

Histogram

Tarakčija, Dinko

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:337481>

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

Download date / Datum preuzimanja: **2024-10-03**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

HISTOGRAM

Diplomski rad

Dinko Tarakčija

Osijek, 2020.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 21.09.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Dinko Tarakčija
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 808 R, 23.10.2019.
OIB studenta:	21801436559
Mentor:	izv. prof. dr.sc. Josip Job
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Časlav Livada
Član Povjerenstva 1:	izv. prof. dr.sc. Josip Job
Član Povjerenstva 2:	Izv.prof.dr.sc. Ratko Grbić
Naslov diplomskog rada:	Histogram
Znanstvena grana rada:	Obradba informacija (zn. polje računarstvo)
Zadatak diplomskog rada:	Opisati postupak izrade histograma i implementirati programsko rješenje za njegovu izradu, a primjenu demonstrirati na odgovarajućim primjerima.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	21.09.2020.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 25.09.2020.

Ime i prezime studenta:	Dinko Tarakčija
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 808 R, 23.10.2019.
Turnitin podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom: **Histogram**

izrađen pod vodstvom mentora izv. prof. dr.sc. Josip Job

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD	1
2. PREGLED PODRUČJA RADA	2
3. HISTOGRAM I NJEGOVA UPOTREBA	4
3.1. Vrste histograma.....	7
4. ANALIZA PODATAKA POMOĆU HISTOGRAMA.....	16
4.1. Glazbeno - teoretska podloga	16
4.2. Fourierova transformacija	20
4.3. Prikupljanje podataka pomoću Pythona	24
4.4. Prikaz prikupljenih podataka u obliku histograma	30
5. ZAKLJUČAK	40
LITERATURA.....	41
SAŽETAK.....	43
ABSTRACT	44
ŽIVOTOPIS	45
PRILOZI.....	46

1. UVOD

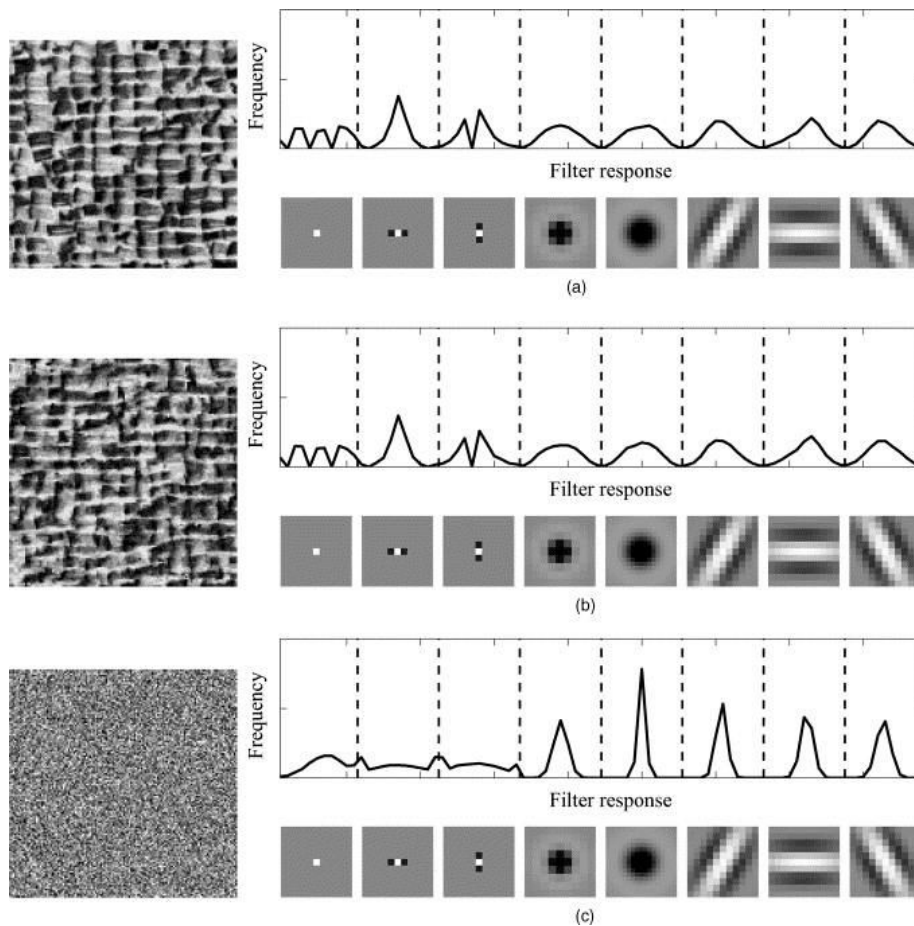
Histogram pripada grani statistike i grafički prikazuje zadane podatke u obliku pravokutnika ili košarica (eng. *bins*, *buckets*). U radu će se prikazati razne vrste histograma, njihove karakteristike te matematička podloga iza njih. Pozornost će se usmjeriti na stvaranju histograma iz odsviranih tonova na električnoj gitari, odnosno, koliko točno računalo može prepoznati odsvirane tonove na gitari i prikazati ih u obliku histograma.

U ovom radu bit će opisana teorija histograma, njegove glavne karakteristike i klasifikacija vrsta. Prikazat će se osnove glazbene teorije, frekvencije žica i Fourierove transformacije. Pojasnit će se kako se podaci uz pomoć Python programskog jezika prikupljaju u računalo te na šest primjera napraviti histograme, izvršiti analizu i razlike između njih. Pažnja će se usmjeriti na karakteristike histograma ovisno o točnosti programa i glazbenoj podlozi primjera.

U prvom poglavlju bit će opisani histogrami, za kakve podatke su korišteni te prednosti i nedostaci histograma. Nakon toga, prikazat će se neke vrste histograma i za svaku od navedenih vrsta histograma matematički će se obraditi. Za prikazivanje različitih vrsta histograma bit će korištene Googleove skripte. U drugom poglavlju bit će objašnjena osnova glazbene teorije, u kojoj će naglasak biti na povišenim i sniženim tonovima. Pojasnit će se kako se osnove glazbene teorije prenose na električnu gitaru. Nakon toga bit će opisana analiza frekvencija žice gitare, Fourierovi redovi i Fourierove transformacije (diskretna Fourierova transformacija i brza Fourierova transformacija), frekvencijska analiza zvuka pomoću Fourierove transformacije i Hannovi prozori. Slijedi pojašnjenje kako se s Pythonom prikupljaju podaci u računalo. Zadnje potpoglavlje rada prikazuje šest različitih histograma koji su dobiveni provođenjem Python skripte na šest različitih primjera u .wav formatu te se uspoređuju njihovi oblici. Prva četiri primjera prikazuju kako računalo prepoznaje podatke kad se unose zasebno, a peti i šesti pokazuju unošenje više podataka u isto vrijeme. Prikazano je kako računalo izbacuje histograme iz kojih se lakše određuju primjeri ako se podaci unose zasebno, a ne u isto vrijeme.

2. PREGLED PODRUČJA RADA

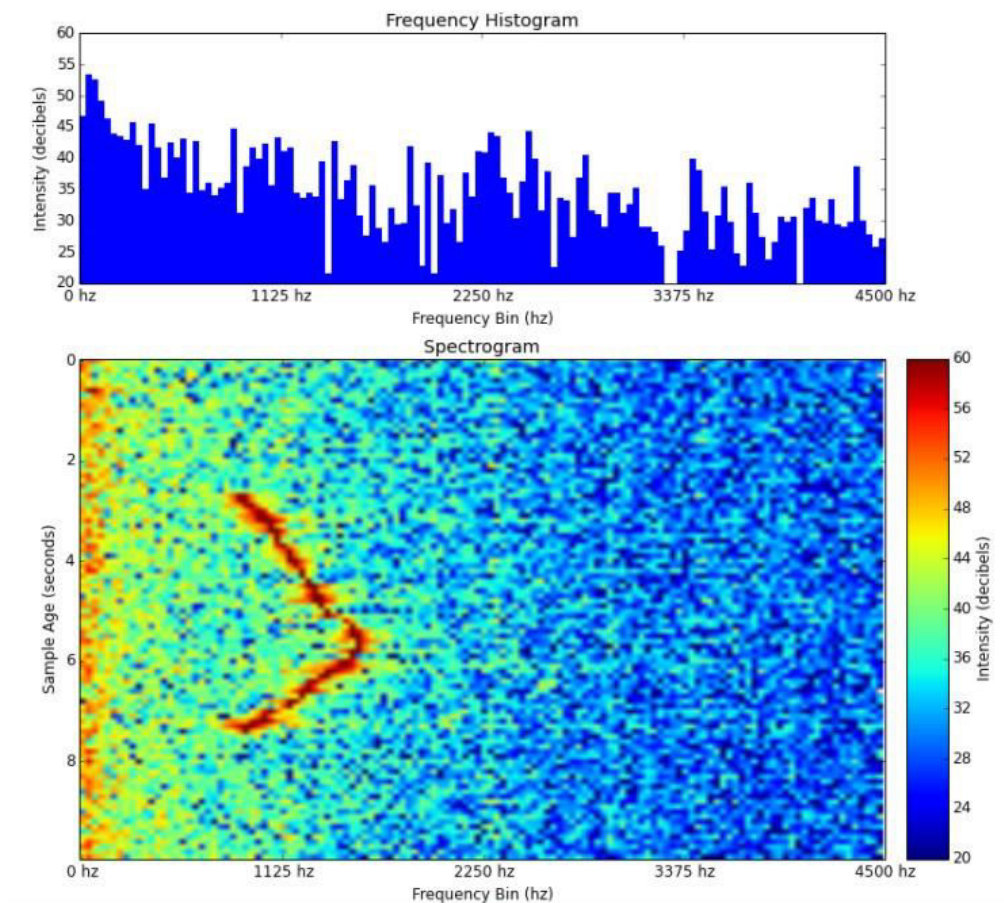
Histogrami ne moraju nužno biti korišteni samo u glazbi kao što su korišteni u ovom radu, korišteni su za sve vrste podataka te ima raznih vrsta histograma. Neki su kompliciraniji i korišteni su u, primjerice, obradi slika. Jedan od problema koji je obrađen spektralnim histogramima je klasifikacija tekstura.



Sl. 2.1. Spektralni histogram u analizi slika [1]

Moguće je prikazivati Fourierovu transformaciju kao spektrograme¹ koji se dobiju iz histograma te na taj način analizirati tonove i glasnoću kojom su odsvirani. [2]

¹ Vizualni način prikaza jačine signala, tj. glasnoće



Sl. 2.2. Spektrogram frekvencijskog histograma

Ako se uzme u obzir analiza akorda [3], računalo je sposobno na osnovu analize par tonova izračunati o kojim se akordima radi, no potrebno je poznavati glazbenu teoriju i zadati računalu pravila po kojima će znati kako odrediti koji su akordi u pitanju, tj. kako zanemariti šumove. Također, kad bi računalo moglo s određenom točnošću pokazivati koje se note sviraju, postoji opcija u kojoj bi se automatski mogli stvarati notni zapisi ili tablature². [4]

Histogram prikazuje frekvencijsku distribuciju što bi, ako bi postojala aplikacija koja prepoznaje tonove i izbacuje vizualnu reprezentaciju odsviranih tonova, bilo korisno glazbenicima za vježbanje glazbene teorije (određivanje tonaliteta, provjera točnosti skale i slično).

² Notni zapis za gitaru gdje se brojevima označava koji prst ide na koje polje određene žice gitare

3. HISTOGRAM I NJEGOVA UPOTREBA

Histogram označava prikaz frekvencijske raspodjele preko pravokutnika čije širine prikazuju razredne intervale i čije su površine proporcionalne odgovarajućim frekvencijama. Izgledom je sličan stupčastim dijagramima, ali se razlikuje u načinu grupiranja podataka. U histogramu se podaci grupiraju u košarice (eng. *bins, buckets*) koje predstavljaju učestalost pojavljivanja podataka prema prethodno zadanim granicama.

Izraz „histogram“ potekao je britanskog matematičara i statističara Karla Pearsona koji je histogram definirao kao jedan od načina grafičkog prikaza podataka iako, povijesno gledajući, histogrami su se koristili i prije nego su dobili ime. Budući da slične stupčastim dijagramima i činjenica kako se prvi stupčasti dijagram pojavio u knjizi „*The Commercial and Political Atlas*“ Williama Playfaira 1786. godine, može se reći kako su histogrami nastali po uzoru na stupčaste dijagrame.

Neka se uzme relacija R s n brojevanih oznaka X_i ($i = 1 \dots n$). Vrijednost V_i oznake X_i je skup vrijednosti X_i koji se nalaze u R .

Neka je $V_i = \{ v_i(k) : 1 \leq k \leq D_i \}$, gdje je $v_i(k) < v_i(k+1)$ kada je $k < j$.

Širina $s_i(k)$ od $v_i(k)$ se definira kao $s_i(k) = v_i(k+1) - v_i(k)$, za $1 \leq k < D_i$; uzima se kako je $s_i(D_i) = 1$.

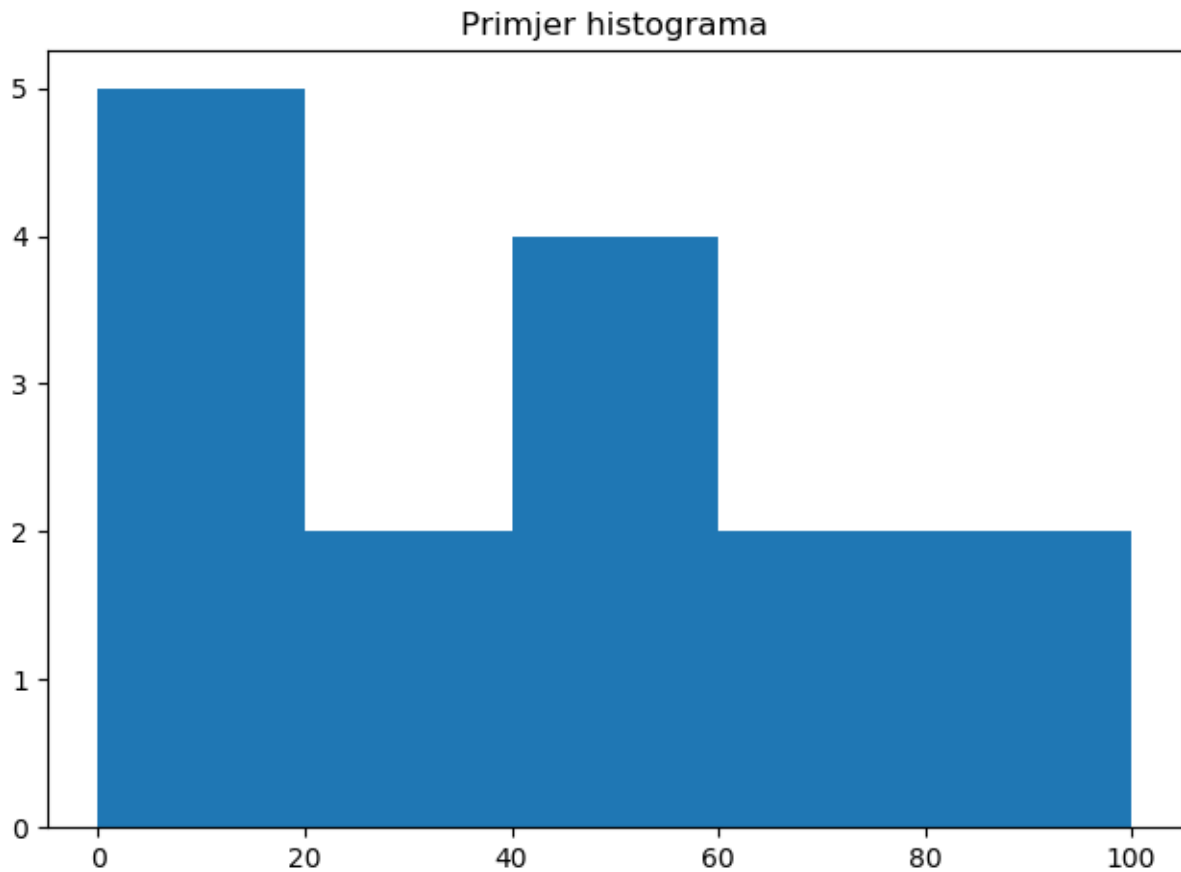
Učestalost ili frekvencija $f_i(k)$ od $v_i(k)$ je n -torka u R sa $X_i = v_i(k)$.

Površina $a_i(k)$ od $v_i(k)$ se definira kao $a_i(k) = f_i(k) \times s_i(k)$.

Raspodjela podataka X_i je set parova $T_i \{ (v_i(1), f_i(1)), (v_i(2), f_i(2)), \dots, (v_i(D_i), f_i(D_i)) \}$.

Učestalost pojavljivanja zajedničkih podataka (eng. *joint-frequency*) $f_i(k_1, \dots, k_n)$ vrijednosti kombinacije $\langle v_i(k_1), \dots, v_n(k_n) \rangle$ je n -torka u R koja sadrži $v_i(k_i)$ u brojčanoj oznaci X_i za svaki i .

Iz toga slijedi kako je raspodjela $T_{1, \dots, n}$ od $X_{1, \dots, n}$ cijeli set parova vrijednosti kombinacija i učestalosti pojavljivanja istih. [5]

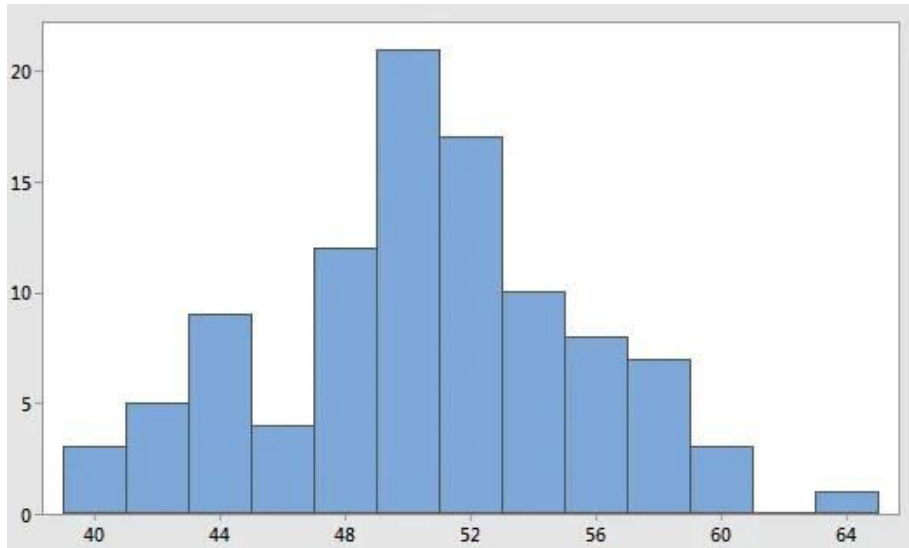


Sl. 3.1. Primjer histograma sa zadanim podacima

Na slici 3.1. prikazan je primjer histograma napravljen u Python programskom jeziku i njegovim bibliotekama Matplotlib, Numpy i Pyplot. Podaci iz kojih je histogram napravljen je niz od petnaest nasumično odabranih brojeva između 1 i 100:

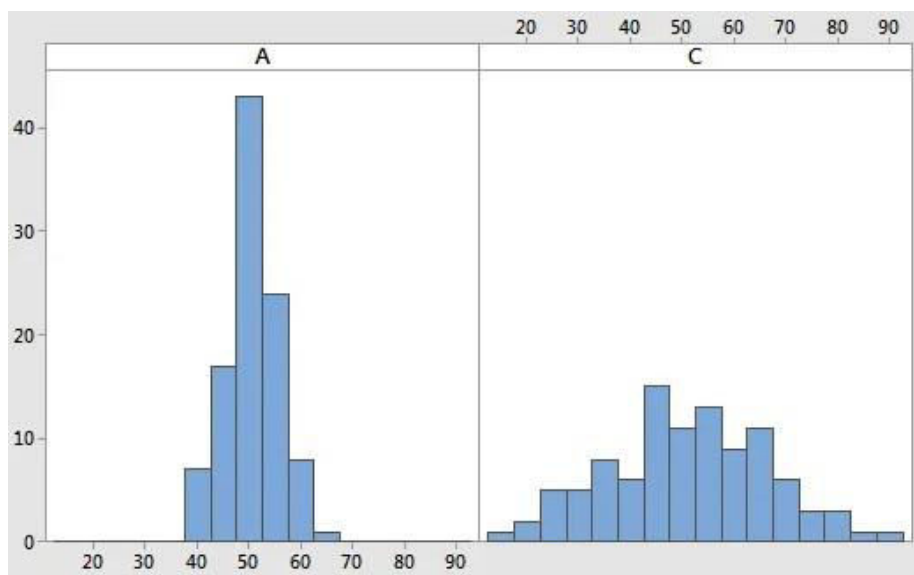
11, 77, 7, 53, 46, 83, 45, 64, 10, 30, 41, 15, 89, 19, 37

S obzirom da histogram prikazuje frekvencijsku raspodjelu podataka, histogramima je nekad lakše u većem setu podataka odrediti aritmetičku sredinu nego stupčastim dijagramima. [6]



Sl. 3.2. Histogram s ravnomjerno raspoređenim podacima

Na slici 3.2., raspoređenost podataka je otprilike simetrična, a najviše podataka nalazi se oko košarice vrijednosti 50, dok se vrijednosti košarica lijevo i desno smanjuju.



Sl. 3.3. Dva histograma s sličnim aritmetičkim sredinama

Na slici 3.3., oba histograma imaju slične aritmetičke sredine, ali su podatci različito raspoređeni. U prvom histogramu, većina podataka leži između 40 i 60, dok su u drugom raspoređeni kroz cijeli histogram. Bez obzira što im je aritmetička sredina možda slična, postoji razlika u izgledu histograma.

Histogrami pokazuju koje su vrijednosti učestalije, a koje se manje pojavljuju s obzirom na njihovu raspršenost, što se ne može shvatiti iz same liste vrijednosti. Aritmetička sredina može pomoći, ali svejedno treba promotriti same histograme kako bi se utvrdilo o kakvim distribucijama podataka se radi. Također, histogrami su korisni za izdvajanje podataka koji se ne pojavljuju često u nizu podataka. Oni se pojavljuju kao izdvojena košarica u bilo kojem dijelu histograma.

Iako je moguće stavljati više histograma na jedan graf kako bi se dobilo bolje saznanje o vrijednostima za određene grupe podataka, dodavanjem više grupa na jedan graf gdje je svaka grupa jedan histogram, graf se može pretrpati tako da se ne mogu donijeti zaključci o distribucijama podataka. Obično je najbolje iskoristiti histograme za jednu ili dvije grupe podataka odjednom.

Nedostatak histograma su činjenica da su subjektivni [7] – širina košarica je stvar osobnog izbora što ima veliki utjecaj na izgled histograma. Ako se ne uzmu dobre granice za košarice, moguće je dramatično promijeniti izgled histograma, pogotovo kad se radi o maloj grupi podataka za histogram. Također, histogrami dosta ovise o temi koju se analizira, radi li se istraživanje u kojem se uspoređuje više kategorija, nije ih moguće staviti na jedan graf pa analizirati. Pojavi li se za histogram previše ili premalo podataka, postanu teški za analizirati jer previše podataka može značiti težu analizu, a premalo podataka može značiti izostavljanje bitnih podataka.

3.1. Vrste histograma

Postoje razne vrste histograma, za potrebe prikaza i objašnjavanja raznih distribucija podataka, koristit će se Google Developers Chart [8] biblioteka, gdje korisnik sam može ubaciti podatke i lokalno na vlastitom računaru dobiti rezultat

Primjer: odigran je jedan šahovski turnir. Na turniru je bilo 8 igrača, šahovske partije su bile raspoređene po Bergerovom sustavu [9] u kojem svaki igrač igra protiv svakog igrača i bilo je 28 partija (jer igrač ne može igrati sam protiv sebe). Želi se histogramom prikazati trajanje šahovske partije u potezima svih partija na turniru.

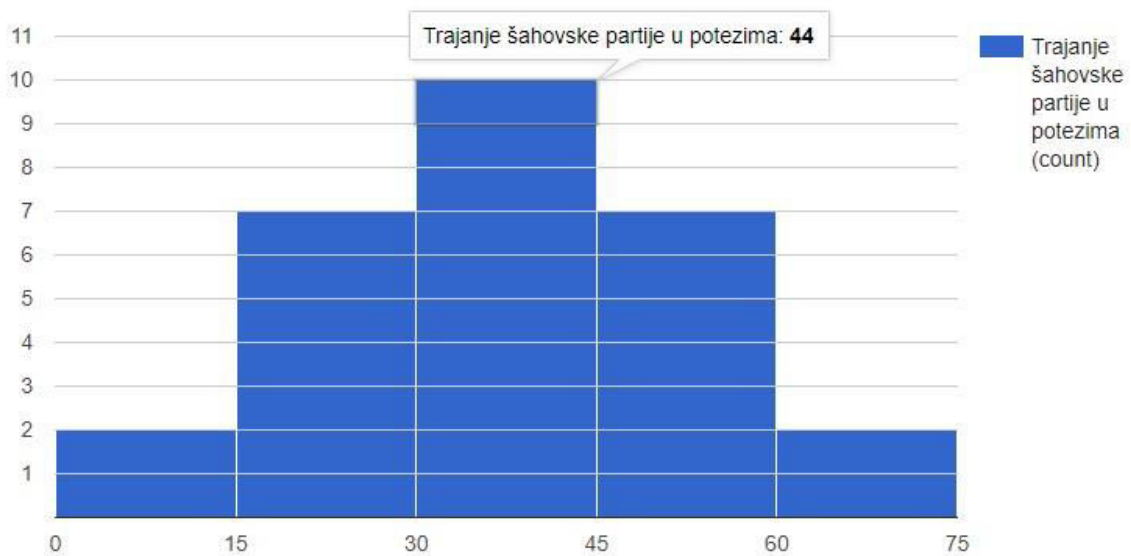
Neka idući niz predstavlja broj poteza u partiji: 31, 13, 67, 53, 74, 9, 38, 39, 24, 16, 47, 50, 44, 32, 19, 32, 48, 20, 29, 58, 53, 36, 40, 57, 43, 25, 24, 40.

Slijedeća tablica prikazuje podatke o broju poteza šahovskih partija raspoređene prema veličini prema košaricama.

Tab. 3.1. Broj poteza u partiji i frekvencija pojavljivanja

Veličina košarice	Broj poteza u partiji	Količina partija u košarici
0 – 15	9, 13	2
16 – 30	16, 19, 20, 24, 24, 25, 29	7
31 – 45	31, 32, 32, 36, 38, 39, 40, 40, 43, 44	10
46 – 60	47, 48, 50, 53, 53, 57, 58	7
61 – 75	67, 74	2

Kad se napravi histogram pomoću Googleove skripte, dobije se slijedeći prikaz:



Sl. 3.4. Histogram broja poteza u šahovskim partijama

Na slici 3.4. se vidi tzv. zvonasti histogram koji obično predstavlja normalnu razdiobu. Prvi i najjednostavniji korak provjere predstavlja li histogram normalnu razdiobu ili ne bio bi provjera aritmetičke sredine, moda i medijana.

Aritmetička sredina [10]:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad \#(3-1)$$

gdje je:

- \bar{x} – aritmetička sredina partija
- n – broj promatranih partija
- x_i – broj poteza u jednoj partiji

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{28} \sum_{i=1}^{28} x_{28} = \frac{x_1 + x_2 + x_3 + \dots + x_{28}}{28} = \frac{9 + 13 + 16 + \dots + 74}{28} = 37.89 \quad \#(3-2)$$

Medijan je središnji broj u nizu brojeva koji su poredani od najmanjeg prema najvećem pod uvjetom da niz ima neparan broj brojeva. Ako niz ima paran broj brojeva, tada se traži aritmetička sredina između dva središnja broja u poredanom nizu. Kada je niz brojeva neparan, ali velik i ne može se odrediti koji je član niza medijan, tada može poslužiti iduća formula³:

$$\text{medijan} = \frac{(n+1)}{2} \quad \#(3-3)$$

gdje je:

- n – broj člana u nizu

Ako je broj veći, za računanje medijana parnog broja niza može se promotriti na idući način:

$$\text{medijan} = \frac{n}{2} - 1 \quad \#(3-4)$$

gdje je:

- n – sveukupan broj članova u nizu

Prema (3-3), dobije se koliko članova treba biti ispod i iznad srednja 2 člana u nizu.

³ <http://www.statisticshowto.com/probability-and-statistics/statistics-definitions/median-formula/>

Iz tablice 3.1. i činjenici da je bilo 28 partija, može se odrediti da je niz parnog broja i da se medijan računa na idući način:

$$\text{medijan} = \frac{28}{2} - 1 = 13 \#(3 - 5)$$

Odnosno, rezultat gore znači da su 14. i 15. član medijani u zadanom nizu, a to su partija s 38 i partija 39 poteza. Uzme li se aritmetička sredina te dvije partije, dobije se slijedeći rezultat, što je u ovom slučaju zapravo medijan:

$$x = \frac{1}{2} \sum_{i=1}^2 x_2 = \frac{38 + 39}{2} = 38.5 \#(3 - 6)$$

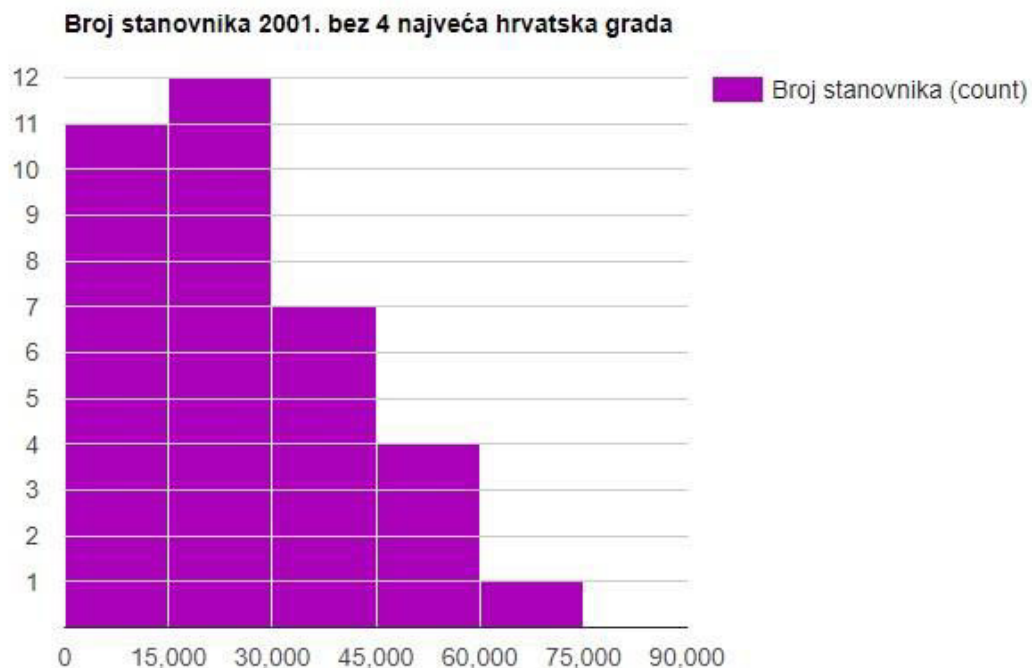
Posljednje što ostaje za odrediti je mod. Mod je jednak onome broju koji se najčešće pojavljuje u nizu. Unutar jednog niza može postojati više modova.

Za podatke iz tablice 3.1., vidljivo je kako postoji 4 moda: partije koje imaju 24, 32, 40 i 53 poteza (zato što se jedine pojavljuju dva puta u nizu).

Budući da aritmetička sredina iznosi 37.89, medijan 38.5 i sva četiri moda se razlikuju od aritmetičke sredine i medijana, može se zaključiti kako ovaj histogram ne predstavlja normalnu razdiobu.

Za iduću karakteristiku histograma, neka se uzme slijedeći primjer: želi se napraviti histogram koliko je 2001. godine bilo stanovnika u hrvatskim gradovima iznad 10,000 bez Osijeka, Rijeke, Splita i Zagreba. Podaci su idući: 10208, 10633, 10923, 11231, 11478, 11821, 12090, 13056, 13426, 13735, 14688, 15147, 15329, 15683, 15956, 16696, 19491, 19506, 19644, 23955, 23956, 26468, 27024, 31533, 32029, 34302, 38667, 38839, 42615, 44322, 46833, 53531, 54085, 57460, 71471 (gradovi po redu kao i brojevi su Slatina, Knin, Trogir, Križevci, Sinj, Nova Gradiška, Županja, Rovinj, Makarska, Kutina, Virovitica, Čakovec, Metković, Petrinja, Samobor, Poreč, Đakovo, Požega, Zaprešić, Koprivnica, Solin, Vukovar, Bjelovar, Velika Gorica, Vinkovci, Šibenik, Kaštela, Varaždin, Dubrovnik, Sisak, Karlovac, Slavonski Brod, Sesvete, Pula, Zadar).

Ubacivanjem podataka u Google skriptu, dobije se sljedeći histogram:



Sl. 3.5. Histogram broja stanovnika bez 4 najveća hrvatska grada

Zanimljiva stvar kod ovog histograma je ta što su podaci skupljeni s lijeve strane i kako se košarice kreću prema desno, smanjuje se količina podataka unutar košarica. Takvi histogrami nazivaju se asimetrični histogrami (eng. *skewed*, iskrivljeni), a mogu biti iskrivljeni na desno i na lijevo. Najčešća greška kod određivanja je li histogram iskrivljen prema desno ili lijevo jest gledanje gdje je skupljeno najviše podataka, tj. gdje su košarice najviše i određivanje prema košaricama stranu iskrivljenosti. Najsigurniji test za određivanje strane iskrivljenosti treba gledati aritmetičku sredinu, medijan i mod (iako ni to nije apsolutno točno zbog multimodalnih distribucija). Također, vrijedi obratiti pozornost na koju stranu apscise padaju ili rastu vrijednosti košarica. Ako vrijednosti košarica padaju u pozitivnom smjeru apscise, obično se radi o histogramu iskrivljenom prema desno, a ako vrijednosti košarica padaju u negativnom smjeru apscise, vjerojatno se radi o histogramu iskrivljenom prema lijevo.

Za histogram koji je iskrivljen prema desno obično vrijedi kako je aritmetička sredina veća od medijana, a mod manji od medijana

Za histogram koji je iskrivljen prema lijevo obično vrijedi kako je aritmetička sredina manja od medijana, a mod veći od medijana.

Kada se izračuna aritmetička sredina za histogram na slici 3.5., dobije se rezultat od 26223.743, medijan 19506, a moda nema jer se svaka vrijednost pojavljuje samo jednom. Vidljivo je kako je aritmetička sredina veća od medijana i vrijednost košarica pada u pozitivnom smjeru apscise, a to govori da kako se radi o histogramu iskrivljenom na desno.

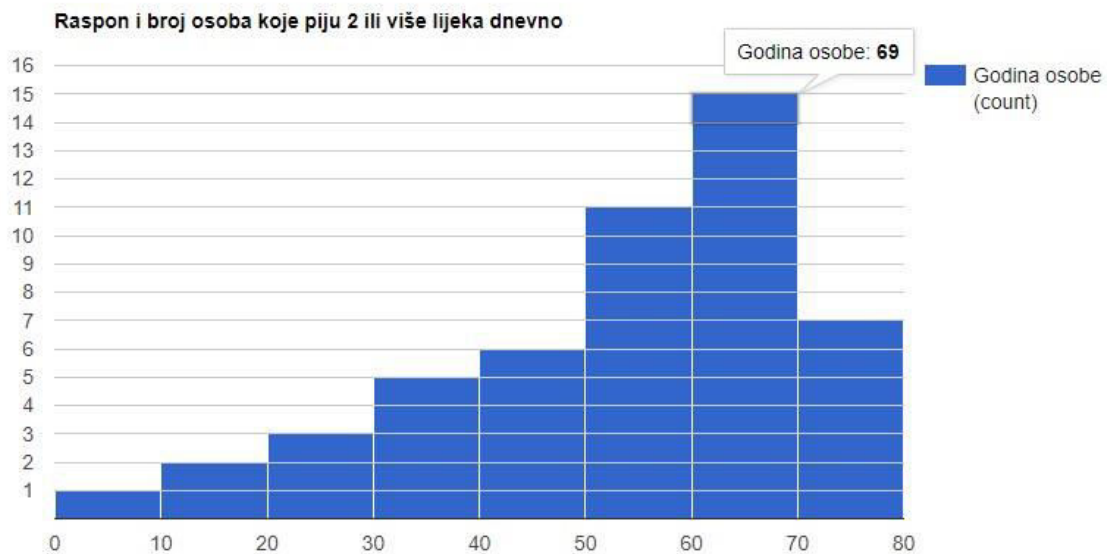
Neka se promotri idući primjer: ispitano je 50 ispitanika koji piju 2 ili više lijeka dnevno. Za potrebe histograma, bitne su godine. Podaci su slijedeći: 8, 16, 19, 23, 26, 28, 31, 31, 34, 36, 39, 41, 42, 46, 46, 48, 48, 52, 52, 52, 53, 54, 54, 57, 57, 58, 59, 59, 61, 61, 61, 61, 62, 63, 63, 65, 65, 66, 66, 68, 68, 69, 69, 71, 72, 72, 72, 75, 78, 79.

Dakle, od 1 do 10 godina je jedna osoba, od 11 do 20 godina je 2 osobe, od 21 do 30 godina je 3 osobe, od 31 do 40 godina je 5 osoba, od 41 do 50 godina je 6 osoba, od 51 do 60 godina je 11 osoba, od 61 do 70 godina je 15 osoba i od 71 do 80 godina je 7 osoba. Slijedeća tablica prikazuje podatke:

Tab. 3.2. Raspon godina i broj ispitanika

Raspon godina ispitanika	Broj ispitanika
1 – 10	1
11 – 20	2
21 – 30	3
31 – 40	5
41 – 50	6
51 – 60	11
61 – 70	15
71 - 80	7

Stavljanjem podataka u Google skriptu, dobije se idući histogram:



Sl. 3.6. Histogram omjera raspona i broja godina osoba koje piju 2 ili više lijekova dnevno

Za razliku od histograma na slici 3.5., vidljivo je kako vrijednosti košarica padaju u negativnom smjeru apscise. Potrebno je još izračunati aritmetičku sredinu, medijan i mod.

Aritmetička sredina iznosi 53.12, medijan je 57.5, a mod 61 (pojavljuje se 4 puta). Rezultati govore kako je aritmetička sredina manja od medijana, a medijan manji od moda s čime se može zaključiti kako se radi o histogramu iskrivljenom na lijevo.

Slijedeća vrsta histograma opisuje situaciju kada postoje podaci koji su raspoređeni s jedne strane apscise i s druge strane apscise. Neka se uzme idući primjer: 50 studenata je pisalo ispit i od 100 bodova, imali su iduće rezultate: 95, 95, 95, 85, 85, 85, 85, 85, 85, 85, 85, 80, 80, 80, 75, 75, 75, 75, 70, 70, 70, 70, 65, 65, 60, 45, 35, 35, 25, 25, 25, 25, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 15, 10, 10, 10, 10, 10, 5, 0, 0.

Tab. 3.3. Raspon broja bodova na ispitu

Raspon bodova	Rezultati učenika	Broj učenika
0 – 9	5, 0, 0	3
10 – 19	15, 10, 10, 10, 10, 10	6
20 – 29	25, 25, 25, 25, 20, 20, 20, 20, 20, 20, 20, 20, 20	14
30 - 39	35, 35	2
40 - 49	45	1
50 – 59	-	0
60 – 69	65, 65, 60	3
70 – 79	75, 75, 75, 75, 70, 70, 70, 70	8
80 – 89	85, 85, 85, 85, 85, 85, 85, 80, 80, 80	10
90 – 100	95, 95, 95	3

Google skripta izbacuje idući histogram:



Sl. 3.7. Histogram podataka iz tablice 3.3.

Ovakav histogram naziva se bimodalan (eng. *bi* - dvostruko) zato što ima dvije odvojene grupe podataka, a svaka od njih ima svoj mod. Vidljivo je na slici 3.7. kako je u grupi od 0 do 49 bodova ima najviše studenata koji su postigli rezultat između 20 i 29 bodova, dok u grupi od 60 do 100 bodova ima najviše studenata koji su postigli između 80 i 89 bodova. Treba također obratiti pozornost na lokalne i globalne maksimume podataka u histogramu.

Globalni maksimum bi bio mod cijelog bimodalnog histograma, a lokalni maksimum je mod jedne grupe podataka i mod druge grupe podataka. U slučaju histograma na slici 3.7. i podataka iz tablice 3.3., mod prvog dijela podataka (studenti koji su imali od 0 do 50 bodova na ispitu) je 20, a mod drugog dijela podataka (studenti koji su imali od 60 do 100 bodova) je 85. Globalni maksimum, odnosno mod cijelog bimodalnog histograma je 20 zato što se taj rezultat najviše ponavlja.

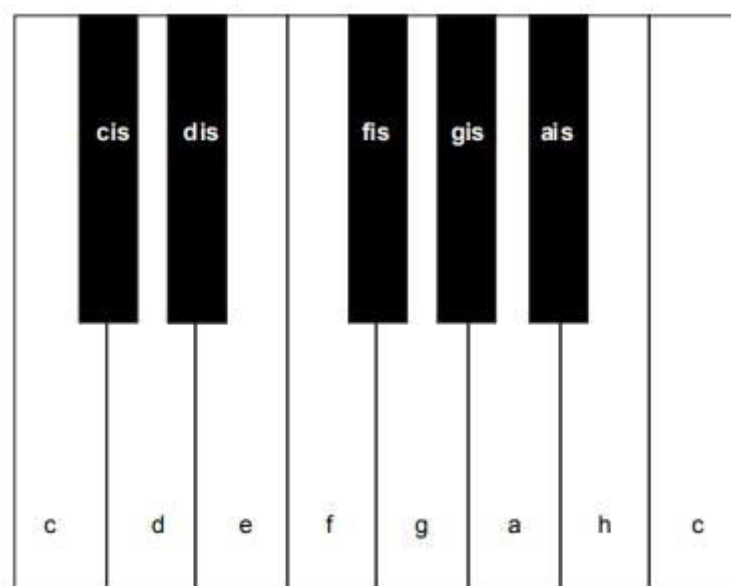
Također, s obzirom da postoji više od jednog moda, moguć je i naziv multimodalni histogram. Razlika između multimodalnog i bimodalnog histograma je u tome što bimodalni histogram ima dva moda, dok multimodalni histogram ima dva ili više moda. Multimodalni histogram se također naziva „češljasti“ histogram. [11]

4. ANALIZA PODATAKA POMOĆU HISTOGRAMA

Kako bi se podaci mogli analizirati, prvo je potrebno objasniti neke pojmove iz glazbe koji će biti korišteni. Glavni pojmovi u glazbi koji će se objašnjavati bit će tonovi i njihove frekvencije koje će Fourierovom transformacijom biti prenošeni prevedeni u frekvencije i obrnuto. Dobiveni rezultati će na kraju biti prikazani u obliku histograma. Programski jezik koji će biti korišten je Python i njegove biblioteke.

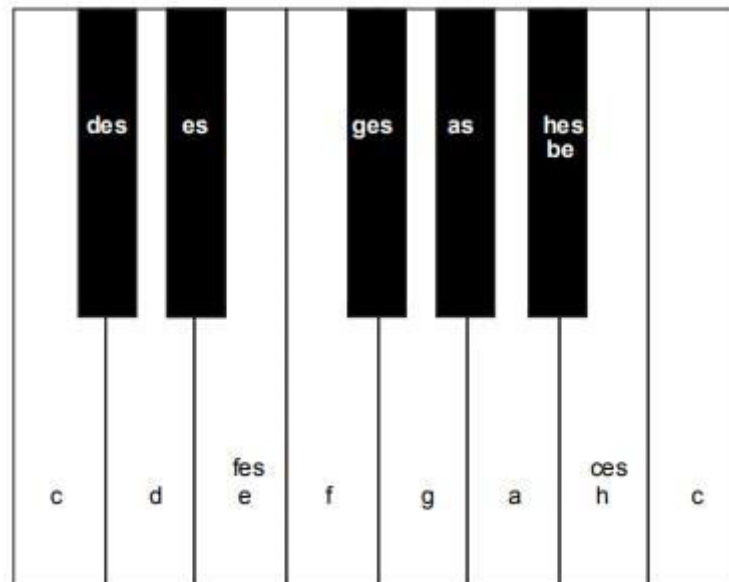
4.1. Glazbeno - teoretska podloga

Glazbena teorija obuhvaća širok spektar pojmova, no samo će se melodija i intervali promatrati za stvaranje histograma. Melodija je niz tonova različite visine i trajanja koji čine zaokružen tonski oblik i glazbeno-sadržajnom smislu tvore smislenu i dovršenu cjelinu. Osnovni tonski niz sastoji se od sedam osnovnih tonova, a imenuje se na dva načina: glazbenom abecedom (C, D, E, F, G, A, H) ili solmizacijom (do, re, mi, fa, so, la, ti). [12] To su osnovni znaci, lako su prepoznatljivi kao bijele tipke na glasoviru, iako se glasovir sastoji od crnih i bijelih tipki. Crne tipke predstavljaju povišene ili snižene tonove - polutonove. Za povišene tonove dodaje se predmetak „is“ na osnovni ton, a za snižene tonove dodaje se predmetak „es“, tako da su povišeni tonovi: cis, dis, fis, gis, ais, his; a sniženi tonovi: ces, des, es, fes, ges, as, hes (hes ili b). Treba napomenuti kako je u Hrvatskoj ton B snizilica za osnovni ton H, a u nekim državama, primjerice, Sjedinjenim Američkim Državama, ton B je osnovni ton koji je jednak hrvatskom tonu H.



Sl. 4.1. Osnovni tonovi i povišice na klavijaturi

Na slici 4.1. vidi se kako između tonova E i F te H i C ne postoje crne tipke jer se radi o prirodnim polutonovima. Želi li se ton E povisiti jedan poluton više, onda se svira ton F, isto vrijedi i za ton H, svira se C.



Sl. 4.2. Osnovni tonovi i snizilice na klavijaturi

Prema slici 4.1. i slici 4.2., vidi se konačan odnos između osnovnih tonova, povisilica i snizilica. Može se reći kako je jednom osnovnom tonu povisilica idućem osnovnom tonu snizilica, osim za E i H. Dakle, ton cis (povisilica osnovnog tona C) je jednaka tonu des (snizilici tona D). Kad se bude definirao histogram, neće se bazirati na snizilicama nego na povisilicama, pošto je za frekvencijsko prepoznavanje tonova dovoljno „prepoznati“ koji je ton. Razlika između povisilica i snizilica je puno važnija kad se slažu skale i gledaju intervali jer kako bi određena skala bila složena, moraju se poštovati točno određeni tonovi i intervali, odnosno, nekad nije moguće reći da je ton fis jednako tonu ges iako po zvuku jesu. To svojstvo u teoriji glazbe se naziva enharmonija – tonovi su jednaki po zvuku, ali se drugačije zapisuju. Povisilice se inače označavaju znakom sličnom ljestvicama (\sharp), a snizilice znakom sličnom slovu *b* (*b*). Ako se spoje svi osnovni tonovi i povisilice (snizilice) u jednu ljestvicu, dobit će se kromatska ljestvica – C, C \sharp , D, D \sharp , E, F, F \sharp , G, G \sharp , A, A \sharp , H. Ona se može ponavljati jer se tonovi nakon određenog vremena ponavljaju u oktavi, ali nakon određenog vremena, ljudsko uho više ne može primjetiti ton zbog frekvencije titranja tog tona.

Frekvencije do koje ljudsko uho odraslog čovjeka može čuti jesu između 16 i 20000 herca. Ispod frekvencije od 16 Hz, radi se o infrazvuku. Primjer za infrazvuk može biti mahanje rukom, stvara se val s frekvencijom od nekoliko Hz, no ljudsko uho ne može čuti tu frekvenciju. Ako je frekvencija iznad 20000 Hz, radi se o ultrazvuku što također čovjek ne može čuti, ali ima primjenu u medicini (ultrazvuk djeteta).

Gitara, samim time i električna gitara, ima sličan način na koji su note raspoređene. Glavni dijelovi su tijelo na kojem je pričvršćen vrat i žice kojih najčešće ima šest. Glavna razlika između akustičnih i električnih gitara je, osim u izgledu, način na koji proizvode zvuk. Akustične gitare imaju rupu u sredini tijela i proizvode zvuk samo od titranja žica. Nakon što se trzne žica, titranje žica prolazi kroz šuplji prostor unutar rupe od akustične gitare i rezoniranjem stvara zvuk. Na taj način se dobije glasniji zvuk nego kad se svira električna gitara (ako električna nije uključena u pojačalo). Električna gitara nema šupljinu, nego je obično napravljena od jednog komada drveta i na sebi ima magnete koji hvataju titraje žice, pretvaraju u električne signale i prenose do pojačala.

Gledajući od dolje prema gore, odnosno, od najtanje prema najdebljoj žici, tonovi su E, H, G, D, A, E (donji ton E i gornji ton E su u različitim oktavama). Na vratu se nalaze pragovi i polja kojih ima najčešće 21, 22 ili 24. Pragovi odvajaju jedno polje od drugog. Svako polje na gitari označava jedan poluton.

Tab. 4.1. Note na poljima gitare

	Broj polja na gitari												
	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
6. žica	E	F	F#	G	G#	A	A#	H	C	C#	D	D#	E
5. žica	H	C	C#	D	D#	E	F	F#	G	G#	A	A#	H
4. žica	G	G#	A	A#	H	C	C#	D	D#	E	F	F#	G
3. žica	D	D#	E	F	F#	G	G#	A	A#	H	C	C#	D
2. žica	A	A#	H	C	C#	D	D#	E	F	F#	G	G#	A
1. žica	E	F	F#	G	G#	A	A#	H	C	C#	D	D#	E

U tablici 4.1. pod stupcem „0.“ prikazane su note gitare koje označavaju trzanje otvorene žice, tj. bez da je ijedno polje pritisnuto. Vidljivo je kako je raspored tonova otvorenih žica identičan rasporedu tonova na dvanaestom polju. Nakon dvanaestog polja, note se počinju ponavljati, primjerice, ako je ton na prvom polju A žice jednak A#, ton na trinaestom polju A žice bit će jednak A#.

Sve dosad navedeno za električnu gitaru vrijedi ako je u najpopularnijem štimu (eng. *tuning*), odnosno, da A_4 iznosi 440 Hz i da žice jesu naštimane kao E, H, G, D, A, E.

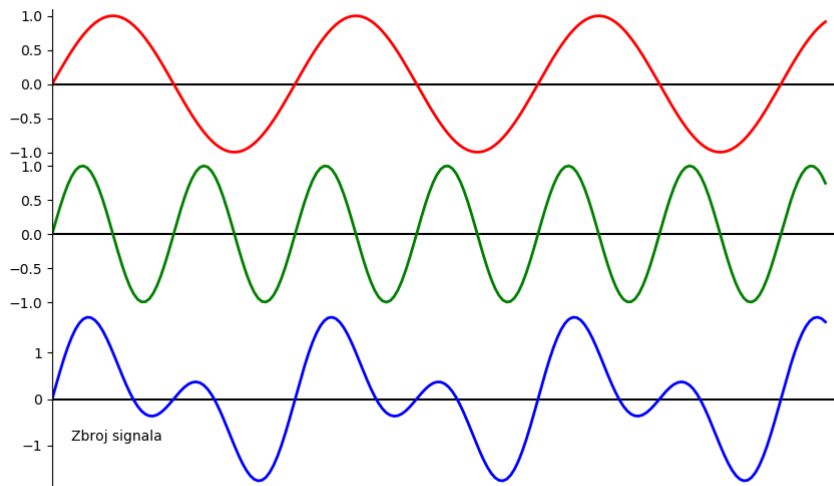
Primjerice, ton A_4 ima 440 Hz, a njegova oktava A_5 (12 polutonova više) ima frekvenciju 880 Hz. Svaka oktava je, frekvencijski gledano, omjera 2:1. Budući da između tonova A_4 i A_5 ima 12 polutonova, svaki od tih polutonova ima omjer $2^{1/12}$ između sebe. [13] Zbog tog omjera i činjenice da je A_4 440 Hz, moguće je iz bilo koje frekvencije odrediti koji to približno ton. Neka dvije note imaju frekvencije f_1 i f_2 i frekvencijski omjer f_2/f_1 . Omjer u oktavi je 2:1, stoga broj oktava koji se nalazi između frekvencija f_1 i f_2 iznosi $\log_2(f_2/f_1)$. Oktave se sastoje od polutonova koje imaju frekvencijski omjer od $2^{1/12}$, A_4 je jednak 440 Hz, frekvencija tražene note f_n jest $2^{1/12}440$ Hz. Lako je za odrediti koliko je polutonova između frekvencije f_n i A_4 formulom $n = 12 \log_2(f_n/440)$.

Frekvencije žica na gitari od najtanje prema najdebljoj žici su 330 (E), 247 (H), 196 (G), 147 (D), 110 (A) i 82 (E) Hz. Te frekvencije vrijede samo kad su žice otvorene i trzaju se bez da se ijedno polje pritisne na vratu. Pritiskanjem bilo kojeg polja na bilo kojoj žici, ton se mijenja i ako se ide više prema ruci kojom se trza, ton će biti viši kao što je opisano na tablici 3.1. Ako je ton viši i ruka koja pritišće polja bude bliže ruci koja trza, žica koja se trza bit će kraća, više će titrati i proizvodit će veću frekvenciju. Primjerice, u tablici 4.1. može se vidjeti kako na najdebljoj E žici peto polje je jednako otvorenoj A žici (odmah ispod E žice). Gitaristi se obično orijentiraju po tome i po zvuku petog polja popravljaju zvuk ako nisu zvuči kao da gitara nije naštimana jer na petom polju žice proizvode sličnu frekvenciju kao slijedeća otvorena.

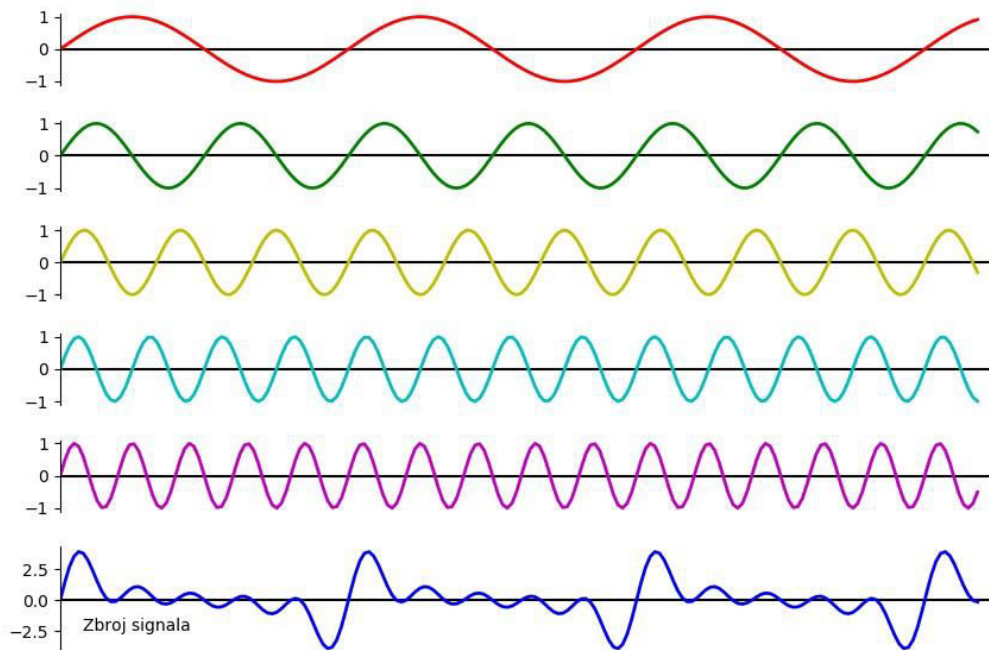
Svaki ton ima svoj MIDI broj na osnovu svoje frekvencije. Za ton A_4 dogovorno vrijedi da njegov MIDI broj iznosi 69. Svaki poluton iznad A_4 ima za jedan broj više, a svaki poluton ispod ima jedan broj manje. Brojevi također se računaju po frekvencijama jer A_4 ima broj MIDI 69 i frekvenciju 440 Hz, znači da će njegova oktava imati MIDI broj 81 i frekvenciju 880 Hz, odnosno, MIDI broj će biti jednak $12\log_2(f_n/440) + 69$. MIDI tonovi se obično koriste u elektronskoj glazbi gdje se koriste MIDI klavijature kao instrument.

4.2. Fourierova transformacija

Svi signali u vremenskoj domeni mogu biti prikazani sumom sinusoidalnih valova različitih amplituda i frekvencija. Na toj činjenici se i zasniva Fourierova transformacija – uzima se signal, i razlomi se na sumu sinusoidalnih valova različitih amplituda i frekvencija. [14]



Sl. 4.3. Zbroj $\sin(x)$ i $\sin(2x)$



Sl. 4.4. Zbroj $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$ i $\sin(5x)$

Prema [15], svaki matematički izraz oblika:

$$A_0 + \sum_{n=1}^{\infty} [A_n \cos(nx) + B_n \sin(nx)] \quad \#(4-1)$$

naziva se Fourierovim redom.

Fourierov polinom je definiran kao:

$$F_n(x) = a_0 + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \quad \#(4-2)$$

gdje je $i = 1, \dots, n$, a konstante a_0 , a_k i b_k su Fourierovi koeficijenti $F_n(x)$.

Fourierovi polinomi su 2π periodične funkcije i mogu se upotrijebiti trigonometrijske transformacije:

$$\sin(mx) \cos(nx) = \frac{1}{2} \{ \sin[(m+n)x] + \sin[(m-n)x] \} \quad \#(4-3)$$

$$\cos(mx) \cos(nx) = \frac{1}{2} \{ \cos[(m+n)x] + \cos[(m-n)x] \} \quad \#(4-4)$$

$$\sin(mx) \sin(nx) = \frac{1}{2} \{ \cos[(m-n)x] - \cos[(m+n)x] \} \quad \#(4-5)$$

Iz (4-3), (4-4) i (4-5) mogu se dokazati sljedeće formule:

za $n \geq 0$:

$$\int_{-\pi}^{\pi} \cos(nx) dx = 0, \quad \int_{-\pi}^{\pi} \sin(nx) dx = 0 \quad \#(4-6)$$

za m i n vrijedi:

$$\int_{-\pi}^{\pi} \sin(mx) \cos(nx) dx = 0 \quad \#(4-7)$$

za $n \neq m$ vrijedi:

$$\int_{-\pi}^{\pi} \cos(mx) \cos(nx) dx = 0, \quad \int_{-\pi}^{\pi} \sin(mx) \sin(nx) dx = 0 \quad \#(4-8)$$

za $n \geq 1$ vrijedi:

$$\int_{-\pi}^{\pi} \cos^2(nx) dx = \pi, \quad \int_{-\pi}^{\pi} \sin^2(nx) dx = \pi \quad \#(4-9)$$

Koristeći (4-2), (4-6), (4-7), (4-8) i (4-9), slijedi:

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(x) dx \quad \#(4 - 10)$$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F_n(x) \cos(kx) dx, \quad 1 \leq k \leq n \quad \#(4 - 11)$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F_n(x) \sin(kx) dx, \quad 1 \leq k \leq n \quad \#(4 - 12)$$

Na ovaj način moguće je bilo koju 2π periodičnu funkciju povezati s Fourierovim redovima.

Iako Fourierova transformacija generalno rečeno rastavlja ili sastavlja signale, treba reći u kojim domenama – pravi rastavljanje vremenskog prikaza signala u frekvencijski prikaz signala. Frekvencijska domena je samo drugačiji način gledanja istog signala. Podaci kao što su diskretne vrijednosti nije moguće prikazati običnom Fourierovom transformacijom, za njih se koristi diskretna Fourierova transformacija.

Ako postoji diskretan skup kompleksnih brojeva (koji se mogu dobiti, primjerice, uzorkovanjem funkcije na nekom intervalu) [16], Fourierova transformacija definirana za kontinuirani skup neće odgovarati. Za takav skup podataka koristi se diskretna Fourierova transformacija:

$$F_j = \sum_{k=0}^{n-1} f_k e^{\frac{2\pi i j k}{n}} \quad \#(4 - 13)$$

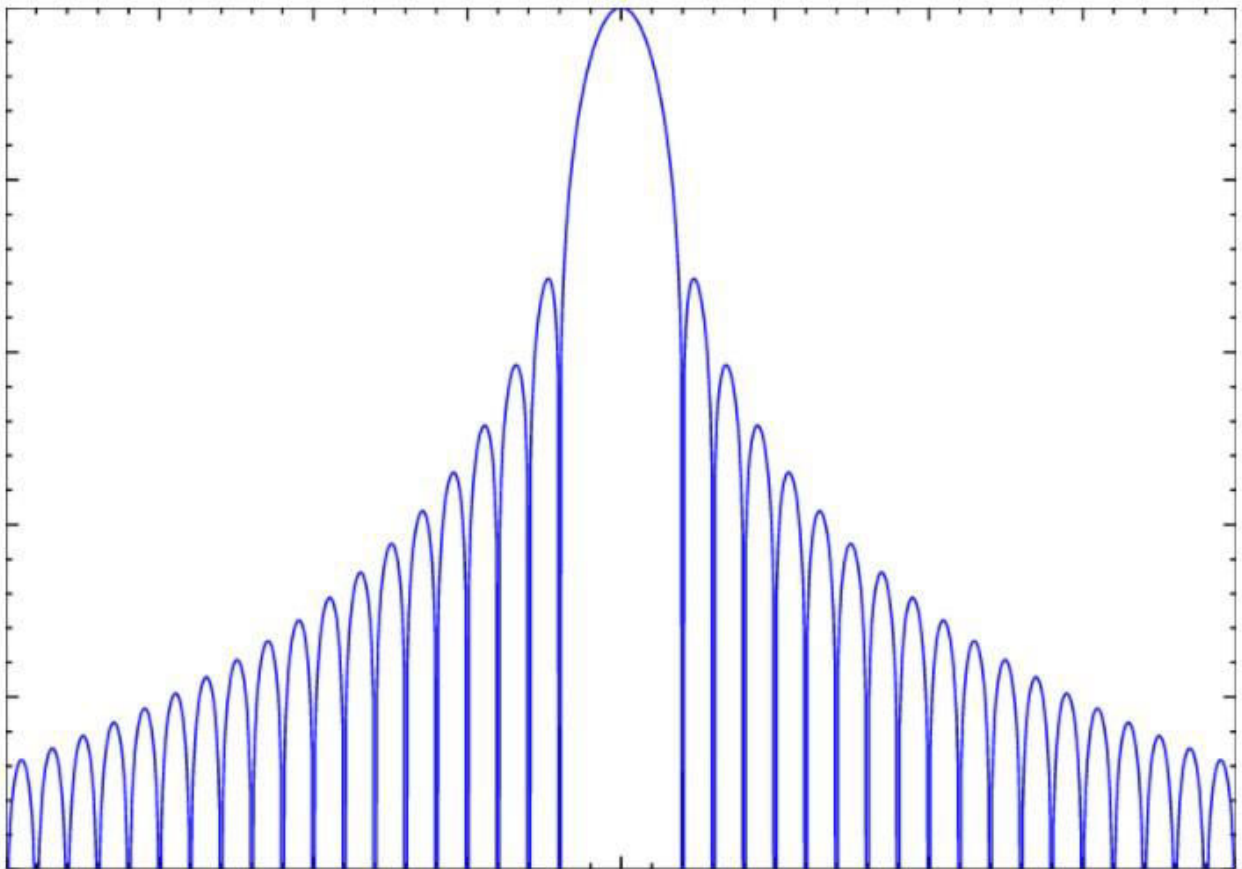
koja će niz kompleksnih brojeva f_0, \dots, f_{n-1} pretvoriti u niz kompleksnih brojeva F_0, \dots, F_{n-1} . Inverzna funkcija od (4-13) je:

$$f_k = \frac{1}{n} \sum_{j=0}^{n-1} F_j e^{\frac{-2\pi i k j}{n}} \quad \#(4 - 14)$$

Također, postoji i brza Fourierova transformacija što je unaprijeđena diskretna Fourierova transformacija koja rastavlja signal, ali zahtjeva manje izračuna. Glavna karakteristika brze Fourierove transformacije (drugim imenom FFT) je izračunavanje diskretne Fourierove transformacije u $O(n \log n)$ vremenu. Zbog svoje brzine i efikasnosti postaje temelj velikog dijela modernih multimedijских primjena (glazba, obrada slika). Ukoliko se valovi mogu zbrajati i stvarati sinusoidalne valove, mogu se i rastavljati na sinusoidalne valove te nakon što je rastavljen, mogu se razlučiti razne frekvencije koje su prisutne u originalnom valu. Takav način obrade valova ima raznih primjena, jedna od njih je kompresija podataka na računalu – ako se

zna koje su frekvencije u podacima manje bitne, moguće je zanemariti te frekvencije čime se dobije bolja reprezentacija podataka u memoriji, odnosno, kompresija podataka. Moguće je također rastaviti zvučne valove čime se ignoriraju određene frekvencije za optimizaciju zvuka i micanje šumova.

Fourierova transformacija na signalu je odličan način analize signala, ali važno je znati točnost transformacije i kako unaprijediti jasnoću signala korištenjem prozora. Korištenjem Fourierove transformacije za izračun frekvencijske komponente signala s točno određenom bazom podataka daje dobre rezultate, ali ako se koriste neodređeni brojevi podataka, dolazi do nepravilnosti u signalu. Ograničavanjem baze podataka koji se analiziraju bi značilo da se postavi prozor na podatke. Postavljanjem prozora smanjuju se amplitude nepravilnosti kod neodređenih broja podataka tako što se vremenska domena množi određenom bazom podataka čija se amplituda mijenja polagano i postepeno čime se dobije ustaljeni val bez oštih prijelaza. Ima više vrsta, neke od njih su Hamming, Blackman, Hann, Kaiser Bessel ili flattop.



Sl. 4.5. Hannov prozor u frekvencijskoj domeni

U ovom radu korišten je Hannov (ili Hanningov) prozor, nazvan po Juliusu von Hannu koji generalno zadovoljava preko 90 posto slučajeva u kojima treba postaviti prozor na signal.

Sinusoidalnog je oblika, ima visok vrh, sa strane se vrijednosti smanjuju i s obje strane dodiruju nulu čime se uklanjaju nepravilnosti u signalu. U praksi je uvijek dobro provjeriti kako različite vrste prozora djeluju na signal kako bi se dobio najbolji rezultat za aplikaciju, no s obzirom na postotak točnosti, Hannov prozor daje zadovoljavajuće rezultate. Hannov prozor se kod analize frekvencija žica gitare primjenjuje na jednako raspoređenim podacima gdje podaci predstavljaju umnožak uzoraka po okviru i broju okvira po Fourierovoj transformaciji.

4.3.Prikupljanje podataka pomoću Pythona

Za unos frekvencije žice gitare u računalo, obradu signala i frekvencije, prepoznavanje tonova od tih frekvencija i iscrtavanje histograma, korišten je Python 3.7 i njegovi paketi:

- PyAudio – paket u kojem se definira koji *line-in* utor će se koristiti, broj tonova koji će se analizirati, podaci za Fourierovu transformaciju i slično
- Mingus – omogućuje rad s tonovima gitare i općenito glazbenom teorijom
- Numpy – Pythonov paket koji služi za brzi rad s matematičkim operacijama, arrayjevima i računanjem s njima
- Matplotlib – stvaranje vizualizacija u Pythonu
- Time – paket kojim je moguće dohvatiti trenutno vrijeme

Paket PyAudio se koristi inicijalizacijom `pyaudio.PyAudio()` funkcije u varijablu i otvaranjem *stream*-a u kojem se definiraju podaci koji će se promatrati.

```
audio_parameters = {'format': pyaudio.paInt16,
                    'channels': 1,
                    'rate': sampling_rate,
                    'input': True,
                    'frames_per_buffer': samples_per_frame,
                    'input_device_index': self.input_device}

stream = pyaudio.PyAudio().open(**audio_parameters)
```

Programski kod 4.1.

Tab. 4.2. Varijable u *audio_parameters*

<i>format</i>	<i>int</i> ili <i>float</i> različitih veličina
<i>channels</i>	broj kanala kroz koje će se preslušavati (jedan ili dva)
<i>rate</i>	broj uzoraka koji će se promatrati
<i>input</i>	definiranjem je <i>False</i> , ali se preslušavaju podaci pa se prebacuje u <i>True</i>
<i>frames_per_buffer</i>	definira broj okvira u međuspremniku (<i>samples_per_frame</i> je prethodno u kodu definiran s vrijednošću od 48000)
<i>input_device_index</i>	označava koji ulaz na računalo (<i>line-in</i>) će se promatrati

Zadnja linija u programskom kodu 4.2. označava pokretanje preslušavanja ulaza signala u računalo.

Prema [4-1], prvo je potrebno definirati funkcije za formule kojima se frekvencije pretvaraju u određene broj i obrnuto te dodati uz pomoć paketa *Mingus* funkcije kojima se u konzoli ispisuju imena tonova iz dobivenih brojeva.

```

"""Dobiveni broj u notu/oktavu note"""

def int_to_note_name(n):

    return str>Note().from_int(n - 12))

"""Note/oktave note u broj"""

def note_name_to_int(name):

    return int>Note(name)) + 12

```

Programski kod 4.2.

Objekt *Note()* je iz paketa *Mingus*, tj. jednog od njegovih *containera*. Jedan problem je da *Note()* umjesto tona H, ispisuje ton B i vraća *int* vrijednosti od 0 do 11 (jer ima 12 tonova i polutonova, a brojanje počinje od 0). Tako, primjerice, poluton ces i B budu jednaki 11. U programskom kodu 4.2., prva funkcija *from_int* iz dobivenog broja vraća ime note i oktavu, počevši od toga da je ton C-0 broj 0 i svakim polutonom, broj raste za 1. Gitara ima raspon tonova od E-2 do C-6 gdje zadnji ton ovisi o tome koliko polja gitara ima. Razlog zbog kojeg u prvoj funkciji postoji -

12, a u drugoj +12 je razlika u oktavama u Mingusovom paketu i načinu na koji on pretvara note i polutonove u brojeve. Cijeli program je sastavljen u jednoj klasi u čijoj *_init_* metodi stoji vrijednost kojeg ulaza u računalo se promatra. U funkciji *listen*, navedeni su podaci koji su potrebni za računanje Fourierove transformacije, frekvencije, glasnoće, trajanje titranja žice te određivanja note.

```
def listen(self, num_notes=np.inf, duration=np.inf, sampling_rate=48000, samples_per_frame=2048,
frames_per_fft=16, instrument_range=('E-2', 'C-6'), min_note_duration=.05, min_volume=10,
output_on=False):
```

Programski kod 4.3.

Tab. 4.3. Argumenti u funkciji *listen*

<i>num_notes</i>	broj tona nakon kojeg se program zaustavlja, moguće ga je izmijeniti kasnije u kodu
<i>duration</i>	varijabla za praćenje trajanja titranja određene note
<i>sampling_rate</i>	broj uzoraka, koristi se u dijelu programa s Fourierovom transformacijom, poželjno je mijenjati broj jer ovisno o njemu, moguće je ukloniti neke šumove koji mogu nastati unosom signala u računalo
<i>samples_per_frame</i>	također se koristi kod Fourierove transformacije te spremanja podataka u međuspremnik
<i>frames_per_fft</i>	korišteno kod stvaranja Hannovog prozora i početka određivanja frekvencije tona
<i>instrument_range</i>	ograničavanje tona i polutona kojeg instrumenta se promatra, za električnu gitaru je, ako je u najpopularnijem štimu, od E-2 do C-6
<i>min_note_duration</i>	vrijednost trajanja titranja žice ispod koje se neće promatrati koja je frekvencija note
<i>min_volume</i>	vrijednost glasnoće titranja žice ispod koje se neće promatrati koja je frekvencija note
<i>output_on</i>	<i>False</i> u ovom dijelu programa, no u izvršnom dijelu programa prelazi u <i>True</i> čime se cijeli programski kod pokreće

Nakon što je definirana funkcija *listen*, definira se Hannov prozor, ograničavaju se tonovi koje će se promatrati spremanjem vrijednosti *instrument_range* u varijable, određuju se vrijednosti Fourierove transformacije Hannovog prozora, spremaju se vrijednosti frekvencija žica koje se promatraju i počinje funkcija *stream* iz PyAudio paketa prema programskom kodu 4.1.

Kako bi se spremile vrijednosti odsviranih tonova za analizu, potrebno je napraviti listu u kojoj će biti MIDI vrijednosti pretvorene u brojeve prema Mingusu te koliko kojih tonova u toj listi ima.

```
n = freq_to_number(freq)

n0 = int(round(n))

current_note = int_to_note_name(n0)

if previous_note != current_note:

    found = False

    for i in range(len(note_list)):

        if note_list[i][0] == n0:

            note_list[i][1] += 1

            found = True

            break

    if not found:

        note_list.append([n0,1])

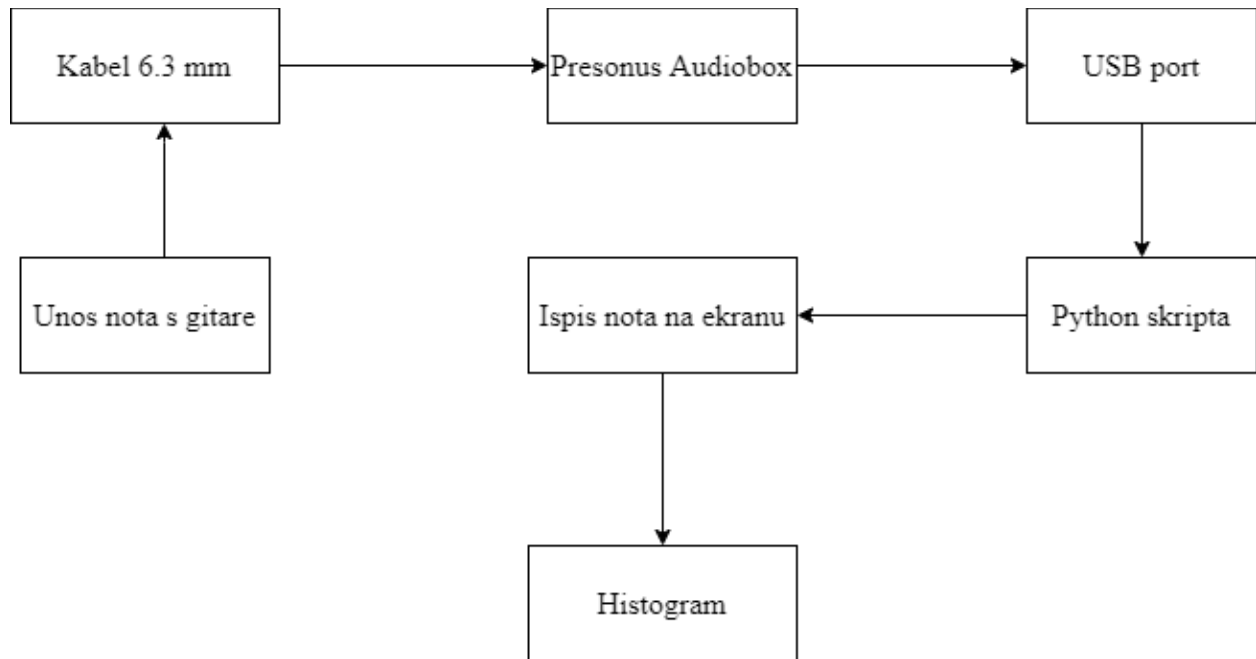
    sound_start_time = time()
```

Programski kod 4.4.

Varijabla n predstavlja MIDI vrijednost dobivene frekvencije tona, ali s obzirom na formulu iz [4-1], potrebno je zaokružiti dobiveni broj jer Mingus ne radi ako su rezultati u decimalnom obliku. Za to je zadužena varijabla $n0$ i u idućem redu, spremljena je u varijablu gdje se nad njom vrši funkcija za pretvaranje broja u ime tona. Zatim je potrebno spremiti dobivene brojeve vrijednosti u listu preko *if* petlje u kojoj se provjerava je li se trenutni ton promjenio. Na početku *if* petlje, postavlja se *bool* operator za provjeru postoji li taj ton trenutno u listi i tad preko *for*

petlje slijedi provjera za dodavanje količine trenutnog tona. Ako *for* petlja ne pronalazi taj ton, preko funkcije *append* dodaje se na trenutnu listu.

Sviranjem gitare dobije se ton koji se preko kabela debljine 6.3 mm šalje u Presonus Audiobox, koji je spojen USB kabelom u računalo na kojem je pripremljena Python skripta i u kojoj je podešeno da se očitavanje tona izvršava na utoru *line-in* utoru gdje je Presonus Audiobox, zatim se program izvršava, na ekranu se ispisuju tonovi, lista s tonovima i stvara se histogram.



Sl. 4.6. Dijagram toka unosa frekvencije žice u računalo

Presonus Audiobox je audio sučelje za snimanje koje na prednjoj strani ima dva 6.3 mm utičnice, potencijometre za oba utičnice, glasnoću uređaja, glasnoću slušalica (čiji je utičnik na stražnjoj strani uređaja) i regulator glasnoće između ulaznih i izlaznih signala. Gitara se u njega spaja s 6.3 mm kabelom. Presonus Audiobox nema vanjsko napajanje nego se napaja preko USB 2.0 utičnika kojim se spaja u računalo. Spajanjem gitare u jedan od 6.3 mm utičnika, moguće je podesiti jačinu izlaznog signala, odnosno, koliko jako se želi prenositi signal iz gitare u računalo te smanjiti kašnjenje signala od gitare do računala. Bez obzira što neke matične ploče na računalima imaju *line-in* utičnice 3.5 mm ili posebne utičnice za mikrofonski signal od 3.5 mm i mogu primiti signale od gitare ako se iskoristi adapter koji pretvara kabel od 6.3 mm u 3.5 mm, signali će previše varirati i neće dati toliko točne rezultate kao što će dati korištenje Presonus Audioboxa.



Sl. 4.7. Presonus Audiobox

Za uređivanje i pokretanje koda, korišten je Pycharm. Pokretanjem koda, u konzoli se ispisuju koji je ton, koliko je trajao, kolika je glasnoća i sveukupni niz koji je dosad napunjen s tonovima u kojem je prvi član MIDI vrijednost tona, a drugi član koliko se dosad taj ton puta pojavio.

```

freq: 246.94 Hznote: 'B-3' duration: 0.05239534378051758 volume: 18.4482523182709
40
[[52, 45], [40, 47], [59, 20], [71, 5], [64, 13], [58, 2]]
freq: 82.41 Hznote: 'E-2' duration: 0.0879671573638916 volume: 18.566204355621505
52
[[52, 45], [40, 48], [59, 20], [71, 5], [64, 13], [58, 2]]
freq: 164.81 Hznote: 'E-3' duration: 0.0790104866027832 volume: 18.909786084106255
40
[[52, 46], [40, 48], [59, 20], [71, 5], [64, 13], [58, 2]]
freq: 82.41 Hznote: 'E-2' duration: 0.08614635467529297 volume: 22.559731487816087
[[52, 46], [40, 49], [59, 20], [71, 5], [64, 13], [58, 2]]

```

Sl. 4.8. Ispis rezultata u Pycharm konzoli

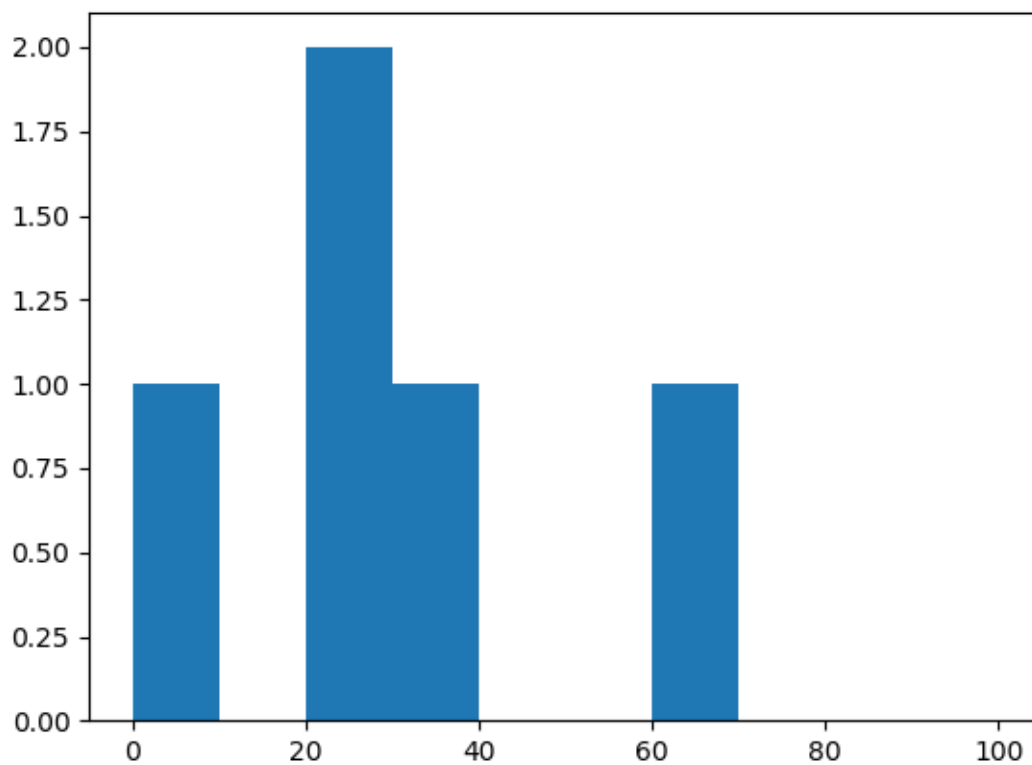
4.4.Prikaz prikupljenih podataka u obliku histograma

Za set podataka korišteno je 100 odsviranih tonova. Prije početka eksperimenta, potrebno je provjeriti glasnoću unosa signala gitare u računalo u postavkama operacijskog sustava. Ako je glasnoća unosa prevelika, program će hvatati puno tonova od šumova signala i spremati rezultate u listu te ih prikazivati na histogramu, a ako je glasnoća premala, program neće ništa prepoznati, tj. čuti unos signala u računalo. Potrebno je naći određenu granicu na kojoj se gitara dovoljno čuje, a da program ne uzima previše šumova kao note. Primjeri nad kojim su vršenja mjerenja i iz kojih su izvedeni histogrami nalaze se u prilogu. Također, u primjerima je gitara svirana trzalicom što će dovesti do većih odstupanja u rezultatima jer sviranje trzalicom grebe metalne žice i stvara dodatne šumove koje računalo može procesirati kao note. Jedan problem koji treba naglasiti prije početka mjerenja je da program, odnosno, računalo ne može prepoznavati isti ton cijelo vrijeme, samo onda kad se dogodi promjena s jednog na drugi ton. Kad se dogodi promjena s jednog na drugi, onda se spremaju u niz. Primjerice, u dvadeset trzanja, devetnaest uzastopnih bude E-2, a odmah iza toga jedan F-3, računalo će prepoznati to kao dva trzanja, jedno E-2 i jedno F-3.

Prvi primjer bazira se na naizmjeničnom ponavljanju 3 note: E-2, E-3 i F-3. S obzirom na snimku, pretpostavka je da bi trebalo biti najviše E-2 i E-3 tonova. U konzoli je ispisan slijedeći niz: [[40, 24], [64, 69], [52, 38], [65, 23], [53, 2]]. Kao što je navedeno u [2-4], prvi član predstavlja MIDI vrijednost note što znači da u ovoj snimci, računalo po frekvencijama prepoznaje 5 tonova:

E-4 [64, 69], E-3 [52, 38], E-2 [40, 24], F-4 [65, 23], F-3 [53, 2]

Note su točne, ali nisu identične oktave prema onome što je svirano iako nema previše šumova. Razlog zašto broj tonova u oktavama nije točan je što vibriranjem žica, ton može vibrirati u različitoj oktavi ako ga se pusti da odzvanja. Broj tonova koji je trebalo mjeriti bio ograničen na 100, ali program uvijek uzme više, u ovom mjerenju je uzeo čak 56 posto više (156).



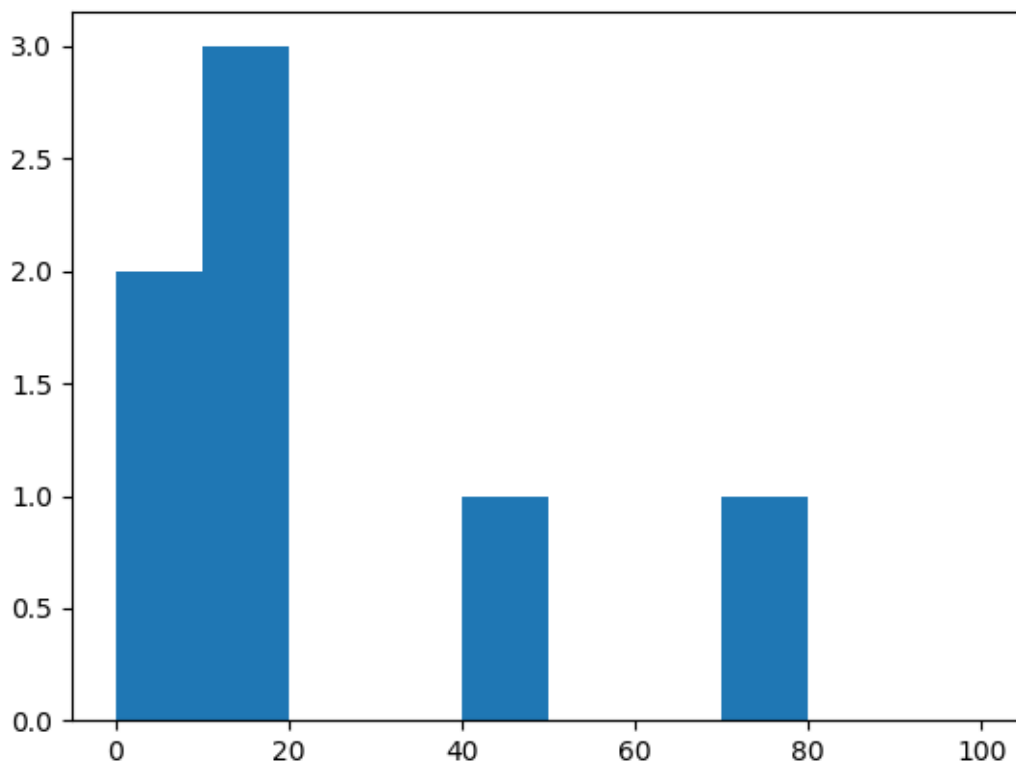
Sl. 4.9. Histogram prvog primjera

S obzirom na malu količinu podataka koja se analizira histogramom, svejedno se može iz njega zaključiti da od 5 tonova odsviranih, najviše se nalazi u grupi od 20 i 40, njih 3. Najviše odudaraju tonovi se nalaze u košarici od 0 do 10 (F-3) te 60 do 70 (E-4).

Drugi primjer je skoro identičan prvome primjeru, dodan je još jedna ton, A#-3. Slušanjem snimke, svaki drugi put kad bi se trebao pojaviti ton F-3, u primjer je ubačen jedan A#-3 ton. U konzoli bi trebalo biti četiri tona, a ovo je ispis konzole:

E-3 [52, 47], E-4 [64, 70], F-4 [65, 14], A#-3 [58, 10], E-2 [40, 15], H-4 [71, 2], F-3 [53, 4]

Program je prepoznao 162 odsvirane note, odstupanje od 62%. Note su kao i u prvom primjeru, točne, ali u različitim oktavama. U ovom primjeru se dva puta pojavljuje ton H-4 što se može protumačiti kao šum nastao od trzalice, način sviranja ili ljudska pogreška kod sviranja. Također, moguće je da je ton H alikvotni ton [17] od tona E.



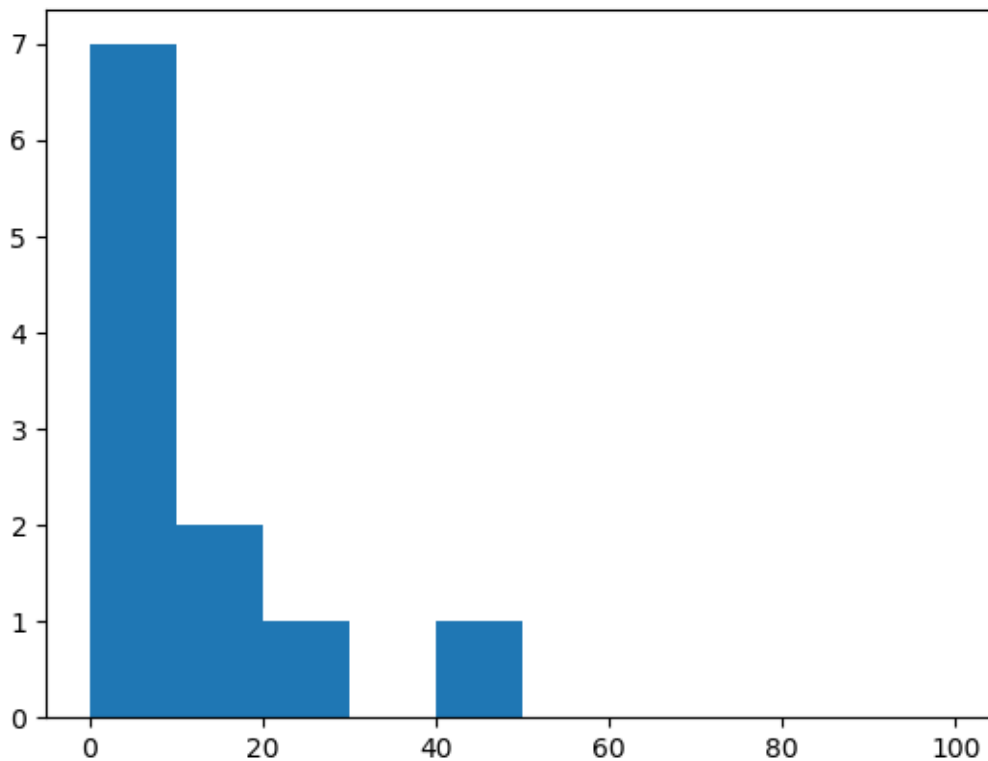
Sl. 4.10. Histogram drugog primjera

Dodavanjem samo jednog tona u prvi primjer, vidi se lagana asimetrija u histogramu. Pet od sedam tonova koje je program pronašao se nalazi u grupi od 0 do 20, dok su dva tona koji dominiraju (E-3 i E-4) odvojeni od ostatka.

Treći primjer spaja prvi i drugi primjer te dodaje još 6 tona, A#-2, F#-2, G-2, H-2, C-3, F-2. Dodatkom šest tona na primjere, trebalo bi biti 10 tonova u ispisu. Slijedi ispis niza iz konzole:

E-3 [52, 27], E-4 [64, 42], F-4 [65, 12], A#-3 [58, 10], F-3 [53, 6], F#-2 [42, 5], G-2 [43, 5], H-3 [59, 5], C-4 [60, 5], E-2 [40, 8], F-2 [41, 1]

Analizirani tonovi su s obzirom na odsvirano točni, ali kao i u prvom i drugom primjeru, različite oktave. Ima više analiziranih tonova nego što je odsvirano, no svi su točni, samo u različitim oktavama. U ovom primjeru, ton H je odsviran za razliku od prošlog primjera. Broj odsviranih tonova prema programu je 126, odstupanje je 26%, ali je znatno manje nego u prvom i drugom primjeru.



Sl. 4.11. Histogram trećeg primjera

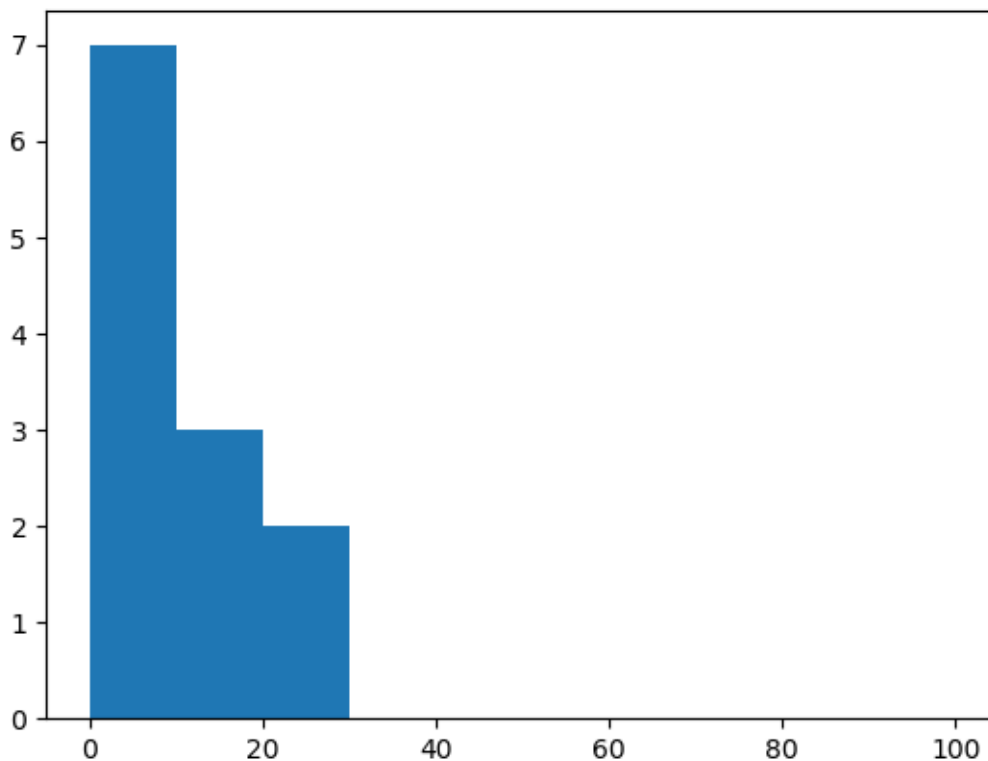
Nakon šest dodanih tonova, histogram se sve više iskrivljuje prema lijevo. Košarica između 0 i 10 najviše odudara u cijelom histogramu jer se u njoj nalazi čak 7 tonova (F-3, F#-2, G-2, H-3, C-4, E-2 i F-2), dok je košarica između 40 i 50 odvojena sadrži ton E-2.

Četvrti primjer je kompletno drugačiji od dosad analiziranih, gledajući melodiju i žice na kojima se svira. Prva tri primjera svirana su pretežito na prve tri žice (E, A, D), dok je četvrti primjer sviran isključivo na zadnje dvije (E i H). Sadrži se od idućih tonova: A-4, G#-4, C-5, H-4, E-4, F#-4, D-5, G-4, E-5.

Ispis u konzoli:

H-4 [71, 25], A-4 [69, 19], G#-5 [80, 10], E-5 [76, 8], F#-4 [66, 3], C-5 [72, 22], D-5 [74, 11], G-5 [79, 3], G-4 [67, 9], A-5 [81, 9], E-4[64, 1], F#-5 [78, 1]

Program je prepoznao 121 ton, odstupanje od 21%. Svi tonovi koje je program prepoznao su točni uz njihove oktave, jedino nije prepoznao G#-4, ali je 10 puta prepoznao njegovu oktavu G#-5. U analizi ovog primjera nema nijednog tona koji odudara od odsviranog što se može pripisati točnosti odsviranome ili činjenici da je veće titranje tanjih žica te da su manja preklapanja frekvencija i manja mogućnost stvaranja odudarajućih alikvotnih tonova.



Sl. 4.12. Histogram četvrtog primjera

S obzirom na veću količinu tonova i manje ponavljanje jednog tona kao što je dosad bio slučaj u prijašnjim primjerima, histogram se sve više povlači prema lijevoj strani te nema nijedna košarica koja previše odudara s desne strane. Vidljivo je kako je u košarici od 0 do 10 najveća koncentracija tonova, njih 7.

Tab. 4.4. Točnost analiziranih tonova u prva četiri primjera

	Odstupanje u broju analiziranih tonova	Broj netočno analiziranih tonova
Primjer 1	56	0
Primjer 2	62	1
Primjer 3	26	0
Primjer 4	21	0
Ukupno	41.25	1

U tablici 4.4. kod zbrajanja netočno analiziranih tonova, sve oktave istog tona su zanemarene, odnosno, nisu uzimane u obzir kao različit ton od odsviranog, u tablici su ubačeni tonovi koji ne pripadaju u odsviranom primjeru. Odstupanje u broju analiziranih tonova je veliko, što se može pripisati brzini sviranja, sporosti analize programa, titranju žica i alikvotnim tonovima (računalo iz jednog odsviranog tona možda prepozna određeni ton, ali i njegovu oktavu i napuni međuspremnik). U prvom i drugom primjeru, koji nisu komplicirani što se izvedbe tiče, odstupanje je veće nego u trećem i četvrtom. Razlog je što imaju manje tonova u sebi, sporiji su i izvode se, za razliku od četvrtog primjera, na dvije najdeblje žice na gitari koje šalju veće valove prema magnetima gitare. Program iz trzaja najdeblje žice uzima informaciju kao da je to E-2 ili njegove oktave E-3 i E-4 zbog manje frekvencije koju žica šalje i alikvotnih tonova. Četvrti primjer ima najmanje odstupanje jer je u cijelosti odsviran na dvije najtanje žice gitare, koje titraju najvećom brzinom od svih otvorenih žica. Jedini netočno analizirani ton u sva četiri primjera dogodio se u drugom primjeru.

Peti primjer sastoji se od 4 akorda koji su uzastopno svirani dok je između mijenjanja svirano par tonova iz tih akorda. Teško je napraviti analizu koji se svi točno tonovi ponavljaju jer je namjerno nepravilno svirano za potrebe demonstracije histograma. Analizu dodatno otežava činjenica da je akord u glazbenoj teoriji istodobno zvučanje triju ili više tonova različite visine⁴, a između akorda je svirano par tonova iz tih akorda. Naglasak je na koje akorde se svira: a-mol, C-dur, G-dur i e-mol.

Niz u konzoli je:

E-4 [64, 14], A-4 [69, 9], A-3 [57, 14], C-5 [72, 5], G-5 [79, 5], C-4 [60, 8], G-4 [67, 6], H-5 [83, 6], D-5 [74, 13], F#-5 [78, 7], G-3 [55, 7], G-2 [43, 6], E-3 [52, 7], H-3 [59, 8], E-5 [76, 4], D-4 [62, 1], H-4 [71, 3], E-2 [40, 2], A-2 [45, 1], C-3 [48, 1], A#-3 [58, 1]

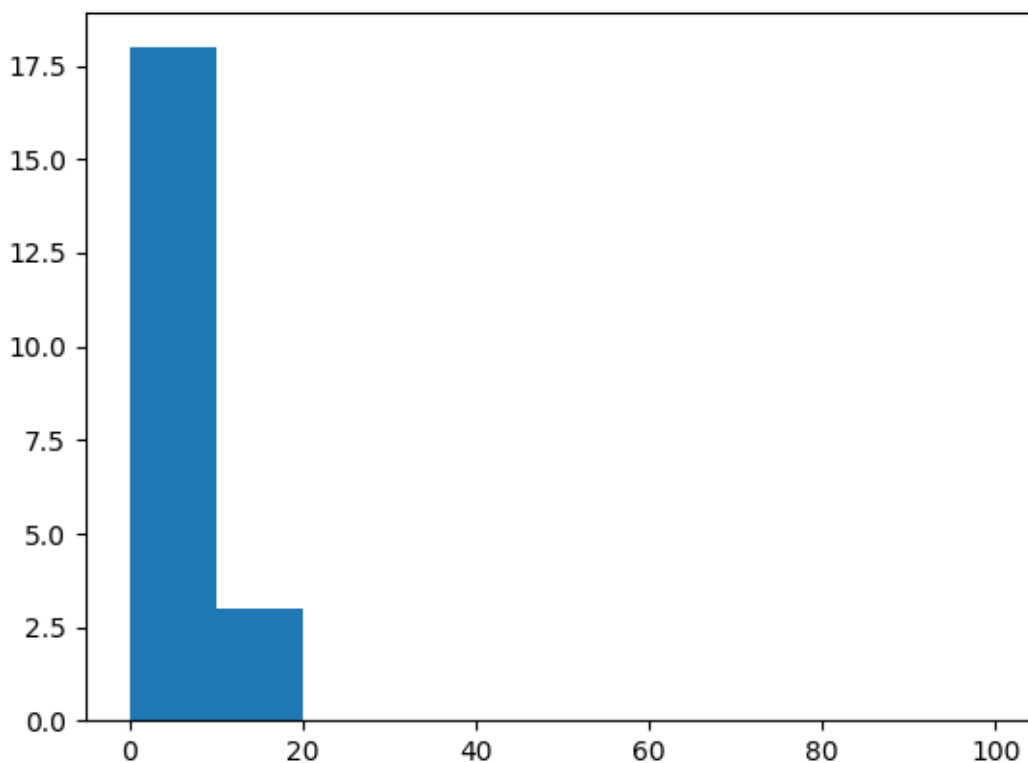
Program je prepoznao 128 nota, odstupanje od 28%.

Od svih analiziranih primjera dosad, ovo je najveći broj različitih tonova koji se dosad pojavio. Budući da je akord sam po sebi zvučanje triju ili više tonova različite visine (te svaki od ta tona može imati šum ili alikvotni ton za vrijeme rezoniranja), program teško određuje točno koji je ton ako se u isto vrijeme svira više različitih tonova. Pobroje li se svi tonovi, bez obzira na oktave, analizom tonova i poznavanjem intervala u glazbenoj teoriji moguće je odrediti o kojim akordima se radi.

Tab. 4.5. Ukupan broj tonova u petom primjeru po oktavama

		Tonovi							
		E	A	C	G	H	D	F#	A#
Oktave	2	2	1	-	6	-	-	-	-
	3	7	14	1	7	8	-	-	1
	4	14	9	8	6	3	1	-	-
	5	4	-	5	5	6	13	7	-
	Ukupno	27	24	14	24	17	14	7	1

⁴ <https://www.enciklopedija.hr/natuknica.aspx?id=1136>



Sl. 4.13. Histogram petog primjera

Usporede li se tablica 4.5. i slika 4.13., vidjet će se kako 3 tona prevladavaju – E, G i A. Kao što je prethodno navedeno, u primjeru su svirani a-mol, C-dur, G-dur i e-mol. Kad ne bi bilo navedeno koji se akordi sviraju u primjeru, a pogledalo bi se note, osoba koja se razumije u intervale i slaganje skala u glazbenoj teoriji bi mogla zaključiti kako se radi o tim akordima i o C-durskoj skali, pogotovo zbog učestalijeg pojavljivanja C i H tona (dovoljno puta su se pojavili u nizu da se mogu protumačiti kao odsvirane note, a ne kao šumovi ili alikvotni tonovi).

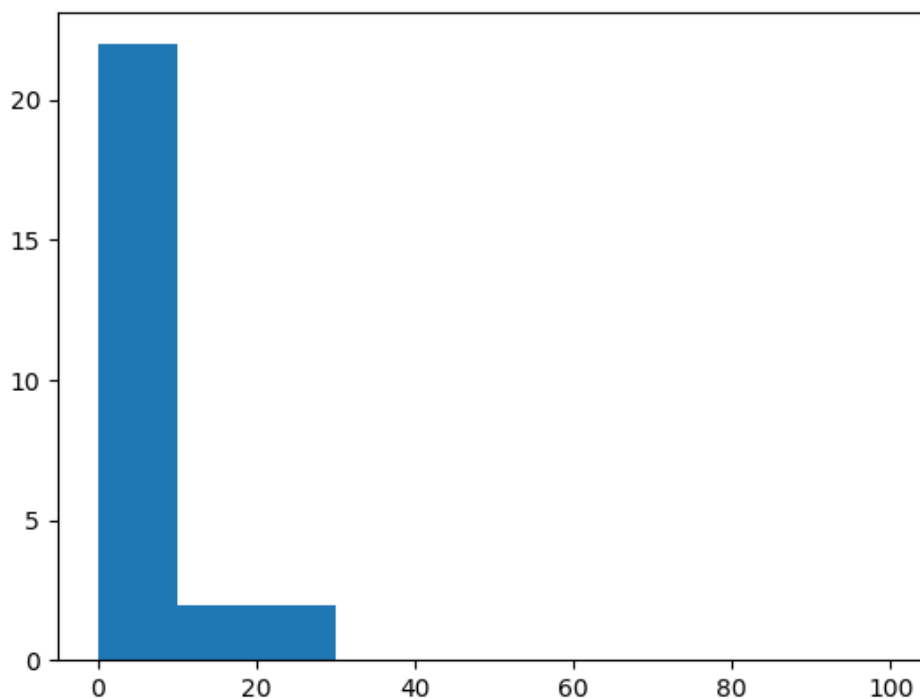
Šesti primjer je najkompliciraniji za analizirati pomoću programa jer se sastoji od samog sviranja akorda i to na način da nisu odsvirani potpuno jasno. Sastoji se od A-dura, a-mola, G-dura i F-dura uz razne udarce na žice dok se ne drži pritisnuto nijedno polje što dodatno otežava analizu.

Program izbacuje niz:

A-3 [57, 25], A-2 [45, 23], A-4 [69, 9], E-5 [76, 2], E-4 [64, 12], D-3 [50, 5], D-4 [62, 6], G-3 [55, 10], G-2 [43, 4], E-2 [40, 2], H-4 [71, 4], C-4 [60, 5], C-3 [48, 3], F-3 [53, 8], C#-4 [61, 1], E-3 [52, 7], H-3 [59, 5], F-2 [41, 5], F-4 [65, 1], C-5 [72, 1], F#-2 [42, 1], C#-3 [49, 3], D#-3 [51, 1], G#-4 [67, 1], G#-2 [44, 1], A#-3 [58, 1]

Tab. 4.6. Ukupan broj tonova u šestom primjeru po oktavama

		Tonovi											
		C	C#	D	D#	E	F	F#	G	G#	A	A#	H
Oktave	2	-	-	-	-	2	5	1	4	1	23	-	-
	3	3	3	5	1	7	8	-	10	-	25	1	5
	4	5	1	6	-	12	1	-	-	1	9	-	4
	5	1	-	-	-	2	-	-	-	-	-	-	-
	Ukupno	9	4	11	1	23	14	1	14	2	57	1	9



Sl. 4.14. Histogram šestog primjera

Sveukupan broj tonova koje je program analizirao je 146, odstupanje od 46%. U ovom primjeru su se pojavili svi tonovi kromatske ljestvice, ali neki samo po jednom (D#, F#, A#). Zbrajanjem svih oktava tona A, vidi se kako je upravo on najučestaliji ton, što se može i zaključiti s obzirom na činjenicu kako od četiri akorda u primjeru, jedan je A-dur, a drugi a-mol. Prema slici 4.14., dobiveni histogram je potpuno iskrivljen prema lijevo i glavnina podataka se nalazi u košarici između 0 i 10. Od svih analiziranih primjera, šesti primjer ima najveću koncentraciju podataka u prvoj košarici jer se sadrži kompletno od akorda, bez ijednog tona koji je odsviran samostalno.

5. ZAKLJUČAK

Kroz ovaj rad opisivani su histogrami, osnove glazbene teorije, karakteristike električne gitare, tonovima na njoj, osnovni podaci o frekvencijama i valovima te primjena histograma na frekvencijama žica električne gitare. Analiza frekvencije žica je u širokoj upotrebi u različitim primjenama, konkretno na električnim gitarama se koriste kao štimeri kako bi se namjestila frekvencija žice na određenu vrijednost. Histogrami analiziranih frekvencija daju drugačiji pogled na analizu frekvencija žica. U radu je kroz šest primjera uspoređeno kako se program ponaša i vrste histograma koje pokazuje na osnovu odsviranih primjera. Ako je korisnik dobro upoznat s glazbenom teorijom, može iskoristiti kao program za vježbanje i nadogradnju vlastitog znanja u glazbi.

Primjeri su testirani odvojeno te su njihovi rezultati uspoređivani kroz točnost broja analiziranih tonova i odstupanja od količine analiziranih podataka. U histogramima prva dva primjera, vidljivo je veliko odstupanje u broju analiziranih tonova, što se pripisuje različitoj debljini žica gitare i rastućoj frekvenciji titranja ako je žica tanja. U ostalim primjerima, gdje raste broj odsviranih tonova i broj žica na kojima se svira, vidi se smanjivanje odstupanja od broja analiziranih tonova.

Iduće iteracije rada bi se mogle bazirati samo kao štimer i prenijeti na mobilne uređaje ili aplikacija s mogućnošću snimanja gdje bi se nakon sviranja izbacivala analiza odsviranog u kojem bi korisnik mogao vježbati dijelove glazbene teorije koji ga zanimaju (skale, intervali, harmonija, melodija...).

Računalu je teško savršeno analizirati sviranje gitare jer, osim što je samo analiziranje titranja žica teško, uvijek je prisutan i ljudski faktor, odnosno, koliko dobro čovjek svira zadano djelo. Također, neke tehnike mijenjanja tonova i polutonova u sviranju gitare je računalu vrlo teško prepoznati. Kad bi se radilo o MIDI klavijaturama koje daju točno određen MIDI ton, tad ne bi bilo alikvotnih tonova, preklapanja signala i frekvencija, program bi bio puno precizniji, bez obzira na ljudske pogreške.

LITERATURA

- [1] X. Liu, D. Wang, *A spectral histogram model for texton modeling and texture discrimination*, Department of Computer Science, Florida State University, studeni 2002., <https://www.sciencedirect.com/science/article/pii/S0042698902002973>, rujan 2020.
- [2] T. DiCola, *FFT: Fun with Fourier Transforms*, Adafruit Industries, listopad 2013., <https://learn.adafruit.com/fft-fun-with-fourier-transforms/tone-input>, rujan 2020.
- [3] N. Hilbish, *Multiple Fundamental Frequency Pitch Detection for Real Time MIDI Applications*, VCU, 2012., <https://scholarscompass.vcu.edu/cgi/viewcontent.cgi?article=3852&context=etd>, rujan 2020.
- [4] G. Hori, S. Sagayama, *HMM-Based Automatic Arrangement For Guitars With Transposition and its Implementation*, Atena, Grčka, 2014., <https://quod.lib.umich.edu/cgi/p/pod/dod-idx/hmm-based-automatic-arrangement-for-guitars-with.pdf?c=icmc;idno=bbp2372.2014.193;format=pdf>, rujan 2020.
- [5] Y. Ioannidis, *The History of Histograms (abridged)*, *Proceeding VLDB '03 - Proceedings of the 29th international conference on Very large data bases - Volume 29*, stranice 19-30, Berlin, Njemačka — 12.9.2003.
- [6] J. Frost, *Using Histograms to Understand Your Data*, *Statistics By Jim*, <https://statisticsbyjim.com/basics/histograms>, rujan 2020.
- [7] NCSS Statistical Software, NCSS.com, *Chapter 143, Histograms*, <http://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Histograms.pdf>, rujan 2020.
- [8] Google Charts, *Histogram*, <https://developers.google.com/chart/interactive/docs/gallery/histogram>, rujan 2018.
- [9] Tournament Director, *Berger Pairings*, http://www.tournamentdirector.co.uk/berger_pairings.html, kolovoz 2018.
- [10] J. Medhi, *Statistical Methods: An Introductory Text*, New Age International, 1992.
- [11] P. M. Charantimath, *Total Quality Management*, Pearson Education Media, 2001.
- [12] S. Šamanić, *Metodički priručnik za glazbeni praktikum*, https://bib.irb.hr/datoteka/611198.Lektorirani_Prirunik1.pdf, kolovoz 2020.
- [13] J. Wolfe, *Note names, MIDI numbers and frequencies*, <https://newt.phys.unsw.edu.au/jw/notes.html>, kolovoz 2020.

- [14] National Instruments, *Understanding FFTs and Windowing*,
<https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf>, kolovoz 2020.
- [15] V. Grozdanić, *Fourierovi redovi, Polinomi i Integrali*, <https://vdocuments.mx/vedranagrozdanic-fourierovi-redovi-polinomi-i-integrali.html>, rujan 2020.
- [16] D. Matijević, *Fourierov red i Fourierova transformacija*, stranice 1-2, 12-13, Zagreb, Hrvatska – 2011.
- [17] Hrvatska enciklopedija, svezak I, Zagreb, 1941. – stranice 244-245

SAŽETAK

Naslov: Histogram

Ovaj rad prikazuje histograme kao vrstu predodžbe frekvencije žice gitare te točnost računalne analize unosa frekvencija žica gitare u računalo kroz šest primjera. Pokazane su vrste histograma te kakve će histograme izbaciti ti primjeri. Također, uspoređena je točnost unosa frekvencija žica te određeni su najveće prednosti i najveći nedostaci kod analize. Primjeri su poredani po kompliciranosti za analizu računalu, objašnjeni su dobiveni histogrami te su određene prednosti i nedostaci programa. Kroz primjere, koji rastu po kompliciranosti, povećava se točnost u analizi broja tonova, ali smanjuje točnost određivanja tonova te mogućnost prilagodbe programa. Prikazano je kako je program više namjenjen za analizu uzastopno sviranih niza tonova različite visine i trajanja, a ne akorda te bi bio idealan kao štimer za gitare ili kao alat za vježbanje glazbene teorije.

Ključne riječi: histogram, glazbena teorija, gitara, frekvencija

ABSTRACT

Title: Histogram

This paper shows histograms as a way of presenting value of guitar string frequency and the accuracy of analysing those frequencies through six examples. Also, the types of histograms are shown, what histograms those examples are outputting, accuracy of inputting guitar string frequency in computer is compared and biggest advantages and disadvantages are defined. Examples are sorted by how complex they are for the computer to analyse and histograms that have been outputted have been described. Through those examples, which get more complex as they are analysed, accuracy in analysing the number of tones is growing, but defining which tone it is grows more difficult for the computer. It is shown how the program is more suited for melody and less for harmony and it would be best used as a guitar tuner or as a tool to practice music theory.

Keywords: histogram, music theory, guitar, frequency

ŽIVOTOPIS

Dinko Tarakčija rođen je u Osijeku 19.7.1993. godine. Završava Osnovnu školu Vladimir Nazor u Čepinu 2008. godine s odličnim uspjehom i upisuje II. gimnaziju u Osijeku koju završava 2012. s odličnim uspjehom. Iste godine upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku te je 2015. godine upisao diplomski studij na istom fakultetu, izborni blok informacijske i podatkovne znanosti.

Potpis: _____

PRILOZI

Prilozi se nalaze na CD-u.

1. Diplomski rad u .pdf formatu
2. Diplomski rad u .docx formatu
3. Izvorni kod s kojim su izvođena mjerenja preuzet s <https://github.com/mzucker>, modificiran za rad s histogramima
4. Šest snimki iz kojih su napravljena mjerenja u .wav formatu