Pronalaženje *Blazhko* zvijezda izvan skupova zvijezda

Natjecanje iz astronomije 2024.

2. razred SŠ, jedan natjecatelj

[Sažetak Rada 3](#_Toc161845579)

[1 UVOD 4](#_Toc161845580)

[1.1 Motivacija i cilj 4](#_Toc161845581)

[1.2 Promjenjive zvijezde 5](#_Toc161845582)

[1.2.1 Tipovi promjenjivih zvijezda 5](#_Toc161845583)

[1.2.2 Pulsirajuće promjenjive zvijezde 5](#_Toc161845584)

[1.2.3 Povezanost perioda i oscilacije slojeva promjenjivih zvijezda 7](#_Toc161845585)

[1.2.4 Svjetlosne krivulje 8](#_Toc161845586)

[1.2.5 RR Lire zvijezde 9](#_Toc161845587)

[1.3 Blazhko efekt 10](#_Toc161845588)

[2 METODOLOGIJA 13](#_Toc161845589)

[2.1 Dohvat i priprema podataka 13](#_Toc161845590)

[2.1.1 **LINEAR** podaci 13](#_Toc161845591)

[2.1.2 Selekcija RR Lira iz **LINEAR** podataka 14](#_Toc161845592)

[2.1.2 **ZTF** podaci 16](#_Toc161845593)

[2.4 Analiza RR Lira 18](#_Toc161845594)

[2.4.1 Računanje preciznog perioda za RR Lire 18](#_Toc161845595)

[2.4.2 Algoritam analize periodograma 19](#_Toc161845596)

[2.4.3 Računanje najboljeg *fita* za svjetlosnu krivulju 20](#_Toc161845597)

[2.5 Selekcija kandidata za Blazhko zvijezde 21](#_Toc161845598)

[2.6 Analiza Blazhko efekt kandidata 22](#_Toc161845599)

[3 REZULTATI 26](#_Toc161845600)

[3.1 Analiza RR Lira 26](#_Toc161845601)

[3.2 Analiza RR Lira i pronalazak Blazhko zvijezda 28](#_Toc161845602)

[3.2.1 Detaljan opis 3 Blazhko zvijezde: 810169, 1736308 i 10420063 29](#_Toc161845603)

[4 RASPRAVA I ZAKLJUČAK 34](#_Toc161845604)

[5 LITERATURA I IZVORI 38](#_Toc161845605)

[5.1 Popis slika 40](#_Toc161845606)

[5.2 Popis grafova 40](#_Toc161845607)

[5.3 Popis tablica 41](#_Toc161845608)

[6 PRILOG – programi 42](#_Toc161845609)

# Sažetak Rada

U ovom praktičnom radu bavili smo se potragom za posebnom vrstom promjenjivih pulsirajućih zvijezda, **Blazhko zvijezdama.** 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 zvijezdama kroz promjenu perioda, amplitude te sporednih frekvencija koje utječu na period u periodogramu, što je graf različitih mogućih frekvencija za period 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 glavni podaci dolazili od LINEAR pregleda neba, a sporedni podaci za usporedbu i za potvrdu modulacije krivulje sjaja je ZTF pregled neba. Od početnih 7010 LINEAR zvijezda s pogodnim krivuljama sjaja (više od 250 točaka promatranja), odabrali smo 2914 RR Lira te preko njihovih ekvatorijalnih koordinata pronašli dotične ZTF parove LINEAR zvijezdama.

Zatim smo analizirali svjetlosne krivulje tako što smo računali period pomoću 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.

Nakon početne analize, napravili smo algoritam koji prolazi kroz svih 1703 dobrih Blazhko kandidata 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 57 do sada nepoznatih Blazhko zvijezda, od kojih je 82.5 % RRab tipa, a 17.5 % RRc tipa što je u skladu sa dosadašnjim istraživanjima Blazhko zvijezda. Od početnog skupa podataka kojeg smo analizirali, samo oko 2 % 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. Glavni faktor koji možda utječe na nizak broj Blazhko zvijezda je manja preciznost LINEAR i ZTF podataka nego u drugim istraživanjima.

# 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 funkcionira.

Ciljevi ovog praktičnog rada su:

* Istražiti promjenjive zvijezde te kako funkcioniraju,
* 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 *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 funkcionira zbog fizičkih procesa unutar same zvijezde. Ekstrinzične promjenjive zvijezde mijenjaju svoj sjaj zbog vanjskih utjecaja, kao npr. ekliptične binarne zvijezde, zvjezdane 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 zvjezdane 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 neperiodi*č*no-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 povećavati svoj volumen ili smanjivati ga koji gura ostale vanjske slojeve unutrašnjosti zvijezde da mijenjaju svoju veličinu. Možemo pretpostaviti da se u ovom sloju nalazi plin koji se ponaša kao *idealan plin.* Idealni plinovi ponašaju se po idućem zakonu:

|  |  |  |
| --- | --- | --- |
|  |  | 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 zvjezdane 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]

A graph of a function

Description automatically generated

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

|  |  |  |
| --- | --- | --- |
|  |  | 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.

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 prodor fotona kroz sloj zvijezde.

A diagram of a hydrogen molecule

Description automatically generatedA diagram of a molecule

Description automatically generated

Slika 2: prikaz interakcije fotona i atoma i [6]

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

1. Gornji slojevi zvijezde zbog utjecaja gravitacije djeluju velikom silom usmjerenom prema središtu, volumen zvijezde se smanjuje,
2. 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,
3. Povećanjem neprozirnosti fotoni iz jezgre ne mogu jednako efikasno putovati do površine zvijezde, smanjujući ukupnu svjetlost zvijezde, odnosno sjaj je najmanji u ciklusu,
4. Velika količina fotona koji ne prolaze kroz dotičan sloj su energični te pritisak i temperatura počinju se povećavati od dotičnog, ali i slojeva ispod,
5. Povećanje pritiska i temperature pobjeđuje silu gravitacije i povećava ukupan volumen zvijezde, gurajući dotičan sloj prema površini,
6. 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 te radijus je najmanji i temperatura najviša,
7. Sloj se ponovno vraća na staru lokaciju zbog nedostatka pritiska koji djeluje na sloj te gravitacija opet kreće djelovati na dotičan sloj,
8. 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**. Možemo ovaj 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 modelirati tako što jedan kraj stojnog vala učvrstimo za zatvoren kraj cijevi te on titra tako što mu vrh dotakne sam rub otvorenog dijela cijevi. [4,5,6]

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

A diagram of a mathematical equation

Description automatically generated

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 kompliciranije pulsirati, tako što postoje 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 traži 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 mjere 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 svijetlosti 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.

A graph of a graph of a graph

Description automatically generated with medium confidence

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]

A graph of a graph of a graph

Description automatically generated with medium confidence

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 prvome tonu. Također, imaju manju promjenu magnitude od RRab tipa. [14, 15, 16, 17, 18, 19]

A graph of a line

Description automatically generated with medium confidence

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]

A graph of a phase

Description automatically generated

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 funkcionira 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]

A graph of a function

Description automatically generatedA graph of a graph showing a number of numbers

Description automatically generated with medium confidence

Slika 4: prikaz Blazhko efekta kod RR Lira [?]

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:

|  |  |  |
| --- | --- | --- |
|  |  | 3) |

gdje je amplituda, frekvencija vala i 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:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 4) | |
|  |  | 5) |

Dobili bismo iduću jednadžbu:

|  |  |  |
| --- | --- | --- |
|  |  | 6) |

Budući da su obje frekvencije vrlo slične u vrijednosti, možemo prvu zagradu interpretirati kao , a drugu kao , odnosno prosječnu frekvenciju. Budući da je razlika u frekvenciji mnogo manja od prosječne frekvencije, 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:

A graph of a graph showing a number of different colored lines

Description automatically generated with medium confidence

Slika 5: interferencija valova pri malom period promatranja

A graph showing a number of different colors

Description automatically generated with medium confidence

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

A blue and pink waves

Description automatically generated

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. LINEAR je primarna baza podataka te ZTF uglavnom služi za provjeravanja LINEAR vjerodostojnosti. U ovom praktičnom radu koristili smo sljedeće materijale:

* Računalo s pristupom internetu
* Web stranicu GitHub
* Programski jezik Python
* Jupyter notebooks – Anaconda 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 kalibrirani (baždareni), odnosno znanstvenici au morali osigurati da koordinate objekata u podacima odgovaraju drugom istraživanju, poput SDSS-a. 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 u podacima i ispravljanje podcijenjenih pogrešaka magnitude. Pošto je kalibracija odrađena, 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 kategorizirajući krivulje svjetlosti, izračunavajući njihove periode te izvodeći zaključke iz prethodnih faktora. [30,31]

**LINEAR** podaci nalaze se unutar Python modula *AstroML*[[1]](#footnote-1)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** je sljedeća:

1. dataL je objekt AstroML.datasets i sadrži sljedeće:
   1. dataL.ids popis je svih ID-ova zvijezda, gdje je ID identifikacijski broj zvijezde, svaka ima jedinstveni broj
   2. 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** **magnitude**. 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  '''  # ----------------------  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

A table of numbers with black text

Description automatically generated

Gdje su značenja stupaca sljedeća:

1. **ra** – rektascencija
2. **dec** – deklinacija
3. **ug** – UG indeks boje (Ultraviolet – Green ili Ultraljubičasto – Zeleno)
4. **gi** – GI indeks boje (Green – Infrared ili Zeleno – Infracrveno)
5. **iK** – dodatni infracrveni indeks boje
6. **JK** - dodatni infracrveni indeks boje
7. **logP** – logaritam perioda (u danima) zvijezde
8. **Ampl** – amplituda promjene sjaja (u magnitudama)
9. **Skew** – asimetričnost vrijednosti magnituda svjetlosnih krivulja [12]
10. **Kurt** – kurtozija vrijednosti magnituda svjetlosnih krivulja [12]
11. **magMed** – medijan (srednjak) 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 mi 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**) # resetiraj indeks  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* iliti 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 IPAC[[2]](#footnote-2) institutu u Kaliforniji, kojem se može pristupiti putem IRSA[[3]](#footnote-3) 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]

#### 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 ZTF-a 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 ZTF-a. 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 mjerenja) te filterom. Zatim se brišu očite lažne točke. Ako program nije uspio pronaći par LINEAR ID-u, sprema se prazna tablica podataka.

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

**if** os.path.isfile(NAME): # ako ova datoteka postoji

ZTF\_data\_lc = np.load(NAME, allow\_pickle=**True**) # učitaj podatke

**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 izduženija 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*.

Svjetlosnu 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 1.3.

A graph with red and blue lines

Description automatically generated

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.

A graph of a graph of a graph

Description automatically generated with medium confidence

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

Graf prikazuje jedan glavni 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 iz dijela 1.3. 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 funkcionira na sljedeći način:

1. Dijeli periodogram na 2 dijela: lijevi dio, sve frekvencije prije glavnog maksimuma te desni dio, sve frekvencije nakon glavnog maksimuma,
2. 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,
3. For petlja prolazi kroz svaki element 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
   1. 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
   2. Ako petlja ne pronađe ovaj vrh, smatra se da periodogram ne prikazuje Blazhko zvijezdu te se spremaju prazne vrijednosti za blazhko parametre.

### 2.4.3 Računanje najboljeg *fita* 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]

|  |  |  |
| --- | --- | --- |
|  |  | 7) |

Gdje je hi-kvadrat, promatrana varijabla odnosno vrijednost (magnituda svjetlosne krivulje), s je greška mjerenja, te 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, rezultirajući u visoku vrijednost hi-kvadrata.

Koristeći sve prethodno objašnjene metode, analiziram svih 2941 parova LINEAR i ZTF zvijezda, gdje računam sve dotične parametre samo za parove gdje LINEAR podaci imaju više od 250 promatranja. Nakon analize, potrebno je malo počistiti podatke od zvijezda s ZTF podacima koji imaju premalo promatranja ili nemaju podatke. Granica za ZTF količinu podataka je 40. Nakon rezanja broja zvijezda ostalo je 1703 s dovoljnom količinom podataka.

## 2.5 Selekcija kandidata za Blazhko zvijezde

Nakon analize RR Lira, potrebno je birati zvijezde koje su potencijalno Blazhko zvijezde. Stvaram algoritam koji sistematično određuje dobre zvijezde za Blazhko efekt na sljedeći način:

1. **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, ne uzimamo u obzir,
2. **Periodogram zvijezde:** ako su svi kriteriji zadovoljeni vezani za pokazatelje Blazhko efekta preko periodograma, zvijezda je automatski Blazhko kandidat. Kriteriji su:
   1. ne smije postojati dnevni alias perioda (tj. lažni vrh zbog utjecaja dnevnog ciklusa promatranja),
   2. Blazhko vrh mora imati relativnu snagu veću od 0.05, odnosno mora biti viši od pozadinskog šuma drugih frekvencija,
   3. statistička značajnost Blazhko vrha mora biti veća od 5,
   4. Blazhko period odnosno period ponavljanja Blazhko modulacije mora biti između 35 i 325 dana [22,18].
3. **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,
   1. **Amplituda:** 
      1. , vrijedno 1 bod,
      2. , vrijedno 2 boda.
   2. **Razlika perioda:** računa se tako da ZTF period oduzme od LINEAR perioda te podijeli sa prosjekom perioda,
      1. , vrijedno 2 boda,
      2. , vrijedno 4 boda.
   3. **Vrijednost hi-kvadrata ZTF i LINEAR podataka:**
      1. Oba hi-kvadrata su između 2.5 i 4.5, vrijedno 4 boda,
      2. Oba hi-kvadrata su iznad 4.5, vrijedno 6 bodova,
      3. Samo LINEAR ili samo ZTF između 2.5 i 4.5, vrijedno 2 boda,
      4. Samo LINEAR ili samo ZTF hi-kvadrat je iznad 4.5, vrijedno 3 boda.

Konačna bodovna skala imala je maksimum od 12 mogućih postignutih bodova za svaku svjetlosnu krivulju te su birane sve zvijezde s više od 5 bodova (od 6 nadalje).

## 2.6 Analiza Blazhko efekt kandidata

Nakon biranje Blazhko kandidata pomoću algoritma objašnjenog iznad, potrebno je *vizualno analizirati Blazhko kandidate 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:

1. **LINEAR periodogrami:** svi Blazhko kandidati koji su prema LINEAR periodogramu bili izabrani kao kandidat,
2. **ZTF periodogrami:** svi Blazhko kandidati koji su prema ZTF periodogramu bili izabrani kao kandidat,
3. **LINEAR hi-kvadrat:** svi Blazhko kandidati koji su prema vrijednosti LINEAR hi-kvadrata bili izabrani kao kandidat,
4. **ZTF hi-kvadrat:** svi Blazhko kandidati koji su prema vrijednosti ZTF hi-kvadrata bili izabrani kao kandidat,
5. **Ostalo:** svi kandidati koji nisu zadovoljili nijedan kriterij za grupe iznad.

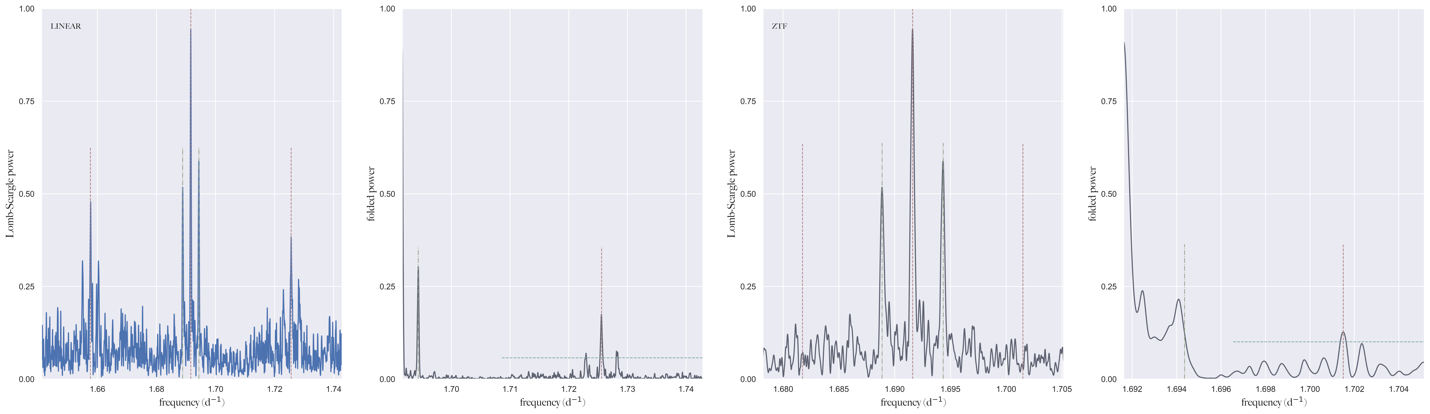
Vizualna analiza provodi se kroz 4 faza promatranja 4 različitih grafova prikazani u sučelju. Prva faza je analiza faze RR Lira LINEAR i ZTF podataka te pregledavanja razlike u periodu.

A graph with a line

Description automatically generated

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.

A graph of a graph

Description automatically generated with medium confidence

Slika 12: treća faza vizualne analize Blazhko kandidata

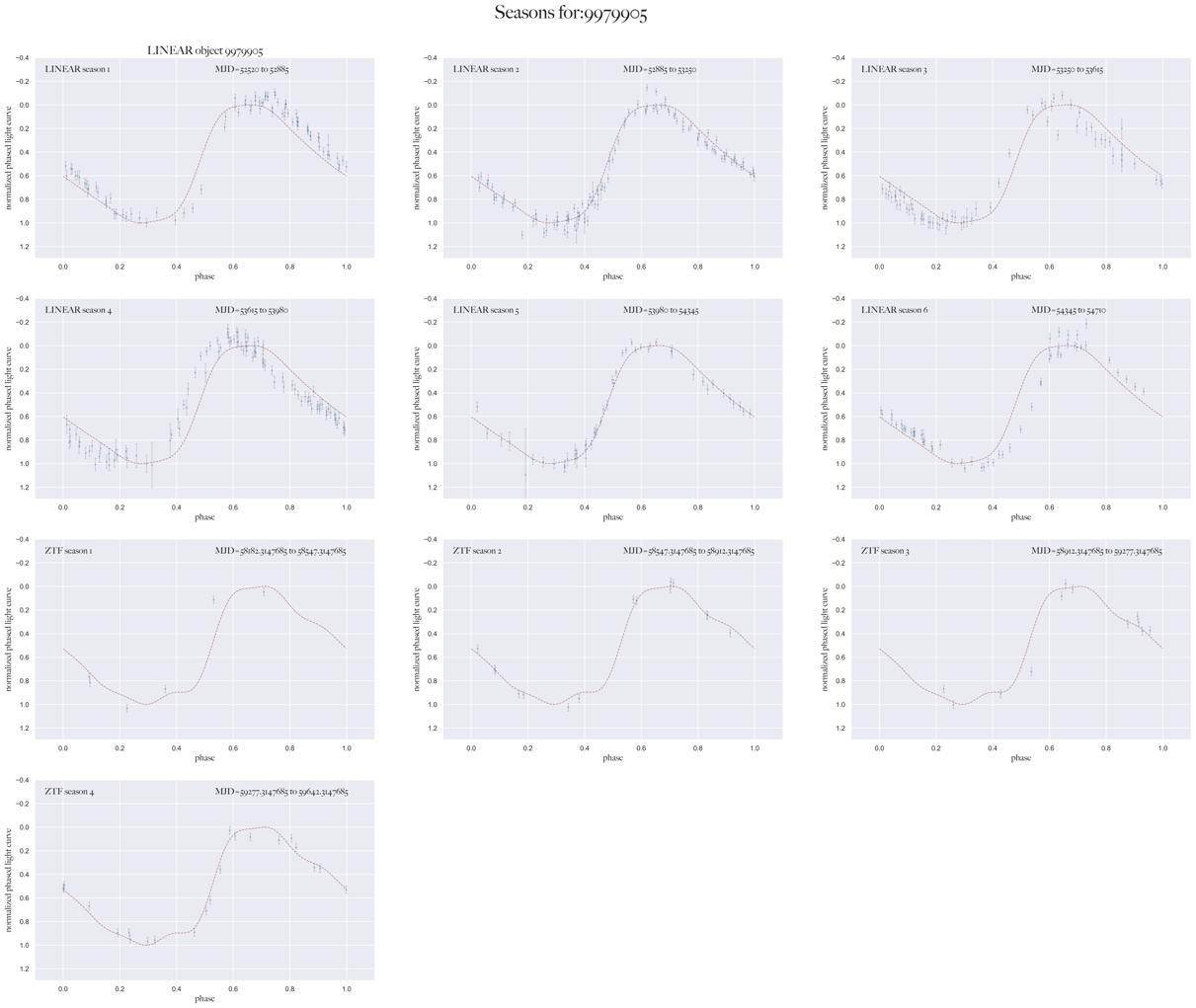
Treća faza uključuje promatranje ukupne sezone promatranja LINEAR i ZTF podataka za pojedinu zvijezdu. Potrebno je vidjeti je li magnituda konstantna tokom čitavog vremenskog perioda promatranja ili postoji modulacija amplitude. U primjeru iznad, za LINEAR podatke možemo vidjeti da postoji blaga razlika 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.

A graph of different types of graphs

Description automatically generated with medium confidence

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 sezonama. Gleda se postoji li **modulacija amplitude** ili **fazna modulacija**. U primjeru iznad vidljiva je blaga modulacija amplitude, 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 amplitude.



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

A graph of a number of people

Description automatically generated

Graf 1: raspodjela mjerenja kod LINEAR RR Lira

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

A graph of a number of different colored lines

Description automatically generated with medium confidence

Graf 2: raspodjela mjerenja kod ZTF RR Lira

Vidimo da distribucije svih filtera imaju sličan oblik, međutim infracrveni grafikon ima manji raspon x-osi. Zeleni filtar najviše zvijezda ima broj mjerenja 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 filtar 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 sezonama 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:

A graph with numbers and lines

Description automatically generated

Graf 3: distribucija vremenskih perioda LINEAR zvijezda

A graph of a graph

Description automatically generated with medium confidence

Graf 4: 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.

A diagram of a star

Description automatically generated with medium confidence

Graf 5: 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 30º. 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 1703 zvijezda koje imaju više od 250 podataka (u LINEAR setu podataka).

Pomoću algoritma za prepoznavanje Blazhko kandidata, selektirano je 489 zvijezda. Zatim su one bile analizirane vizualno, procesom objašnjenim u §2.6. Nakon vizualne analize, bilo je selektirano sveukupno 57 Blazhko zvijezda kao konačne zvijezde iz početnog skupa 2914 RR Lira, od koji smo mogli obraditi samo 1703 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

A screenshot of a table

Description automatically generated

### 3.2.1 Detaljan opis 3 Blazhko zvijezde: 810169, 1736308 i 10420063

U ovom dijelu opisujemo 3 primjera Blazhko zvijezda za ilustraciju načina biranja Blazhko zvijezda te njihovih značajki.

#### 3.2.1.1 Zvijezda 810169

A graph with a line

Description automatically generatedA graph of a graph of a graph

Description automatically generated with medium confidenceA comparison of a graph

Description automatically generated

A graph of different types of graphs

Description automatically generated with medium confidence

Slika 15: sučelje za vizualnu analizu Blazhko zvijezde 810169

Iz slika iznad možemo primijetiti da periodogram i kod LINEAR i kod ZTF podataka ne prikazuje signifikantnu prisutnost Blazhko efekta, no to ne znači da zvijezda nije Blazhko. Iz dijagrama za sezone promatranja LINEAR i ZTF podataka sveukupno, vidimo da kod LINEAR podataka ima naznake promjene amplitude. Kada pobliže pogledamo zadnju sliku koja prikazuje pojedinačne sezone, *očita je modulacija i amplitude i perioda* kod LINEAR podataka te je upravo i zato lijep primjer Blazhko zvijezde.

#### 3.2.1.2 Zvijezda 1736308

A graph with a line

Description automatically generated

A graph of a graph of a graph

Description automatically generated with medium confidence

A comparison of a graph

Description automatically generatedA graph of different types of graphs

Description automatically generated with medium confidence

Slika 16: sučelje za vizualnu analizu Blazhko zvijezde 1736308

Već iz faznog prikaza ove zvijezde možemo uočiti amplitudnu modulaciju. Gledajući periodograme i ZTF i LINEAR podataka možemo vidjeti lijep primjer lokalnih Blazhko frekvencija, dajući snažan dokaz prisutnosti Blazhko efekta. Na kraju, grafovi svjetlosti po pojedinim sezonama promatranja uočavamo amplitudnu modulaciju. Iako nije vrlo signifikantna, svejedno je prisutna i pokazuje Blazhko efekt.

#### 3.2.1.3 Zvijezda 10420063

A graph with a line drawn on it

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A comparison of a graph

Description automatically generated

A group of graphs on a white background

Description automatically generated

Slika 17: sučelje za vizualnu analizu Blazhko zvijezde 10420063

Prva slika koja prikazuje jednu fazu svjetlosne krivulje kod LINEAR podataka već naznačuje prisutnost amplitudne modulacije. Periodogram pokazuje mogućnost Blazhko frekvencija, no nažalost ima previše šuma i drugih frekvencija s visokom moću da bismo mogli zasigurno reći da je siguran pokazatelj Blazhko efekta. Gledanjem sezonskih prikaza svjetlosne krivulje vrlo je uočljiva amplitudna modulacija, što pokazuje prisutnost Blazhko efekta.

# 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 **57 Blazhko zvijezda**, odnosno oko 2% od početnog skupa podataka za RR Lire.

A graph with numbers and a number of different numbers

Description automatically generated with medium confidenceA graph showing the amount of time

Description automatically generated

Graf 6: 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. Pobliže možemo analizirati razliku glavne frekvencije (period) i Blazhko frekvencije (Blazhko period) pomoću idućih grafova:

A graph with a line graph

Description automatically generatedA graph with numbers and lines

Description automatically generated

Graf 7: 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.

A graph with dots and numbers

Description automatically generated

Graf 8: 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 svijetlo-plava boja prikazuje Blazhko zvijezde. Većina razlika u periodu su manje od 0.0001 dana, a razlike u vrijednostima hi-kvadrata su primjetne i signifikantne.

A graph with a bar

Description automatically generated

Graf 9: udio RRab i RRc tipova Blazhko zvijezda

Od 57 novootkrivenih Blazhko zvijezda, 82.46 % su RRab tipa, a 17.54 % su RRc tipa. Udio RRab i RRc tipa nije se mnogo promijenio između Blazhko seta zvijezda i početnog skupa zvijezda te je u skladu sa drugim istraživanjima [31, 38] udio RRab mnogo veći od udjela RRc tipa.

A diagram of a star

Description automatically generated with medium confidence

Graf 10: pozicije Blazhko zvijezda pomoću galaktičkih koordinata

Gledajući prostorni raspored Blazhko zvijezda po galaksiji, većina ih se nalazi između 0º i 150º, iznad 30º galaktičke širine. Mali dio zvijezda nalaze se nakon 240º 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, samo 2 % 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 te su ZTF podaci imali manju preciznost od LINEAR podataka, što je onemogućilo kvalitetnu obradu svake zvijezde. 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. Neovisno o poteškoćama tokom potrage za Blazhko zvijezdama, smatramo da smo uspjeli pronaći veliki broj Blazhko zvijezda.

U budućnosti želimo nastaviti analizu ovih Blazhko zvijezda te gledati postoje li neki trendovi među pronađenih 57 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. 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

1. <https://chandra.harvard.edu/graphics/edu/earth_scientist_stars.pdf>
2. <https://openstax.org/books/astronomy-2e/pages/19-3-variable-stars-one-key-to-cosmic-distances>
3. <http://spiff.rit.edu/classes/phys370/lectures/mags/mags.html>
4. <http://spiff.rit.edu/classes/phys370/lectures/pulse_ii/pulse_ii.html>
5. <http://spiff.rit.edu/classes/phys370/lectures/pulse_i/pulse_i.html>
6. <http://spiff.rit.edu/classes/ladder/lectures/varstars/varstars.html#pulse>
7. <https://web.physics.ucsb.edu/~jatila/LambdaLabs/Globulars/HRdiagramlab_JKV.pdf>
8. <https://articles.adsabs.harvard.edu/full/1992ApJS...79..507R/0000512.000.html>
9. <https://astronomy.swin.edu.au/cosmos/h/hertzsprung-russell+diagram>
10. <https://www.animations.physics.unsw.edu.au/jw/beats.htm>
11. <https://phys.libretexts.org/Courses/University_of_California_Davis/UCD%3A_Physics_7C_-_General_Physics/8%3A_Waves/8.6%3A_Beats>
12. https://hrcak.srce.hr/file/175513
13. https://home.ifa.hawaii.edu/users/mendez/ASTRO110LAB11/variables.html
14. Understanding Variable Stars, John R. Percy, Cambridge, izdano 2007. 3. i 6. poglavlje
15. <https://www.assa.org.au/resources/variable-stars/classifying-variable-stars/>
16. <https://www.aavso.org/vsots_rrlyr>
17. <https://en.wikipedia.org/wiki/RR_Lyrae_variable>
18. <https://ogle.astrouw.edu.pl/atlas/RR_Lyr.html>
19. https://www.aavso.org/sites/default/files/Variable%20Star%20Classification%20and%20Light%20Curves%20Manual%202.1.pdf
20. <https://www.eso.org/sci/publications/messenger/archive/no.13-jun78/messenger-no13-15-17.pdf>
21. http://www.physics.smu.edu/sdalley/quarknet/Variable%20Star%20Astronomy.pdf
22. <https://en.wikipedia.org/wiki/Blazhko_effect>
23. <https://arxiv.org/pdf/1309.3969v1.pdf> - Blazhko effect in Cepheids and RR Lyrae stars, Robert Sazbo et al. 2013.
24. <https://ui.adsabs.harvard.edu/abs/2009MNRAS.400.1006J/abstract> - The Konkoly Blazhko Survey: is light-curve modulation a common propery of RRab stars?, Jurcsik, Sodor, et al. 2009.
25. <https://iopscience.iop.org/article/10.1088/1742-6596/118/1/012060/pdf> - K Kolenberg 2008 J. Phys.: Conf. Ser. 118 012060
26. <https://arxiv.org/pdf/1007.3404.pdf> - R. Szabo, Z. Kollath et al. 2010.
27. <https://iopscience.iop.org/article/10.1086/508135/pdf> - A new explanation of the Blazhko effect in RR Lyrae stars, B. Stothers, 2006.
28. <https://academic.oup.com/mnras/article/480/1/1229/5055626> - Netzel, Smolec et al. 2018
29. <https://www.cambridge.org/core/journals/international-astronomical-union-colloquium/article/blazhko-effect-in-rr-lyrae/5A46E7907CCD38501CBB27028CDB0560>
30. <http://faculty.washington.edu/ivezic/Publications/Sesar2011LINEAR.pdf> - Sesar, Stuart, Ivezić et al. 2011.
31. <http://faculty.washington.edu/ivezic/Publications/Palaversa2013LINEAR.pdf> - Palaversa, Ivezić, Eyer et al. 2013.
32. <https://iopscience.iop.org/article/10.1088/1538-3873/aaecbe/pdf> - Bellm, Kulkarni et al. 2019
33. <https://www.aanda.org/articles/aa/pdf/2009/45/aa12851-09.pdf> - Deb and Singh, 2009.
34. <http://faculty.washington.edu/ivezic/Publications/203458.web.pdf> – Ivezić, Vivas, Lupton and Zinn 2004
35. <https://www.youtube.com/watch?v=J5smgeVN38I>
36. <https://www.youtube.com/watch?v=NJIIwIVZPws>
37. <https://www.youtube.com/watch?v=spUNpyF58BY>

## 

## 5.1 Popis slika

[Slika 1: ovisnost neprozirnosti plina o temperature (aproksimativan model) [8] 6](#_Toc161854148)

[Slika 2: prikaz interakcije fotona i atoma i [6] 6](#_Toc161854149)

[Slika 3: model cijevi sa stojnim valovima unutar promjenjive zvijezde (ilustracija autora) 8](#_Toc161854150)

[Slika 4: prikaz Blazhko efekta kod RR Lira [?] 11](#_Toc161854151)

[Slika 5: interferencija valova pri malom period promatranja 12](#_Toc161854152)

[Slika 6: interferencija valova pri duljem period promatranja, modulacija amplitude je vidljiva 12](#_Toc161854153)

[Slika 7: modulacija amplitude preko punog perioda promatranja, dolazi do izražaja 12](#_Toc161854154)

[Slika 8: simulacija svjetlosne krivulje za analizu periodograma 19](#_Toc161854155)

[Slika 9: Lomb-Scargle periodogram frekvencija i perioda simulacije svjetlosne krivulje 20](#_Toc161854156)

[Slika 10: prva faza vizualne analize Blazhko kandidata 23](#_Toc161854157)

[Slika 11: druga faza vizualne analize Blazhko kandidata 23](#_Toc161854158)

[Slika 12: treća faza vizualne analize Blazhko kandidata 24](#_Toc161854159)

[Slika 13: četvrta faza vizualne analize Blazhko kandidata, modulacija amplitude 24](#_Toc161854160)

[Slika 14: četvrta faza vizualne analize Blazhko kandidata, fazna modulacija 25](#_Toc161854161)

[Slika 15: sučelje za vizualnu analizu Blazhko zvijezde 810169 31](#_Toc161854162)

[Slika 16: sučelje za vizualnu analizu Blazhko zvijezde 1736308 32](#_Toc161854163)

[Slika 17: sučelje za vizualnu analizu Blazhko zvijezde 10420063 34](#_Toc161854164)

## 5.2 Popis grafova

[Graf 1: raspodjela mjerenja kod LINEAR RR Lira 26](#_Toc161854165)

[Graf 2: raspodjela mjerenja kod ZTF RR Lira 27](#_Toc161854166)

[Graf 3: distribucija vremenskih perioda LINEAR zvijezda 27](#_Toc161854167)

[Graf 4: distribucija vremenskih perioda ZTF zvijezda po filterima 28](#_Toc161854168)

[Graf 5: prostorna distribucija RR Lira po galaksiji 28](#_Toc161854169)

[Graf 6: distribucija LINEAR i ZTF Blazhko perioda 35](#_Toc161854170)

[Graf 7: LINEAR i ZTF razlike glavnih i Blazhko frekvencija 36](#_Toc161854171)

[Graf 8: točkasti dijagram LINEAR i ZTF vrijednosti hi-kvadrata 37](#_Toc161854172)

[Graf 9: udio RRab i RRc tipova Blazhko zvijezda 38](#_Toc161854173)

[Graf 10: pozicije Blazhko zvijezda pomoću galaktičkih koordinata 38](#_Toc161854174)

## 5.3 Popis tablica

[Tablica 1: Metapodaci LINEAR zvijezda koji imaju klasifikaciju tipa zvijezde 15](#_Toc161854175)

[Tablica 2: 10 Blazhko zvijezda i njihovi osnovni podaci 29](#_Toc161854176)

# 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'  turqoise = '#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\_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, lsPS=True, 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 alows 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 calculation, please wait...')  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  **if** verbose:print('Frequency and power have been calculated.')  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  **if** verbose:print('The best period is, ',best\_period)  **if** lsPS:  **return** best\_period, frequency, power  **else**:  **return** best\_period  **except**:  # if there is no data, assign everything to 0 or empty  **if** verbose: print('Period calculation unsuccesful.')  best\_period = 0.0  frequency = np.array(())  power = np.array(())  **return** best\_period, frequency, power  **def** **LINEARLS**(LINEARlightcurves, Lid, 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 = LINEARlightcurves.T  ### now compute periods (using LombScargle from astropy.timeseries)  nterms = 3  # LINEAR-only period  **if** verbose:  print(' computing LINEAR period...')  Plinear, fL, pL = doPeriods(tL, mL, mLerr, nterms, Lid, lsPS=**True**)  **if** verbose:  print(' LINEAR period = ', Plinear)  **return** Plinear, fL, pL, tL, mL, mLerr  **def** **ZTFs**(ZTFdata, Lid, lsPS=True, 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): dataframe 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']  nterms = 3  **if** verbose:  print('And now for the ZTF counterpart -------------')  **if** ZTFdata.empty == **True**:  ZTFbestPeriod, ZTFbestfreq, Zbestpow = 0, 0, 0  Zfreq, Zpow = np.array(()), np.array(())  ZTFperiod\_ograms.append((ZTFbestPeriod, Zfreq, Zpow))  timeZ,magZ,magErrZ = np.array(()), np.array(()), np.array(())  **else**:  **if** verbose:  print(' computing ZTF period...')  **for** b **in** ZTFbands:  BandData = ZTFdata.loc[ZTFdata['filtercode'] == b]  timeZ = BandData['mjd']  magZ = BandData['mag']  magErrZ = BandData['magerr']  ZTFperiod, Zfreq, Zpow = doPeriods(timeZ, magZ, magErrZ, nterms, Lid, lsPS=lsPS)  ZTFperiod\_ograms.append((ZTFperiod, Zfreq, Zpow))    ZTFperiod\_ograms.sort(key=**lambda** x: x[0], reverse=**True**)  **if** len(ZTFperiod\_ograms) < 3:  ZTFbestPeriod, ZTFbestfreq, Zbestpow = ZTFperiod\_ograms[0]  **else**:  ZTFbestPeriod, ZTFbestfreq, Zbestpow = ZTFperiod\_ograms[1]  **if** verbose:  print(' ZTF period = ', ZTFbestPeriod)  **return** ZTFbestPeriod, ZTFbestfreq, Zbestpow, Zfreq, Zpow, 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['rms'] = sigG(delmag)  LCanalysisResults['chi'] = delmag/LCanalysisResults['dataTemplateErr']  LCanalysisResults['chi2dof'] = np.sum(LCanalysisResults['chi']\*\*2)/np.size(LCanalysisResults['chi'])  LCanalysisResults['chi2dofR'] = sigG(LCanalysisResults['chi'])  **return** LCanalysisResults  **def** **RR\_lyrae\_analysis**(end, i, Lid, dataL, dataZ, lc\_analysis, ZTF\_data\_best, fits, periodograms, verbose=False):  '''  This function analyzes RR Lyrae light curve data by calculating periods, fitting light curves and conducting BE  candidate analysis of local peaks.  Arguments:  end(str) = how to save this iteration of the dataset  i(int) = iterable  Lids(list) = list of LINEAR ids  ztfdata(dict) = dictionary of ZTF data  lc\_analysis(dict) = dictionary to save light curve analysis  ZTF\_data\_lc(list) = place to save best ztf data  fits(list) = list to save light curve fits  periodograms(list) = list to save periodograms  '''    # accessing data    **if** verbose:  print('Current i:',i)  print('Current LINEAR ID:', Lid)  print(f'Shape of linear:{dataL.shape}, shape of ztf:{dataZ.shape}')  # calculating the periods  Plinear, fL, pL, tL, mL, meL = LINEARLS(dataL, Lid)  Pztf, Zbestf, Zbestp, fZ, pZ, tZ, mZ, meZ = ZTFs(dataZ, Lid)  **if** verbose:  print(f'Plinear: {Plinear}, ZTFperiod: {Pztf}')  # saving the ZTF data  ZTF\_data\_best.append((Lid, (tZ, mZ, meZ)))  # blazhko periodogram analysis  fFoldedL, pFoldedL, fMainPeakL, fBlazhkoPeakL, BlazhkoPeriodL, BpowerRatioL, BsignificanceL = getBlazhkoPeak(fL, pL)  **if** fZ.size==0 **or** pZ.size==0 **or** Plinear==0 **or** Pztf==0 **or** dataZ.shape[0]==0:  fFoldedZ, pFoldedZ, fMainPeakZ, fBlazhkoPeakZ, BlazhkoPeriodZ, BpowerRatioZ, BsignificanceZ = np.array(()), np.array(()), 0, 0, 0, 0, 0  **else**:  fFoldedZ, pFoldedZ, fMainPeakZ, fBlazhkoPeakZ, BlazhkoPeriodZ, BpowerRatioZ, BsignificanceZ = getBlazhkoPeak(fZ, pZ)  # period analysis  Plinear = round(Plinear, 6)  Pztf = round(Pztf, 6)  Pmean = round((Plinear+Pztf)/2, 6)  Pratio = round((Pztf/Plinear), 6)  # saving the periodograms  periodograms.append((Lid, (fL, pL, fFoldedL, pFoldedL), (fZ, pZ, fFoldedZ, pFoldedZ)))  # fitting the light curves  ntermsModels = 6  **if** verbose: print('Starting to fit light curves!')  **if** tZ.size==0 **or** mZ.size == 0 **or** meZ.size == 0 **or** Plinear == 0.0 **or** Pztf == 0.0:  **if** verbose: print(f'We engaged with these parameters: tZ={tZ.size}, mZ={mZ.size}, meZ={meZ.size}, plinear={Plinear}, pztf={Pztf}')  LINEAR\_Plinear = {  'modelPhaseGrid': np.array(()),  'modelFit': np.array(()),  'dataPhasedTime': np.array(()),  'A': 0.0,  'mmax': 0.0,  'modTemplate': np.array(()),  'dataTemplate': np.array(()),  'dataTemplateErr': np.array(()),  'modelFit2data': np.array(()),  'rms': 0.0,  'chi': 0.0,  'chi2dof': 0.0,  'chi2dofR': 0.0  }  LINEAR\_Pmean = {  'modelPhaseGrid': np.array(()),  'modelFit': np.array(()),  'dataPhasedTime': np.array(()),  'A': 0.0,  'mmax': 0.0,  'modTemplate': np.array(()),  'dataTemplate': np.array(()),  'dataTemplateErr': np.array(()),  'modelFit2data': np.array(()),  'rms': 0.0,  'chi': 0.0,  'chi2dof': 0.0,  'chi2dofR': 0.0  }  ZTF\_Pztf = {  'modelPhaseGrid': np.array(()),  'modelFit': np.array(()),  'dataPhasedTime': np.array(()),  'A': 0.0,  'mmax': 0.0,  'modTemplate': np.array(()),  'dataTemplate': np.array(()),  'dataTemplateErr': np.array(()),  'modelFit2data': np.array(()),  'rms': 0.0,  'chi': 0.0,  'chi2dof': 0.0,  'chi2dofR': 0.0  }  ZTF\_Pmean = {  'modelPhaseGrid': np.array(()),  'modelFit': np.array(()),  'dataPhasedTime': np.array(()),  'A': 0.0,  'mmax': 0.0,  'modTemplate': np.array(()),  'dataTemplate': np.array(()),  'dataTemplateErr': np.array(()),  'modelFit2data': np.array(()),  'rms': 0.0,  'chi': 0.0,  'chi2dof': 0.0,  'chi2dofR': 0.0  }  **else**:  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)  STAR = [Plinear, Pztf, Pmean, Pratio, np.size(tL), LINEAR\_Plinear['rms'], round(LINEAR\_Plinear['chi2dof'], 1), round(LINEAR\_Plinear['chi2dofR'], 1),LINEAR\_Pmean['rms'],  round(LINEAR\_Pmean['chi2dof'],1), round(LINEAR\_Pmean['chi2dofR'],1), round(LINEAR\_Plinear['mmax'],2), round(LINEAR\_Plinear['A'],2),  np.size(tZ), ZTF\_Pztf['rms'], round(ZTF\_Pztf['chi2dof'],1), round(ZTF\_Pztf['chi2dofR'],1), ZTF\_Pmean['rms'], 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)))  **return** lc\_analysis, periodograms, fits, ZTF\_data\_best  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 = []  **if** os.path.isfile('../outputs/LC\_ANALYSIS'+end+'.csv'):  print('Loading data!')  LC\_ANALYSIS = pd.read\_csv('../outputs/LC\_ANALYSIS\_'+end+'.csv')  periodogr = pickle.load(open("../outputs/periodograms\_"+end+".pkl", "rb"))  fits = np.load("../outputs/fits\_"+end+".npy", allow\_pickle=**True**)  ztfdata = np.load("../outputs/ztf\_data\_filter"+end+".npy", allow\_pickle=**True**)  **else**:  **for** i **in** tqdm(range(test)):  Lid = Lids[i]  datalinear = dataL.get\_light\_curve(Lid)  **for** j **in** dataZTF:  **if** j[0]==Lid:  dataZ = j[1]    **if** datalinear.T[0].shape[0]>250:  lc\_analysis, periodograms, fits, ZTF\_data\_best = RR\_lyrae\_params.RR\_lyrae\_analysis(end, i, Lid, datalinear, dataZ, lc\_analysis, ZTF\_data\_best, fits, periodograms)  **else**:  **continue**    **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**)    # 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 inputing good candidates  '''  **for** i **in** range(df.shape[0]):    # STEP 1: getting rid of trash  # ---------  **if** df['Ampl\_diff'][i]<2:  #if df["dP"][i]<1:  **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 **or** df['NdataZTF'][i]>250:  # 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]-0.5)>dPmin)&(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**:  # STEP 3: determine scorechart for other parameters  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 > 4e-5 **and** period < 0.001: p\_score += 2  **if** period > 0.001: p\_score += 4    # CHI  **if** (chiZ>=2.5 **and** chiZ<=4.5)**and**(chiL >= 2.5 **and** chiL <= 4.5):  chi\_score += 4  df.loc[i, 'ChiType'] = 'LZ'  **if** (chiL>4.5)**and**(chiZ>4.5):  chi\_score += 6  df.loc[i, 'ChiType'] = 'LZ'  **if** (chiL >=2.5 **and** chiL <= 4.5):  chi\_score += 2  df.loc[i, 'ChiType'] = 'L'  **if** (chiZ>=2.5 **and** chiZ<=4.5):  chi\_score += 2  df.loc[i, 'ChiType'] = 'Z'  **if** chiL>4.5:  chi\_score += 3  df.loc[i, 'ChiType'] = 'L'  **if** chiZ>4.5:  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>5:  row = pd.DataFrame(df.iloc[[int(i)]])  dfnew = pd.concat([dfnew, row.reset\_index(drop=**True**)], ignore\_index=**True**, axis=0)  #else:  #pass  **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]  tZ = self.Zdata[n][1][0].to\_numpy()  #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, capsize=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, capsize=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['Zampl'][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\_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)  # 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  f1yr = flin+1/365.0  ax.plot([f1yr, f1yr], [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=turqoise)  ax.set\_xlim(flin, 1.01\*fBlazhkoPeakL)  **else**:  ax.plot([flin+0.5\*(fBlazhkoPeakL-flin), flin\*fac], [noise5sig, noise5sig], lw = 1, ls='--', c=turqoise)  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  f1yrZ = fztf+1/365.0  ax.plot([f1yrZ, f1yrZ], [0,0.7\*np.max(pFoldedZ)], lw = 1, ls='-.', c=green)  f1yrZ = fztf-1/365.0  ax.plot([f1yrZ, f1yrZ], [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  f1yrZ = fztf+1/365.0  ax.plot([f1yrZ, f1yrZ], [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=turqoise)  ax.set\_xlim(flin, 1.01\*fBlazhkoPeakZ)  **else**:  ax.plot([flin+0.5\*(fBlazhkoPeakZ-fztf), fztf\*fac], [noise5sigZ, noise5sigZ], lw = 1, ls='--', c=turqoise)  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\_ylim(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][0], ztf\_data[order][1][1], ztf\_data[order][1][2]  ax[1].set\_ylim(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, capsize=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, capsize=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**)  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\_ztf\_per = category\_analysis(blazhko\_can, fits, periodogr, ztf\_data, dataLINEAR, 'ztf\_periodogram',id\_list=**None**,parameter='IndicatorType', value='Z')  blazhko\_analyzer = BE\_candidates\_ztf\_per.get\_save\_data()  end = 'ztf\_periodogram'  blazhko\_analyzer.to\_csv("../outputs/blazhko\_list"+end+".csv", index=**False**)  end = 'ztf\_periodogram'  blazhko\_can\_ztf = pd.read\_csv('../outputs/blazhko\_list'+end+'.csv')  BE\_candidates\_linear\_per = category\_analysis(blazhko\_can, fits, periodogr, ztf\_data, dataLINEAR, 'linear\_periodogram',id\_list=**None**,parameter='IndicatorType', value='L')  blazhko\_analyzer = BE\_candidates\_linear\_per.get\_save\_data()  end = 'linear\_periodogram'  blazhko\_analyzer.to\_csv("../outputs/blazhko\_list"+end+".csv", index=**False**)  end = 'linear\_periodogram'  blazhko\_can\_linearper = pd.read\_csv('../outputs/blazhko\_list'+end+'.csv')  BE\_candidates\_ztf\_chi = category\_analysis(blazhko\_can, fits, periodogr, ztf\_data, dataLINEAR, 'ztf\_chi',id\_list=**None**,parameter='ChiType', value='Z')  blazhko\_analyzer = BE\_candidates\_ztf\_chi.get\_save\_data()  end = 'ztf\_chi'  blazhko\_analyzer.to\_csv("../outputs/blazhko\_list"+end+".csv", index=**False**)  end = 'ztf\_chi'  blazhko\_can\_ztfchi = pd.read\_csv('../outputs/blazhko\_list'+end+'.csv')  BE\_candidates\_linear\_chi = category\_analysis(blazhko\_can, fits, periodogr, ztf\_data, dataLINEAR, 'linear\_chi',id\_list=**None**,parameter='ChiType', value='L')  blazhko\_analyzer = BE\_candidates\_linear\_chi.get\_save\_data()  end = 'linear\_chi'  blazhko\_analyzer.to\_csv("../outputs/blazhko\_list"+end+".csv", index=**False**)  end = 'linear\_chi'  blazhko\_canlinearchi = pd.read\_csv('../outputs/blazhko\_list'+end+'.csv')  BE\_candidates\_other = category\_analysis(be\_other, fits, periodogr, ztf\_data, dataLINEAR, 'other')  blazhko\_analyzer = BE\_candidates\_other.get\_save\_data()  end = 'other'  blazhko\_analyzer.to\_csv("../outputs/blazhko\_list"+end+".csv", index=**False**)  end = 'other'  blazhko\_can\_other = pd.read\_csv('../outputs/blazhko\_list'+end+'.csv')  same\_id\_list = blazhko\_can['LINEAR id'].isin(same\_id)  SAME\_BE\_LIST = blazhko\_can[same\_id\_list]  SAME\_BE\_LIST = SAME\_BE\_LIST.reset\_index(drop=**True**)  SAME\_BE\_LIST['Quality flag'] = len(same\_id)\*[3]  SAME\_BE\_LIST.head()  EI\_list1 = [x **for** x **in** EI\_list **if** x **not** **in** same\_id]  EI\_be\_list = blazhko\_can['LINEAR id'].isin(EI\_list1)  EI\_BE\_list = blazhko\_can[EI\_be\_list]  EI\_BE\_list = EI\_BE\_list.reset\_index(drop=**True**)  EI\_BE\_list['Quality flag'] = len(EI\_list1)\*[2]  EI\_BE\_list.head()  ZI\_list1 = [x **for** x **in** ZI\_list **if** x **not** **in** same\_id]  len(ZI\_list1)  ZI\_list1 = [x **for** x **in** ZI\_list1 **if** x **in** list(lc\_analysis['LINEAR id'])]  ZI\_be = lc\_analysis['LINEAR id'].isin(ZI\_list1)  ZI\_be\_list = lc\_analysis[ZI\_be]  ZI\_be\_list = ZI\_be\_list.reset\_index(drop=**True**)  ZI\_be\_list['Quality flag'] = len(ZI\_list1)\*[1]  print(ZI\_be\_list.shape)  ZI\_be\_list.head()  FINAL\_BE\_CATALOGUE = pd.DataFrame()  FINAL\_BE\_CATALOGUE = pd.concat([FINAL\_BE\_CATALOGUE, SAME\_BE\_LIST.reset\_index(drop=**True**)],ignore\_index=**True**, axis=0)  FINAL\_BE\_CATALOGUE = pd.concat([FINAL\_BE\_CATALOGUE, EI\_BE\_list.reset\_index(drop=**True**)],ignore\_index=**True**, axis=0)  FINAL\_BE\_CATALOGUE = pd.concat([FINAL\_BE\_CATALOGUE, ZI\_be\_list.reset\_index(drop=**True**)],ignore\_index=**True**, axis=0)  print(FINAL\_BE\_CATALOGUE.shape)  FINAL\_BE\_CATALOGUE.head()  FINAL\_BE\_CATALOGUE.to\_csv('../outputs/FINAL\_BE\_CATALOGUE.csv', index=**False**) |

1. AstroML: https://www.astroml.org [↑](#footnote-ref-1)
2. IPAC: https://www.ipac.caltech.edu [↑](#footnote-ref-2)
3. IRSA: https://irsa.ipac.caltech.edu/frontpage/ [↑](#footnote-ref-3)