # if not, save None and break from for loop
                dataZ = None
                break
            else:
                # if there is data, assign it and also break
                dataZ = j[1]
                break
    # if there is no data, continue with main loop (skip this iteration)
    if dataZ is None:
        continue

    #print(dataZ.shape)

    # PERIOD ANALYSIS
    Plinear, fL, pL, tL, mL, mEL = LINEARLS(dataL, Lid, 3)
    Pztf, fz, pz, tz, mz, meZ = ZTFs(dataZ, Lid, 3)

    if Plinear==0.0 or Pztf==0.0:
        continue

    # Period calculations

```

```

Plinear = round(Plinear, 6)
Pztf = round(Pztf, 6)
Pmean = round((Plinear+Pztf)/2, 6)
Pratio = round((Pztf/Plinear), 6)

# BLAZHKO ANALYSIS
fFoldedL, pFoldedL, fMainPeakL, fBlazhkoPeakL, BlazhkoPeriodL, BpowerRatioL,
BsignificanceL = getBlazhkoPeak(fL, pL)
fFoldedZ, pFoldedZ, fMainPeakZ, fBlazhkoPeakZ, BlazhkoPeriodZ, BpowerRatioZ,
BsignificanceZ = getBlazhkoPeak(fZ, pZ)

# Saving periodogram data
periodograms.append((Lid, (fL, pL, fFoldedL, pFoldedL), (fZ, pZ, fFoldedZ, pFoldedZ)))

# CHI SQUARE MODELING
ntermsModels = 6
LINEAR_Plinear = LCanalysisFromP(tL, mL, meL, Plinear, ntermsModels)
LINEAR_Pmean = LCanalysisFromP(tL, mL, meL, Pmean, ntermsModels)

ZTF_Pztf = LCanalysisFromP(tZ, mZ, meZ, Pztf, ntermsModels)
ZTF_Pmean = LCanalysisFromP(tZ, mZ, meZ, Pmean, ntermsModels)

# SAVING DATA
STAR = [Plinear, Pztf, Pmean, Pratio, np.size(tL), round(LINEAR_Plinear['chi2dof'], 1),
round(LINEAR_Plinear['chi2dofR'], 1),
round(LINEAR_Pmean['chi2dof'], 1), round(LINEAR_Pmean['chi2dofR'], 1),
round(LINEAR_Plinear['mmax'], 2), round(LINEAR_Plinear['A'], 2),
np.size(tZ), round(ZTF_Pztf['chi2dof'], 1), round(ZTF_Pztf['chi2dofR'], 1),
round(ZTF_Pmean['chi2dof'], 1), round(ZTF_Pmean['chi2dofR'], 1), round(ZTF_Pztf['mmax'], 2),
round(ZTF_Pztf['A'], 2),
fMainPeakL, fBlazhkoPeakL, BlazhkoPeriodL, BpowerRatioL, BsignificanceL,
fMainPeakZ,
fBlazhkoPeakZ, BlazhkoPeriodZ, BpowerRatioZ, BsignificanceZ]

lc_analysis[Lid] = STAR
fits.append((Lid, (LINEAR_Plinear, LINEAR_Pmean, ZTF_Pztf, ZTF_Pmean)))
ZTF_data_best.append((Lid, tZ, mZ, meZ))

# saving the data on the computer
if i%10==0:
    # save dictionary to pkl file
    with open('../outputs/lc_analysis_'+end+'.pkl', 'wb') as fp:
        pickle.dump(lc_analysis, fp)
    with open("../outputs/periodograms_"+end+".pkl", "wb") as x:
        pickle.dump(periodograms, x)
    FITS = np.array(fits, dtype=object)
    np.save("../outputs/fits_"+end+".npy", FITS, allow_pickle=True)
    ZTF_data_lca = np.array(ZTF_data_best, dtype=object)
    np.save("../outputs/ztf_data_filter"+end+".npy", ZTF_data_lca, allow_pickle=True)

# FINAL SAVE
# -----
# save dictionary to pkl file
with open('../outputs/lc_analysis_'+end+'.pkl', 'wb') as fp:
    pickle.dump(lc_analysis, fp)
with open("../outputs/periodograms_"+end+".pkl", "wb") as x:
    pickle.dump(periodograms, x)
FITS = np.array(fits, dtype=object)
np.save("../outputs/fits_"+end+".npy", FITS, allow_pickle=True)
ZTF_data_lca = np.array(ZTF_data_best, dtype=object)
np.save("../outputs/ztf_data_filter"+end+".npy", ZTF_data_lca, allow_pickle=True)

cols = ['Plinear', 'Pztf', 'Pmean', 'Pratio', 'NdataLINEAR',
'L_rms', 'L_chi2dof', 'L_chi2dofR',
'Lmean_rms', 'Lmean_chi2dof', 'Lmean_chi2dofR', 'Lmmax', 'Lampl',
'NdataZTF', 'Z_rms', 'Zchi2dof',
'Zchi2dofR', 'Zmean_rms', 'Zmean_chi2dof', 'Zmean_chi2dofR',
'Zmmax', 'Zampl',
'MainPeakL', 'BlazhkoPeakL', 'BlazhkoPeriodL', 'BpowerRatioL', 'BsignificanceL',
'MainPeakZ', 'BlazhkoPeakZ', 'BlazhkoPeriodZ', 'BpowerRatioZ', 'BsignificanceZ']

```

```

LC_ANALYSIS = pd.DataFrame.from_dict(lc_analysis, orient='index', columns=cols)
LC_ANALYSIS.index.name='LINEAR id'
LC_ANALYSIS = LC_ANALYSIS.reset_index()
LC_ANALYSIS.to_csv("../outputs/LC_ANALYSIS_"+end+".csv", index=False)
LC_ANALYSIS = pd.read_csv("../outputs/LC_ANALYSIS_"+end+".csv")

LC_ANALYSIS['MainPeakL'] = pd.to_numeric(LC_ANALYSIS['MainPeakL'], errors='coerce').fillna(0)
LC_ANALYSIS['BlazhkoPeakL'] = pd.to_numeric(LC_ANALYSIS['BlazhkoPeakL'],
errors='coerce').fillna(0)
LC_ANALYSIS['BlazhkoPeriodL'] = pd.to_numeric(LC_ANALYSIS['BlazhkoPeriodL'],
errors='coerce').fillna(0)
LC_ANALYSIS['BpowerRatioL'] = pd.to_numeric(LC_ANALYSIS['BpowerRatioL'],
errors='coerce').fillna(0)
LC_ANALYSIS['BsignificanceL'] = pd.to_numeric(LC_ANALYSIS['BsignificanceL'],
errors='coerce').fillna(0)
LC_ANALYSIS['MainPeakZ'] = pd.to_numeric(LC_ANALYSIS['MainPeakZ'], errors='coerce').fillna(0)
LC_ANALYSIS['BlazhkoPeakZ'] = pd.to_numeric(LC_ANALYSIS['BlazhkoPeakZ'],
errors='coerce').fillna(0)
LC_ANALYSIS['BlazhkoPeriodZ'] = pd.to_numeric(LC_ANALYSIS['BlazhkoPeriodZ'],
errors='coerce').fillna(0)
LC_ANALYSIS['BpowerRatioZ'] = pd.to_numeric(LC_ANALYSIS['BpowerRatioZ'],
errors='coerce').fillna(0)
LC_ANALYSIS['BsignificanceZ'] = pd.to_numeric(LC_ANALYSIS['BsignificanceZ'],
errors='coerce').fillna(0)

LC_ANALYSIS['Pratio'] = round(LC_ANALYSIS['Pratio'], 6)
LC_ANALYSIS['Plinear'] = round(LC_ANALYSIS['Plinear'], 6)
LC_ANALYSIS['Pztf'] = round(LC_ANALYSIS['Pztf'], 6)
LC_ANALYSIS['Pmean'] = round(LC_ANALYSIS['Pmean'], 6)
LC_ANALYSIS['LampL'] = round(LC_ANALYSIS['LampL'], 2)
LC_ANALYSIS['ZampL'] = round(LC_ANALYSIS['ZampL'], 2)
LC_ANALYSIS['L_chi2dofR'] = round(LC_ANALYSIS['L_chi2dofR'], 1)
LC_ANALYSIS['Lmean_chi2dofR'] = round(LC_ANALYSIS['Lmean_chi2dofR'], 1)
LC_ANALYSIS['Zchi2dofR'] = round(LC_ANALYSIS['Zchi2dofR'], 1)
LC_ANALYSIS['Zmean_chi2dofR'] = round(LC_ANALYSIS['Zmean_chi2dofR'], 1)
LC_ANALYSIS['L_chi2dof'] = round(LC_ANALYSIS['L_chi2dof'], 1)
LC_ANALYSIS['Lmean_chi2dof'] = round(LC_ANALYSIS['Lmean_chi2dof'], 1)
LC_ANALYSIS['Zchi2dof'] = round(LC_ANALYSIS['Zchi2dof'], 1)
LC_ANALYSIS['Zmean_chi2dof'] = round(LC_ANALYSIS['Zmean_chi2dof'], 1)
LC_ANALYSIS['Ampl_diff'] = round(abs(LC_ANALYSIS['LampL'] - LC_ANALYSIS['ZampL']), 2)
LC_ANALYSIS['dP'] = round(abs((LC_ANALYSIS['Plinear']-
LC_ANALYSIS['Pztf'])/LC_ANALYSIS['Pmean']),5)
LC_ANALYSIS['MainPeakL'] = round(LC_ANALYSIS['MainPeakL'], 4)
LC_ANALYSIS['BlazhkoPeakL'] = round(LC_ANALYSIS['BlazhkoPeakL'], 4)
LC_ANALYSIS['BlazhkoPeriodL'] = round(LC_ANALYSIS['BlazhkoPeriodL'], 4)
LC_ANALYSIS['BpowerRatioL'] = round(LC_ANALYSIS['BpowerRatioL'], 4)
LC_ANALYSIS['BsignificanceL'] = round(LC_ANALYSIS['BsignificanceL'], 4)
LC_ANALYSIS['MainPeakZ'] = round(LC_ANALYSIS['MainPeakZ'], 4)
LC_ANALYSIS['BlazhkoPeakZ'] = round(LC_ANALYSIS['BlazhkoPeakZ'], 4)
LC_ANALYSIS['BlazhkoPeriodZ'] = round(LC_ANALYSIS['BlazhkoPeriodZ'], 4)
LC_ANALYSIS['BpowerRatioZ'] = round(LC_ANALYSIS['BpowerRatioZ'], 4)
LC_ANALYSIS['BsignificanceZ'] = round(LC_ANALYSIS['BsignificanceZ'], 4)
LC_ANALYSIS['IndicatorType'] = 'NaN'
LC_ANALYSIS['ChiType'] = 'NaN'
LC_ANALYSIS['BE_score'] = 'NaN'

LC_ANALYSIS.to_csv("../outputs/LC_ANALYSIS_"+end+".csv", index=False)
periodogr = pickle.load(open("../outputs/periodograms_"+end+".pkl", "rb"))
fits = np.load("../outputs/fits_"+end+".npy", allow_pickle=True)
ztf_data = np.load("../outputs/ztf_data_filter"+end+".npy", allow_pickle=True)

```

```

def getBlazhkoPeak(freq, LSpow, verbose=False):
    """
    This function searches for the Blazhko effect in periodograms of light curves. It searches
    for 2 subsequent peaks by
    folding the light curve and searching for local peaks. It also accounts for year aliases.
    
```

Arguments:

```

freq(array): frequency array
LSpow(array): lomb-scargle power array

```

```

    verbose(bool): print statements
    ...
    # no. of points
    Npts = np.size(LSpow)
    # index for the main peak
    imax = np.argmax(LSpow)
    # 1 year alias frequency (factor 1.02 to push it a bit over the maximum)
    f1yr = freq[imax] + 1.02/365
    # iDelta is the max. width for folding around the main peak
    if (imax < Npts/2):
        iDelta = imax
    else:
        iDelta = Npts - imax
    # folded versions
    fFolded = freq[imax:imax+1+iDelta]
    pLeft = LSpow[imax-iDelta:imax+1]
    pRight = LSpow[imax:imax+1+iDelta]
    pFolded = 0*fFolded
    for i in range(0, iDelta-1):
        # multiply the two branches to increase SNR
        pFolded[i] = pLeft[-i-1] * pRight[i]
    # now search for the strongest secondary minimum (after the main one at index=0)
    foundMin = 0
    foldedMax = 0
    ifoldedMax = 0
    # NB: the first point is the highest by construction (the main peak)
    for i in range(1, iDelta):
        if ((foundMin==0)&(pFolded[i] > pFolded[i-1])):
            # the first time we passed through a local minimum
            if (fFolded[i]>f1yr): foundMin = 1
        if foundMin:
            # after the first local minimum, remember the maximum power and its location
            if (pFolded[i] > foldedMax):
                foldedMax = pFolded[i]
                ifoldedMax = i
    # done, return useful quantities
    fMainPeak = freq[imax] # location of the main peak
    fBlazhkoPeak = fFolded[ifoldedMax] # location of the second strongest peak
    if (fBlazhkoPeak - fMainPeak)==0:
        BlazhkoPeriod = 0
    else:
        BlazhkoPeriod = 1/(fBlazhkoPeak - fMainPeak) # expression for Blazhko period
        if BlazhkoPeriod == np.inf:
            BlazhkoPeriod = 0
    BpowerRatio = pFolded[ifoldedMax]/fFolded[0] # the ratio of power for the 2nd and 1st peaks
    if BpowerRatio==np.inf:
        BpowerRatio = 0
    # now compare the second peak's strength to the power at larger frequencies (presumably
    noise)
    powerFar = pFolded[fFolded>fBlazhkoPeak] # frequencies beyond the second peak
    powerFarMedian = np.median(powerFar) # the median power
    powerFarRMS = np.std(powerFar) # standard deviation, i.e. "sigma"
    if powerFarRMS==0:
        Bsignificance = 0
    else:
        Bsignificance = (pFolded[ifoldedMax]-powerFarMedian)/powerFarRMS # how many sigma
        above median?
        if Bsignificance==np.inf:
            Bsignificance = 0
    if (verbose):
        print('main frequency (1/day):', fMainPeak)
        print('detected second peak at index:', ifoldedMax)
        print('Blazhko peak frequency (1/day):', fBlazhkoPeak)
        print('Blazhko peak relative strength:', BpowerRatio)
        print('median power beyond Blazhko peak:', powerFarMedian)
        print('power rms beyond Blazhko peak:', powerFarRMS)
        print('Blazhko peak significance:', Bsignificance)
        print('Blazhko period (day):', BlazhkoPeriod)
    return fFolded, pFolded, fMainPeak, fBlazhkoPeak, BlazhkoPeriod, BpowerRatio, Bsignificance

```

```

# BLAZHKO EFFECT CANDIDATES
# =====
def blazhko_determine(df, dfnew):
    """
        This algorithm sorts through a DataFrame of light curve parameters and decides which are
        bad Blazhko Effect candidates,
        which are inter BE candidates, good BE candidates and excellent BE candidates. The
        parameters we use for determining
        BE candidates are amplitude, chi2 of 2 both LINEAR and ZTF, period and the periodogram
        analysis (sign of local peaks for BE).

    Arguments:
        df(DataFrame) = input dataframe
        dfnew(DataFrame) = new dataframe for inputting good candidates
    """
    for i in range(df.shape[0]):

        # STEP 1: getting rid of trash
        # -----
        if df['Ampl_diff'][i]<2:
            if df['L_chi2dofR'][i]<9 or df['Zchi2dofR'][i]<9 or df['Plinear'][i]<4 or
df['Pztf'][i]<4:
                if df['NdataLINEAR'][i]>250 and df['NdataZTF'][i]>40:
                    if df['Pratio'][i]>0.8 and df['Pratio'][i]<1.2:
                        # STEP 2: determine periodogram likelihood of BE
                        # -----
                        dPmin = 0.01
                        #--- determining if LINEAR part has periodogram indication of BE ---
                        # no daily alias of main period
                        LINEAR_pd_period = (np.abs(df['Plinear'][i]-1.0)>dPmin) & (np.abs(df['Plinear'][i]-2.0)>dPmin)
                        # blazhko period must be within RR Lyrae range
                        LINEAR_pd_pB =
(df['BlazhkoPeriodL'][i]>35)&(df['BlazhkoPeriodL'][i]<325)
                            # relative strength and significance must be above certain value for it
                            to be noticeable
                            LINEAR_pd_sig =
(df['BpowerRatioL'][i]>0.05)&(df['BsignificanceL'][i]>5)
                                #--- determining if ZTF part has periodogram indication of BE ---
                                ZTF_pd_period = (np.abs(df['Pztf'][i]-
0.5)>dPmin) & (np.abs(df['Pztf'][i]-1.0)>dPmin) & (np.abs(df['Pztf'][i]-2.0)>dPmin)
                                ZTF_pd_pB = (df['BlazhkoPeriodZ'][i]>35)&(df['BlazhkoPeriodZ'][i]<325)
                                ZTF_pd_sig = (df['BpowerRatioZ'][i]>0.05)&(df['BsignificanceZ'][i]>5)
                                #---
                                BE = 0
                                if
((LINEAR_pd_period&LINEAR_pd_pB&LINEAR_pd_sig) & (ZTF_pd_period&ZTF_pd_pB&ZTF_pd_sig)):
                                    BE += 1
                                    df.loc[i, 'IndicatorType'] = 'LZ'
                                if (LINEAR_pd_period&LINEAR_pd_pB&LINEAR_pd_sig):
                                    BE += 1
                                    df.loc[i, 'IndicatorType'] = 'L'
                                if (ZTF_pd_period&ZTF_pd_pB&ZTF_pd_sig):
                                    BE += 1
                                    df.loc[i, 'IndicatorType'] = 'Z'
                                # ---
                                if BE>0:
                                    row = pd.DataFrame(df.iloc[[int(i)]])
                                    dfnew = pd.concat([dfnew, row.reset_index(drop=True)],
ignore_index=True, axis=0)
                                else:
                                    period = df['dP'][i]
                                    chiL = df['L_chi2dofR'][i]
                                    chiZ = df['Zchi2dofR'][i]
                                    ampl = df['Ampl_diff'][i]

                                    # ---

                                    p_score = 0
                                    chi_score = 0
                                    amp_score = 0

```

```

# ---
# PERIOD
if period > 0.00002 and period < 0.00005: p_score += 2
if period >= 0.00005: p_score += 4

# CHI
if (chiZ>=2.0 and chiZ<=4.0)and(chiL >= 1.8 and chiL <= 3.0):
    chi_score += 4
    df.loc[i, 'ChiType'] = 'LZ'
if (chiL>3.0)and(chiZ>4.0):
    chi_score += 6
    df.loc[i, 'ChiType'] = 'LZ'
if (chiL >=1.8 and chiL <= 3.0):
    chi_score += 2
    df.loc[i, 'ChiType'] = 'L'
if (chiZ>=2.0 and chiZ<=4.0):
    chi_score += 2
    df.loc[i, 'ChiType'] = 'Z'
if chiL>3.0:
    chi_score += 3
    df.loc[i, 'ChiType'] = 'L'
if chiZ>4.0:
    chi_score += 3
    df.loc[i, 'ChiType'] = 'Z'

# AMPL
if ampl>0.05 and ampl<0.15: amp_score += 1
if ampl>0.15 and ampl<2: amp_score += 2

if amp_score > p_score:
    df.loc[i, 'period_vs_amp'] = 'amp'
else:
    df.loc[i, 'period_vs_amp'] = 'period'

# TOTAL SCORE
score = p_score + chi_score + amp_score
df.loc[i, 'BE_score'] = score

if (score>4):
    row = pd.DataFrame(df.iloc[[int(i)]])
    dfnew = pd.concat([dfnew, row.reset_index(drop=True)], ignore_index=True, axis=0)
else:
    pass
return dfnew

# BUILDING THE VISUAL INTERFACE
# =====
# Building a class for the visual interface
class BE_analyzer:
    def __init__(self, linear_ids, tot, database_lightc, be_cand, lightc_fits, lightc_per, Zdata, Ldata, plotSave=False):
        self.linear_ids = linear_ids
        self.database_lightc = database_lightc
        self.be_cand = be_cand
        self.lightc_fits = lightc_fits
        self.lightc_per = lightc_per
        self.Zdata = Zdata
        self.Ldata = Ldata
        self.total_num = tot
        self.plotSave = plotSave

        self.current_i = None
        self.generate = self.plot_BE_data()

        self.keep_button = widgets.Button(description='KEEP')
        self.con_button = widgets.Button(description='CONTINUE')
        self.keep_button.on_click(self.click_keep)
        self.con_button.on_click(self.click_con)

```

```

        self.output = widgets.Output()
        #display(self.output, self.keep_button, self.con_button)

    def plot_BE_data(self):
        #print('Engaging in plotting!')
        for i in range(len(self.linear_ids)):
            self.current_i = i
            #print('My current i:', self.current_i)
            LID = self.linear_ids[self.current_i]
            for n, j in enumerate(self.lightc_fits):
                if j[0]==LID:
                    break
            L1 = self.lightc_fits[n][1][0]
            L2 = self.lightc_fits[n][1][2]

            for o, k in enumerate(self.lightc_per):
                if k[0]==LID:
                    break

            fL = self.lightc_per[o][1][0]
            pL = self.lightc_per[o][1][1]
            fZ = self.lightc_per[o][2][0]
            pZ = self.lightc_per[o][2][1]

            fFoldedL = self.lightc_per[o][1][2]
            pFoldedL = self.lightc_per[o][1][3]
            fFoldedZ = self.lightc_per[o][2][2]
            pFoldedZ = self.lightc_per[o][2][3]

            lc = self.Ldata.get_light_curve(LID)
            tL = lc.T[0]
            for f, g in enumerate(self.Zdata):
                if g[0] == LID:
                    break
            tZ = self.Zdata[f][1]

            #print('Starting to plot!')
            #makeLCplot_info(L1, L2, self.database_lightc, i, LID, self.Ldata)
            if self.plotSave:
                BE_plotting.plotAll(LID, n, i, self.total_num, L1, L2, self.database_lightc,
fL, pL, fZ, pZ, fFoldedL, fFoldedZ, pFoldedL, pFoldedZ, self.Ldata, tL, tZ, self.Zdata,
plotSave=True)
            else:
                BE_plotting.plotAll(LID, n, i, self.total_num, L1, L2, self.database_lightc,
fL, pL, fZ, pZ, fFoldedL, fFoldedZ, pFoldedL, pFoldedZ, self.Ldata, tL, tZ, self.Zdata)

            yield

    def click_keep(self, b):
        row = pd.DataFrame(self.database_lightc.iloc[[int(self.current_i)]]) # assign the
current row we are analyzing
        # concatenate that row with the save_data dataframe
        self.be_cand = pd.concat([self.be_cand, row.reset_index(drop=True)], ignore_index=True,
axis=0)

        with self.output:
            clear_output(wait=True) # clear the previous output
            #print('Clearing output!')
            try:
                #print('Next image generated!')
                next(self.generate) # generate the next plot and update current_i
            except StopIteration: # when the for loop is finished, disable the button
                print("No more plots.")
                self.con_button.disabled = True

    def click_con(self, b):
        #print('Button clicked!')

```

```

    with self.output:
        clear_output(wait=True) # clear the previous output
        #print('Clearing output!')
    try:
        #print('Next image generated!')
        next(self.generate) # generate the next plot and update current i
    except StopIteration: # when the for loop is finished, disable the button
        print("No more plots.")
        self.con_button.disabled = True

    def get_save_data(self):
        return self.be_cand

    def display_interface(self):
        # Create a layout for the widgets
        self.layout = widgets.VBox([self.output, self.keep_button, self.con_button])
        # Display the layout
        display(self.layout)

    def category_analysis(begin_data, fits, periodogr, ztf_data, dataLINEAR,end,
id_list=None,parameter=None, value=None, plotSave=False):
    """
    This function takes in a certain parameter and then generates a seperate dataset and
    interface
    in order to analyze it for Blazhko stars.

    Arguments:
        begin_data(DataFrame) = starting dataframe from which we take data
        parameter(string) = the parameter of interest
        value(string) = the value equal to which we select candidates
        fits = the fits dataset
        periodogr = the periodogram dataset
        ztf_data = the ZTF dataset
        dataLINEAR = the LINEAR dataset
        end(string) = with which we save the data
    """

    if parameter:
        new_dataset = begin_data.loc[(begin_data[parameter] == value)]
        new_dataset = new_dataset.reset_index(drop=True)
        print(f'This dataset has {new_dataset.shape[0]} stars.')

        # ----

        length = new_dataset.shape[0]
        Lids = new_dataset['LINEAR id'].to_numpy()

        blazhko_analyzer = pd.DataFrame()
        if plotSave:
            analysis = BE_analyzer(Lids, length, new_dataset, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR, plotSave=True)
            analysis.display_interface()
        else:
            analysis = BE_analyzer(Lids, length, new_dataset, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR)
            analysis.display_interface()
    else:
        if id_list:
            new_dataset = begin_data[begin_data['LINEAR id'].isin(id_list)]
            new_dataset = new_dataset.reset_index(drop=True)
            print(f'This dataset has {new_dataset.shape[0]} stars.')

            # ----

            length = new_dataset.shape[0]
            Lids = new_dataset['LINEAR id'].to_numpy()

            blazhko_analyzer = pd.DataFrame()
            if plotSave:

```

```

        analysis = BE_analyzer(Lids, length, new_dataset, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR, plotSave=True)
            analysis.display_interface()
    else:
        analysis = BE_analyzer(Lids, length, new_dataset, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR)
            analysis.display_interface()
    else:
        print(f'This dataset has {begin_data.shape[0]} stars.')
        length = begin_data.shape[0]
        Lids = begin_data['LINEAR id'].to_numpy()

        blazhko_analyzer = pd.DataFrame({})
        if plotSave:
            analysis = BE_analyzer(Lids, length, begin_data, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR, plotSave=True)
            analysis.display_interface()
        else:
            analysis = BE_analyzer(Lids, length, begin_data, blazhko_analyzer, fits,
periodogr, ztf_data, dataLINEAR)
            analysis.display_interface()
    return analysis

def makeLCplot_info(L1, L2, dataset, order, Lid, dataL, total_num, plotname='LCplot',
plotSave=False):
    """
    This function plots a single phase of a light curve with fit for both LINEAR and ZTF data,
    along with
    a separate box for text data.

    Arguments:
        L1: fit data for light curve
    """
    fig, ax = plt.subplots(1,3, figsize=(32,8))
    fig.suptitle('STAR '+str(order+1)+ ' from '+str(total_num), fontsize=30,
fontproperties=font)
    fig.set_facecolor('white')

    ax[0].set_xlabel('data phased with best-fit LINEAR period', fontproperties=font,
fontsize=14)
    ax[0].set_ylabel('LINEAR normalized light curve', fontproperties=font, fontsize=14)
    ax[0].set_xlim(-0.1, 1.1)
    ax[0].set_ylim(1.3, -0.3)
    # data
    xx, yy, zz = sort3arr(L1['dataPhasedTime'], L1['dataTemplate'], L1['dataTemplateErr'])
    ax[0].errorbar(xx, yy, zz, fmt='.k', ecolor='gray', lw=1, ms=4, capsiz=1.5, alpha=0.3)
    # fit for Plinear
    ax[0].plot(L1['modelPhaseGrid'], L1['modTemplate'], pink, markeredgecolor=pink, lw=1,
fillstyle='top', linestyle='dashed')

    ax[1].set_xlabel('data phased with best-fit ZTF period', fontproperties=font, fontsize=14)
    ax[1].set_ylabel('ZTF normalized light curve', fontproperties=font, fontsize=14)
    ax[1].set_xlim(-0.1, 1.1)
    ax[1].set_ylim(1.3, -0.3)
    # data
    xx1, yy1, zz1 = sort3arr(L2['dataPhasedTime'], L2['dataTemplate'], L2['dataTemplateErr'])
    ax[1].errorbar(xx1, yy1, zz1, fmt='.k', ecolor='gray', lw=1, ms=4, capsiz=1.5, alpha=0.3)
    # fit for Plinear
    ax[1].plot(L2['modelPhaseGrid'], L2['modTemplate'], 'red', markeredgecolor='red', lw=1,
fillstyle='top', linestyle='dashed')

    ax[2].axis([0, 8, 0, 10])
    ax[2].text(0, 8, 'LINEAR period chi robust: '+str(dataset['L_chi2dofR'][order])+' , LINEAR
mean period chi robust:
'+str(dataset['Lmean_chi2dofR'][order]), fontsize=15, fontproperties=font)
    ax[2].text(0, 7, 'ZTF period chi robust: '+str(dataset['Zchi2dofR'][order])+' , ZTF mean
period chi robust: '+str(dataset['Zmean_chi2dofR'][order]), fontsize=15, fontproperties=font)
    ax[2].text(0, 6, 'LINEAR period chi: '+str(dataset['L_chi2dof'][order])+' , LINEAR mean
period chi: '+str(dataset['Lmean_chi2dof'][order]), fontsize=15, fontproperties=font)

```

```

    ax[2].text(0, 5, 'ZTF period chi: '+str(dataset['Zchi2dof'][order])+' , ZTF mean period chi: '+str(dataset['Zmean_chi2dof'][order]), fontsize=15, fontproperties=font)
    ax[2].text(0, 4, 'LINEAR period: '+str(dataset['Plinear'][order])+' , ZTF period: '+str(dataset['Pztf'][order])+' , Period difference: '+str(dataset['dP'][order]), fontsize=15, fontproperties=font)
    ax[2].text(0, 3, 'Average LINEAR magnitude: '+str(round(np.mean(dataL.get_light_curve(Lid).T[1]), 2)), fontsize=15, fontproperties=font)
    ax[2].text(0, 2, 'LINEAR amplitude: '+str(dataset['Lampl'][order])+' , ZTF amplitude: '+str(dataset['Zamp'][order]), fontsize=15, fontproperties=font)
    if dataset['period_vs_amp'][order] != np.nan:
        ax[2].text(0, 1, '- this star has a stronger '+str(dataset['period_vs_amp'][order])+' score', fontsize=15, fontproperties=font)

    ax[2].grid(False)
    ax[2].axis('off')

    if plotSave:
        plotName = plotname + '.png'
        plt.savefig('../images/'+plotName, dpi=750, bbox_inches = 'tight')
    plt.show()
    #print('Finished plotting!')

    return

def plotBlazhkoPeaksLINEAR(Lid, order, fL, pL, fZ, pZ, fFoldedL, pFoldedL, fFoldedZ, pFoldedZ,
dataset, fac=1.008, plotSave=False, verbose=False):
    flin = fL[np.argmax(pL)]
    fztf = fZ[np.argmax(pZ)]

    # DATA PREP
    # =====
    fBlazhkoPeakL = dataset['BlazhkoPeakL'][order]
    # ---
    fBlazhkoPeakZ = dataset['BlazhkoPeakZ'][order]

    if fL.size==0 or pL.size==0 or fZ.size==0 or pZ.size==0 or fFoldedL.size==0 or pFoldedL.size==0 or fFoldedZ.size==0 or pFoldedZ.size==0:
        print("No available periodogram data.")
    else:
        ## at some point, we will read periodograms back from files...
        fig = plt.figure(figsize=(32, 8))
        fig.subplots_adjust(hspace=0.1, bottom=0.06, top=0.94, left=0.12, right=0.94)

        # plot the power spectrum
        ax = fig.add_subplot(141)

        ax.plot(fL, pL, c='b')
        ax.plot([flin, flin], [0,1], lw = 1, c=pink, ls='--')
        ax.plot([fBlazhkoPeakL, fBlazhkoPeakL], [0, 0.7*np.max(pFoldedL)], lw = 1, c=pink, ls='--')
        ax.plot([2*flin-fBlazhkoPeakL, 2*flin-fBlazhkoPeakL], [0, 0.7*np.max(pFoldedL)], lw = 1, c=pink, ls='--')
        # show 1 year alias
        f1yr = flin+1/365.0
        ax.plot([f1yr, f1yr], [0,0.7*np.max(pFoldedL)], lw = 1, ls='-.', c=green)
        f1yr = flin-1/365.0
        ax.plot([f1yr, f1yr], [0,0.7*np.max(pFoldedL)], lw = 1, ls='-.', c=green)

        ax.text(0.03, 0.96, "LINEAR", ha='left', va='top',
transform=ax.transAxes, fontproperties=font, fontsize=10)
        if (fBlazhkoPeakL > flin*fac):
            ax.set_xlim(0.99*(2*flin-fBlazhkoPeakL), 1.01*fBlazhkoPeakL)
        else:
            ax.set_xlim(flin/fac, flin*fac)
        ax.yaxis.set_major_locator(plt.MaxNLocator(4))

        ylim = ax.get_ylimits()
        ymax = ylim[0] + 1.1 * (ylim[1] - ylim[0])
        if ymax>1.0: ymax=1.0
        ax.set_ylimits(0, ymax)
        ax.set_ylabel('Lomb-Scargle power', fontproperties=font, fontsize=14)

```

```

    ax.set_xlabel('frequency (d$^{-1}$)', fontproperties=font, fontsize=14)

    # plot folder power spectrum
    ax = fig.add_subplot(142)

    ax.plot(fFoldedL, pFoldedL, c=blue)
    ax.plot([fBlazhkoPeakL, fBlazhkoPeakL], [0, 0.4*np.max(pFoldedL)], lw = 1, ls='--',
c=pink)
    # show 1 year alias
    flyr = flin+1/365.0
    ax.plot([flyr, flyr], [0, 0.4*np.max(pFoldedL)], lw = 1, ls='-.', c=green)

    powerFar = pFoldedL[fFoldedL>fBlazhkoPeakL] # frequencies beyond the second peak
    powerFarMedian = np.median(powerFar)          # the median power
    powerFarRMS = np.std(powerFar)                # standard deviation, i.e. "sigma"
    noise5sig = powerFarMedian+5*powerFarRMS

    if (fBlazhkoPeakL > flin*fac):
        ax.plot([flin+0.5*(fBlazhkoPeakL-flin), 1.01*fBlazhkoPeakL], [noise5sig,
noise5sig], lw = 1, ls='--', c=turquoise)
        ax.set_xlim(flin, 1.01*fBlazhkoPeakL)
    else:
        ax.plot([flin+0.5*(fBlazhkoPeakL-flin), flin*fac], [noise5sig, noise5sig], lw = 1,
ls='--', c=turquoise)
        ax.set_xlim(flin, flin*fac)

    ax.yaxis.set_major_locator(plt.MaxNLocator(4))

    ylim = ax.get_ylim()
    ymax = ylim[0] + 1.1 * (ylim[1] - ylim[0])
    if ymax>1.0: ymax=1.0
    ax.set_ylim(0, ymax)
    ax.set_ylabel('folded power', fontproperties=font, fontsize=14)
    ax.set_xlabel('frequency (d$^{-1}$)', fontproperties=font, fontsize=14)

    # ZTF
    # =====

    # PLOTTING THE FULL PERIODOGRAM
    # -----
    ax = fig.add_subplot(143)

    # plotting the periodogram
    ax.plot(fL, pL, c=blue) # plotting basic periodogram
    # adding the structure lines
    ax.plot([fztf, fztf], [0,1], lw = 1, c=pink, ls='--')
    ax.plot([fBlazhkoPeakZ, fBlazhkoPeakZ], [0, 0.7*np.max(pFoldedZ)], lw = 1, c=pink,
ls='--')
    ax.plot([2*fztf-fBlazhkoPeakZ, 2*fztf-fBlazhkoPeakZ], [0, 0.7*np.max(pFoldedZ)], lw =
1, c=pink, ls='--')

    # show 1 year alias for ztf
    flyrZ = fztf+1/365.0
    ax.plot([flyrZ, flyrZ], [0, 0.7*np.max(pFoldedZ)], lw = 1, ls='-.', c=green)
    flyrZ = fztf-1/365.0
    ax.plot([flyrZ, flyrZ], [0, 0.7*np.max(pFoldedZ)], lw = 1, ls='-.', c=green)

    # adding y-axis text
    ax.text(0.03, 0.96, "ZTF", ha='left', va='top',
transform=ax.transAxes, fontproperties=font, fontsize=10)
    if (fBlazhkoPeakZ > fztf*fac):
        ax.set_xlim(0.99*(2*fztf-fBlazhkoPeakZ), 1.01*fBlazhkoPeakZ)
    else:
        ax.set_xlim(fztf/fac, fztf*fac)
    ax.yaxis.set_major_locator(plt.MaxNLocator(4))

    ylim = ax.get_ylim()
    ymax = ylim[0] + 1.1 * (ylim[1] - ylim[0])
    if ymax>1.0: ymax=1.0
    ax.set_ylim(0, ymax)
    ax.set_ylabel('Lomb-Scargle power', fontproperties=font, fontsize=14)

```

```

        ax.set_xlabel('frequency (d$^{-1}$)', fontproperties=font, fontsize=14)

        # PLOTING FOLDED POWER SEQUENCE
        # -----
        ax = fig.add_subplot(144)

        ax.plot(fFoldedZ, pFoldedZ, c=blue)
        ax.plot([fBlazhkoPeakZ, fBlazhkoPeakZ], [0, 0.4*np.max(pFoldedZ)], lw = 1, ls='--',
c=pink)

        # show 1 year alias
        flyrZ = fztf+1/365.0
        ax.plot([flyrZ, flyrZ], [0, 0.4*np.max(pFoldedZ)], lw = 1, ls='-.', c=green)

        powerFarZ = pFoldedZ[fFoldedZ>fBlazhkoPeakZ] # frequencies beyond the second peak
        powerFarMedianZ = np.median(powerFarZ)          # the median power
        powerFarRMSZ = np.std(powerFarZ)                # standard deviation, i.e. "sigma"
        noise5sigZ = powerFarMedianZ+5*powerFarRMSZ

        if (fBlazhkoPeakZ > fztf*fac):
            ax.plot([fztf+0.5*(fBlazhkoPeakZ-fztf), 1.01*fBlazhkoPeakZ], [noise5sigZ,
noise5sigZ], lw = 1, ls='--', c=turquoise)
            ax.set_xlim(flin, 1.01*fBlazhkoPeakZ)
        else:
            ax.plot([flin+0.5*(fBlazhkoPeakZ-fztf), fztf*fac], [noise5sigZ, noise5sigZ], lw =
1, ls='--', c=turquoise)
            ax.set_xlim(fztf, fztf*fac)

        ax.yaxis.set_major_locator(plt.MaxNLocator(4))

        ylim = ax.get_ylim()
        ymax = ylim[0] + 1.1 * (ylim[1] - ylim[0])
        if ymax>1.0: ymax=1.0
        ax.set_ylim(0, ymax)
        ax.set_ylabel('folded power', fontproperties=font, fontsize=14)
        ax.set_xlabel('frequency (d$^{-1}$)', fontproperties=font, fontsize=14)

        if plotSave:
            plotName = '../images/Blazhko.png'
            plt.savefig(plotName, dpi=750, bbox_inches = 'tight')
            #print('saved plot as:', plotName)
        plt.show()
        return

def plotLINEARmarkSeasons(Lid, ztf_data, order, LINEARlightcurves, plotName='season_plot',
plotSave=False):
    tL, mL, mLerr = LINEARlightcurves[Lid].T
    fig, ax = plt.subplots(1,2, figsize=(32,8))
    ax[0].set_xlim(np.min(mL)-0.3, np.max(mL)+0.3)
    ax[0].set_title('LINEAR object {0}'.format(Lid), fontproperties=font, fontsize=14)
    ax[0].errorbar(tL, mL, mLerr, fmt='.b', ecolor=blue)
    ax[0].set_xlabel('Time (days)', fontproperties=font, fontsize=14)
    ax[0].set_ylabel('LINEAR magnitude', fontproperties=font, fontsize=14)
    ax[0].invert_yaxis()
    ax[0].set_xlim(np.min(tL)-200, np.max(tL)+200)

    redL = 0

    for s in range(1, 8):
        tS = 52550 + (s-1)*365
        per = np.median(mL)
        ax[0].plot([tS, tS], [per-per*0.05, per+per*0.05], c=pink)
        if tS>np.min(tL)-200 and tS<np.max(tL)+200:
            redL += 1

    tZ, mZ, meZ = ztf_data[order][1], ztf_data[order][2], ztf_data[order][3]
    ax[1].set_xlim(np.min(mZ)-0.3, np.max(mZ)+0.3)
    ax[1].errorbar(tZ, mZ, meZ, fmt='.b', ecolor=blue)
    ax[0].set_title('ZTF object {0}'.format(order), fontproperties=font, fontsize=14)
    ax[1].set_xlabel('Time (days)', fontproperties=font, fontsize=14)
    ax[1].set_ylabel('ZTF magnitude', fontproperties=font, fontsize=14)

```

```

ax[1].invert_yaxis()
ax[1].set_xlim(np.min(tZ)-200, np.max(tZ)+200)

redZ = 0

for r in range(1, 8):
    tSZ = (np.min(tZ)-50) + (r-1)*365
    ax[1].plot([tSZ, tSZ], [np.min(mZ)-0.1, np.max(mZ)+0.1], c=pink)
    if tSZ>np.min(tZ)-200 and tSZ<np.max(tZ)+200:
        redZ += 1

if plotSave:
    plt.savefig('../images/'+plotName+'.png', dpi=750)
plt.show()

return redL, redZ

def makeLCplotBySeason(Lid, L1, tL, L2, tZ, redL, redZ, plotrootname='LCplotBySeason',
plotSave=False):

    fig = plt.figure(figsize=(32, 30))
    fig.subplots_adjust(hspace=0.2, bottom=0.06, top=0.94, left=0.12, right=0.94)

    fig.suptitle('Seasons for:' + str(Lid), fontsize=30, fontproperties=font)

    def plotPanelL(ax, L1, season):
        ax.set_xlabel('phase', fontproperties=font, fontsize=14)
        ax.set_ylabel('normalized phased light curve', fontproperties=font, fontsize=14)
        ax.set_xlim(-0.1, 1.1)
        ax.set_ylim(1.3, -0.4)
        # fit for Plinear
        ax.plot(L1['modelPhaseGrid'], L1['modTemplate'], pink, markeredgecolor=pink, lw=1,
fillstyle='top', linestyle='dashed')

        # data
        xx, yy, zz, ww = sort4arr(L1['dataPhasedTime'], L1['dataTemplate'],
L1['dataTemplateErr'], tL)
        tSmin = 52520 + (season-1)*365
        tSmax = 52520 + season*365
        condition = (ww > tSmin) & (ww < tSmax)
        xxS = xx[condition]
        yyS = yy[condition]
        zzS = zz[condition]
        wwS = ww[condition]
        ax.errorbar(xxS, yyS, zzS, fmt='.b', ecolor=blue, lw=1, ms=4, capsiz=1.5, alpha=0.3)
        textString = "LINEAR season " + str(season)
        ax.text(0.03, 0.96, textString, ha='left', va='top', transform=ax.transAxes,
fontproperties=font, fontsize=14)
        textString = "MJD=" + str(tSmin) + ' to ' + str(tSmax)
        ax.text(0.53, 0.96, textString, ha='left', va='top', transform=ax.transAxes,
fontproperties=font, fontsize=14)

    # plot each season separately
    for season in range(1,redL):
        # plot the power spectrum
        ax = fig.add_subplot(5, 3, season)
        plotPanelL(ax, L1, season)
        if (season==1):
            ax.set_title('LINEAR object {0}'.format(Lid), fontproperties=font, fontsize=18)

    # =====
    # ZTF
    # =====

    def plotPanelZ(ax, L2, seasonZ):
        ax.set_xlabel('phase', fontproperties=font, fontsize=14)
        ax.set_ylabel('normalized phased light curve', fontproperties=font, fontsize=14)
        ax.set_xlim(-0.1, 1.1)
        ax.set_ylim(1.3, -0.4)
        # fit for Plinear

```

```

        ax.plot(L2['modelPhaseGrid'], L2['modTemplate'], pink, markeredgecolor=pink, lw=1,
fillstyle='top', linestyle='dashed')

        # data
        xx, yy, zz, ww = sort4arr(L2['dataPhasedTime'], L2['dataTemplate'],
L2['dataTemplateErr'], tZ)
        tSmin = (np.min(tZ)-50) + (seasonZ-1-6)*365
        tSmax = (np.min(tZ)-50) + (seasonZ-6)*365
        xxS = xx[(ww>tSmin) & (ww<tSmax)]
        yyS = yy[(ww>tSmin) & (ww<tSmax)]
        zzS = zz[(ww>tSmin) & (ww<tSmax)]
        wws = ww[(ww>tSmin) & (ww<tSmax)]
        ax.errorbar(xxS, yyS, zzS, fmt='.b', ecolor=blue, lw=1, ms=4, capsiz=1.5, alpha=0.3)
        textString = "ZTF season " + str(seasonZ-6)
        ax.text(0.03, 0.96, textString, ha='left', va='top', transform=ax.transAxes,
fontproperties=font, fontsize=14)
        textString = "MJD=" + str(tSmin) + ' to ' + str(tSmax)
        ax.text(0.53, 0.96, textString, ha='left', va='top', transform=ax.transAxes,
fontproperties=font, fontsize=14)

# plot each season separately
for seasonZ in range(redL,redL+redZ-1):
    # plot the power spectrum
    ax = fig.add_subplot(5, 3, seasonZ)
    plotPanelZ(ax, L2, seasonZ)
    if (seasonZ==1):
        ax.set_title('ZTF object {0}'.format(Lid), fontproperties=font, fontsize=18)

if plotSave:
    plotName = plotrootname + '.png'
    plt.savefig('../images/' + plotName, dpi=600, bbox_inches = 'tight')
    #print('saved plot as:', plotName)
plt.show()
return

def plotAll(Lid, orderlc, o, tot, L1, L2, blazhko_can, fL, pL, fZ, pZ, fFoldedL, fFoldedZ,
pFoldedL, pFoldedZ, data, tL, tZ, ztf_data, plotSave=False):
    if plotSave:
        makeLCplot_info(L1, L2, blazhko_can, o, Lid, data, tot, plotSave=True)
        plotBlazhkoPeaksLINEAR(Lid, o, fL, pL, fZ, pZ, fFoldedL, pFoldedL, fFoldedZ, pFoldedZ,
blazhko_can, fac=1.008, plotSave=True, verbose=True)
        redLin, redZtf = plotLINEARmarkSeasons(Lid, ztf_data, orderlc, data, plotSave=True)
        makeLCplotBySeason(Lid, L1, tL, L2, tZ, redLin, redZtf, plotSave=True)
    else:
        makeLCplot_info(L1, L2, blazhko_can, o, Lid, data, tot)
        plotBlazhkoPeaksLINEAR(Lid, o, fL, pL, fZ, pZ, fFoldedL, pFoldedL, fFoldedZ, pFoldedZ,
blazhko_can, fac=1.008, plotSave=False, verbose=True)
        redLin, redZtf = plotLINEARmarkSeasons(Lid, ztf_data, orderlc, data)
        makeLCplotBySeason(Lid, L1, tL, L2, tZ, redLin, redZtf)
    return

blazhko_can = pd.DataFrame(())
blazhko_can = blazhko_determine(lc_analysis, blazhko_can)
blazhko_can.to_csv("../outputs/blazhko_can"+end+".csv", index=False)

# ZTF PERIODOGRAM
# =====
ztf_per = category_analysis(blazhko_can, fits, periodogr, ztf_data, dataL,
'ztf_periodogram', id_list=None, parameter='IndicatorType', value='Z')

blazhko_analyzer = ztf_per.get_save_data()
end = 'ztf_periodogram'
blazhko_analyzer.to_csv("../outputs/blazhko_list_" + end + ".csv", index=False)

print(blazhko_analyzer.shape)

# LINEAR PERIODOGRAM
# =====
linear_per = category_analysis(blazhko_can, fits, periodogr, ztf_data, dataL,
'linear_periodogram', id_list=None, parameter='IndicatorType', value='L')

```

```

blazhko_analyzerl = linear_per.get_save_data()
end = 'linear_periodogram'
blazhko_analyzerl.to_csv("../outputs/blazhko_list_"+end+".csv", index=False)

print(blazhko_analyzerl.shape)

# ZTF CHI2
# =====
ztf_chi = category_analysis(blazhko_can, fits, periodogr, ztf_data, dataL,
'ztf_chi', id_list=None, parameter='ChiType', value='Z')

blazhko_analyzerzc = ztf_chi.get_save_data()
end = 'ztf_chi'
blazhko_analyzerzc.to_csv("../outputs/blazhko_list_"+end+".csv", index=False)

print(blazhko_analyzerzc.shape)

# LINEAR CHI2
# =====
linear_chi = category_analysis(blazhko_can, fits, periodogr, ztf_data, dataL,
'linear_chi', id_list=None, parameter='ChiType', value='L')

blazhko_analyzerlc = linear_chi.get_save_data()
end = 'linear_chi'
blazhko_analyzerlc.to_csv("../outputs/blazhko_list_"+end+".csv", index=False)

print(blazhko_analyzerlc.shape)

# OTHER
# =====
be_other = blazhko_can[(blazhko_can['IndicatorType']!='Z') & (blazhko_can['IndicatorType']!='L') & (blazhko_can['ChiType']!='Z') & (blazhko_can['ChiType']!='L')]
be_other = be_other.reset_index(drop=True)
print(be_other.shape)
be_other.head()

BE_candidates_other = category_analysis(be_other, fits, periodogr, ztf_data, dataL, 'other')

blazhko_analyzerother = BE_candidates_other.get_save_data()
end = 'other'
blazhko_analyzerother.to_csv("../outputs/blazhko_list_"+end+".csv", index=False)

print(blazhko_analyzerother.shape)

BE_LIST = list(blazhko_analyzerlc['LINEAR id']) + list(blazhko_analyzerzc['LINEAR id']) +
list(blazhko_analyzerl['LINEAR id']) + list(blazhko_analyzer['LINEAR id']) +
list(blazhko_analyzerother['LINEAR id'])
BE_LIST = set(BE_LIST)
print(len(BE_LIST))

FINAL_BE_CATALOGUE = pd.DataFrame()
FINAL_BE_CATALOGUE = pd.concat([FINAL_BE_CATALOGUE,
blazhko_analyzer.reset_index(drop=True), ignore_index=True, axis=0])
FINAL_BE_CATALOGUE = pd.concat([FINAL_BE_CATALOGUE,
blazhko_analyzerzc.reset_index(drop=True), ignore_index=True, axis=0])
FINAL_BE_CATALOGUE = pd.concat([FINAL_BE_CATALOGUE,
blazhko_analyzerl.reset_index(drop=True), ignore_index=True, axis=0])
FINAL_BE_CATALOGUE = pd.concat([FINAL_BE_CATALOGUE,
blazhko_analyzerlc.reset_index(drop=True), ignore_index=True, axis=0])
FINAL_BE_CATALOGUE = pd.concat([FINAL_BE_CATALOGUE,
blazhko_analyzerother.reset_index(drop=True), ignore_index=True, axis=0])

print(FINAL_BE_CATALOGUE.shape)
FINAL_BE_CATALOGUE.head()

SHOWCASE_Catalogue1 = pd.DataFrame()
SHOWCASE_Catalogue1['LINEAR id'] = FINAL_BE_CATALOGUE['LINEAR id']
SHOWCASE_Catalogue1['LINEAR period'] = FINAL_BE_CATALOGUE['Plinear']
SHOWCASE_Catalogue1['ZTF period'] = FINAL_BE_CATALOGUE['Pztf']
SHOWCASE_Catalogue1['LINEAR hi-kvadrat'] = FINAL_BE_CATALOGUE['L_chi2dofR']

```

```
SHOWCASE_Catalogue1['ZTF hi-kvadrat'] = FINAL_BE_CATALOGUE['Zchi2dofR']
SHOWCASE_Catalogue1['LINEAR amplituda'] = FINAL_BE_CATALOGUE['Lampl']
SHOWCASE_Catalogue1['ZTF amplituda'] = FINAL_BE_CATALOGUE['Zampl']
SHOWCASE_Catalogue1['LINEAR glavna frekvencija'] = FINAL_BE_CATALOGUE['MainPeakL']
SHOWCASE_Catalogue1['LINEAR blazhko frekvencija'] = FINAL_BE_CATALOGUE['BlazhkoPeakL']
SHOWCASE_Catalogue1['ZTF glavna frekvencija'] = FINAL_BE_CATALOGUE['MainPeakZ']
SHOWCASE_Catalogue1['ZTF blazhko frekvencija'] = FINAL_BE_CATALOGUE['BlazhkoPeakZ']
SHOWCASE_Catalogue1['LINEAR blazhko period'] = FINAL_BE_CATALOGUE['BlazhkoPeriodL']
SHOWCASE_Catalogue1['ZTF blazhko period'] = FINAL_BE_CATALOGUE['BlazhkoPeriodZ']
```