THE PALOS VERDES
ARTS AND SCIENCE JOURNAL

Celestial Symphony: MIDI Sonification of Multiparametric Supernova Data into a Euphonious Composition

Vanya Agrawal

SUMMARY

Astrophysicists constantly seek methods of effectively conveying complex concepts to non-scientific audiences. Sonification, or the conversion of information into audio, improves accessibility of the universe and enhances pattern recognition, data retention, and error detection. However, most studies sonify single events and sound musically discordant. This project sonifies multiparametric supernova data from Palomar Observatory into a musically pleasing and mathematically precise composition, exploring innovative composition mediums to engage audiences.

Data of 8000+ supernovae collected by the Zwicky Transient Facility were manipulated within Python. Each parameter was equated to a musical characteristic (e.g. supernova peak luminosity to volume) and normalized while maintaining core proportionalities. The values were exported as MIDI files into Garageband, where each supernova type was assigned an orchestral instrument. Finally, rhythm, harmony, soundscapes, and mixing were applied to transform the piece into a euphonious composition.

The composition meets the scientific objective as the data are distinguishable, with modifications supporting the patterns and features. It also meets the musical objectives as it resembles lo-fi and ambient music, which are pleasing and functional, and emulates the "sound of space" associated with sci-fi soundtracks.

The project successfully sonifies the most comprehensive supernova data to advance supernova studies. By conveying accessible, recognizable, and comprehensible information, this sonification may broaden both artistic and scientific audiences. Future research should develop an automated sonification algorithm with sonifications streaming in real time. This could also contribute to a Virtual Reality universe to simultaneously observe and listen to the cosmos, elevating the relevance of astronomy in commercial media.

INTRODUCTION

Each planet, hurtling through space at thousands of miles per hour, vibrates quickly to produce a faint humming sound. Sound waves from billions of objects in the universe converge on Earth, creating a cacophony of noise imperceptible to human ears. Musica Universalis, or "Universal Music", Pythagoras called it. The ancient Greek philosopher and physicist theorized that since every celestial body moves, it must produce sound, and humans are only unaware of this phenomenon because they have no distinction between the celestial sound and silence (Samuel, n.d.). Although fascinating, astronomers have long since discredited this theory, yet Pythagoras' discoveries have inspired numerous studies crossing astrophysics and music. In modern astronomy, researchers utilize music to supplement and enhance data interpretation, and in modern music, composers base works and performances on extraterrestrial phenomena. The technological interconnection of these two fields has induced creative innovation and enriched the understanding of both.

Data Sonification

Sonification, or the process of converting information or data into non-speech audio, is a notable convergence between science and music (see Appendix A). It is an important form of data interpretation practiced by scientists, as it can enable them to perceive patterns and irregularities in the data (Crockett, 2019). Dr.

Gregory Kramer – a composer, entrepreneur, teacher, and pioneer in auditory research – describes an application of sonification in geo-seismology in which researchers successfully analyzed earthquake activity and detected instrument-induced error (Kramer, 1994). Another experiment demonstrating sonification's benefits was conducted by Dr. Robert Alexander, the CEO and co-founder of Auralab Technologies and a sonification specialist. His "Think-Aloud Protocol" study discovered that sonification accentuates spectral features that may otherwise go overlooked in data sets and enables listeners to consistently identify wave patterns (Alexander, 2014). As both Alexander's and Kramer's studies illustrate, audification methods enhance research by improving pattern recognition, data retention, and error detection.

In addition to aiding researchers, sonifications help convey information to a non-scientific community. Dr. Alex Parker is a postdoctoral fellow at the Harvard-Smithsonian Center for Astrophysics and was part of NASA's New Horizons spacecraft team. Describing his project Supernova Sonata, he states that sonifications allow audiences to re-interpret mathematical data sets in a more palatable way (Schiller, 2012). Scientists are constantly seeking to increase outreach and bring complex scientific concepts within reach of the general public, and sound patterns connect with audiences better than data. For people with visual impairments in particular, sonifications increase accessibility to science. In 2020, NASA's Chandra X-ray Center converted space images of nebulae and supernovae (SNe), or star explosions, into music by assigning visual characteristics to sound characteristics (see Appendix A; see Appendix B, Link 1). Christine Malec, a visually impaired musician and astronomer on the project, quotes, "Music affords data a spatial quality that astronomical phenomena have, but that words can't quite convey,'" (Temming, 2020). As such, sonifications are valued by the astronomy community for improving the accessibility and comprehensibility of the universe.

Sonification research could also be crucial for scientific progress by improving research funding. Dr. Derek Muller, an expert on designing multimedia for physics education, has experienced the effects of falsified communication in the scientific industry. In his successful YouTube channel Veritasium, he describes that scientists must increasingly rely on public attention for their work to acquire research funding (Muller, 2023). As a result, science has become dependent on press releases and headlines for its continued study, so overstated results that draw attention often fill news sources (Muller, 2023). Sonifications can break through the sea of distorted information by presenting accurate reinterpretations of data in attention-grabbing, digestible ways. If scientists maximize art for public outreach, their work will ethically expand audiences and scientific transparency. In accordance with both Parker's and Muller's findings, interdisciplinary methods can significantly improve the quality and magnitude of audience consumption.

Major Astrophysics Sonification Studies

Researchers have utilized various methods to create audifications of astrophysical phenomena. For this research, qualitative sonification is defined as the indirect translation of data using general patterns (see Appendix A). This composition is musically pleasing but has been interpreted beyond scientific significance. Contrastingly, quantitative sonification is defined as the direct translation of data without additional modifications (see Appendix A). This composition conveys data accurately but does not evoke an emotional response. Combining the two is a composed sonification, which is defined as a direct translation that modifies the piece with qualitative features and compositional techniques (see Appendix A). This composition is both euphonious and scientifically relevant. The previously mentioned Dr. Robert Alexander converted data from NASA's ACE satellite, which measures solar wind, into audio (see Appendix B, Link 2). Alexander incorporates his musical background by qualitatively reflecting every feature of the data within the music. For example, particle surges are translated as crescendos, temperature changes as pitch, and rotation rate as rhythm (Crockett, 2019). The instrument choices evoke the feel of the sun's heat, density, and velocity. Whereas quantitative sonifications may be more beneficial for researchers, qualitative sonifications are often more musical and emotive in nature.

The musicality of qualitative methodology is further illustrated by Dr. Alex Parker's sonification, briefly discussed earlier (see Appendix B, Link 3). His Supernova Sonata sonified 241 Type Ia SNe, which occur in binary – two-star – systems where one star is a white dwarf. This data set limited Parker's instrumentation to only bass and piano (Parker, 2011). Parker describes that most audification projects create purely quantitative translations that do not allow easy listening. His sonification aims to produce a more fun and aesthetic product by restricting notes to modes – scales that give notes context and evoke emotions (see Appendix A). Parker's experiment, however, was conducted over ten years ago as a hobby rather than a formal study, so there are currently no published papers on the subject (Schiller, 2012). Alexander's and Parker's projects, which are realized through coding softwares and digital audio interfaces (DAWs), are innovations of computerized sonification.

Contrastingly, Dr. Valeriya Korol, an astrophysicist at the University of Birmingham, and Andrea Valle, a sound programming professor at the University of Torino, developed an algorithm to sonify data for live performers. Their composition, titled Einstein's Sonata, translated gravitational wave data for a future Laser Interferometer Space Antenna (LISA) space mission (see Appendix B, Link 4). The objective was to provide an artistic rendition of the LISA mission using quantitative methods and create music that would contribute to the cultural heritage of both the scientific and art communities (Valle & Korol, 2022). As gravitational waves cannot be seen or felt, sound is a powerful tool for conveying and perceiving their behavior. Sonification technology creates unique music rooted in the patterns of the universe, a method of composition to be further explored in both computerized and performance music.

Aesthetics of Music

In modern music, certain elements and features are commonly recognized as pleasing. Parker's use of a predetermined mode is one example of compositional modification, which this project defines as any technique implemented to improve musicality (see Appendix A). Expressiveness is another characteristic that attracts listeners. Dr. Kate Hevner is a celebrated 20th-century psychologist who studied emotional and physical responses to music. She determined from self-reports by music listeners that many analyze their emotional responses to music, and this introspection creates an appreciation for the piece (Hevner, 1936). She also created an adjective circle to determine the effects of harmony, asking research participants to point to adjectives that described their reactions to certain chords. Her findings illustrate that dissonant, minor harmonies create sad or yearning reactions while consonant harmonies produce a happy or serene response (see Appendix A). A quiet, repetitive, simple rhythm will create a more soothing and hypnotic response whereas a complex, loud pattern will cause an excited or anxious reaction, explains Wolfgang Hans Martin Stefani in his master's thesis (Stefani, 1981). Sci-fi films also create expressiveness through electronic instruments. Beginning with the theremin, the first electronic instrument used in film, electronic sounds became associated with unsettling, tense emotions. Lisa Schmidt, a Ph.D. candidate at the University of Texas describes that large, sweeping, atonal sounds grew to represent something otherworldly (Bartkowiak, 2010). Aside from melody, the harmonies, rhythm, and electronic sounds can contribute to enjoyable music.

Project Description

Supernova (SN) data from the past five years collected by Palomar Observatory's Zwicky Transient Facility (ZTF) are a strong candidate for this sonification technology. Parker's Supernova Sonata is the only known sonification on a collection of SN points, but as it was not formally published, this will be the first official study on the subject. Technological advancements since Parker's study in 2012 have allowed researchers to detect more SNe – 8000 versus 241 – from a more substantial fraction of the night sky – 1/2 versus 1/10,000 (Zwicky Transient Facility, 2023). While Parker's piece sonified only SN Ia, the ZTF captures every extraterrestrial transient, or event, so all SN types will occur in the data at approximately the same frequency they occur in space (Zwicky Transient Facility, 2023). Each SN type will be assigned an instrument whose timbre, or tone quality, most closely resembles its characteristics (see Appendix A). This sonification aims to fill the gap in SN research, addressing the previous project's limitations with a larger data set, improved technology, more complete data, and refined techniques.

This project's objective is to create an accurate and aesthetic composition that evokes the feeling of space. Its primary purpose is engaging non-expert audiences in SN research through scientifically relevant music that is functional in their daily lives. Using a programming software, raw data provided by researchers at the ZTF will be algorithmically manipulated into MIDI notes. Then, within a Digital Audio Interface (DAW), the MIDI files will be compositionally modified and produced. Instead of melody, this project will create music through the background elements of chords, rhythms, and soundscapes. The music will be heavily inspired by ambient, sci-fi, and lo-fi music, as they meet the aforementioned criteria for pleasing music.

Sonification technology condenses complex information into music. This allows audiences to perceive the universe through sound and evokes new sensory interpretations of astronomical phenomena. This project will make astronomy and SN data mapping accessible to the public, increase engagement with SN research, and create scientifically relevant music. By utilizing methods inspired by previous papers, the process will demonstrate the rate of occurrence of SN characteristics for scientists, including type, distance, and luminosity. The sonification will also be beneficial for the ZTF's publications and presentations, as it forms relatable, attention-grabbing media by which to present their research to the public.

METHOD (CODING)

The data being analyzed was collected by researchers at Palomar Observatory using the Bright Transient Survey (BTS), the largest spectroscopic SN survey to date. Every 2-3 nights, the ZTF surveys the entire northern hemisphere, acquiring spectra of almost every time-dependent, stationary object brighter than 18.5 mag. It currently holds a 95% completeness for extragalactic transients with the remaining 5% due to weather fluctuations. Between March 2018 and September 2023, the researchers captured and categorized over 8300 events, which have already been reviewed for errors and polished by a qualified team (Caltech, 2023). This ensures the data are the most accurate, up-to-date, and comprehensive representation in current SN studies (Figure 1.1).

Serving as a mentor was Dr. Christoffer Fremling, a post-doctoral researcher at Caltech and head scientist at the ZTF. While visiting Caltech, the author had the opportunity to meet with an astrophysics professor, Dr. Mansi Kasliwal. After discussing the research idea with her, the author was connected with her student Dr. Fremling to work with ZTF's data. Fremling provided the data, described the findings, and answered some coding questions (see Appendix C, Figure 1.2). Below are the relevant measured characteristics used for each point:

- Discdate (date of discovery)
- Peakmag (brightness at peak in reverse logarithmic magnitude scale)
- Redshift (distance to the SN)
- Duration (how long the event could be detected by the telescope)
- Type column (SN type)

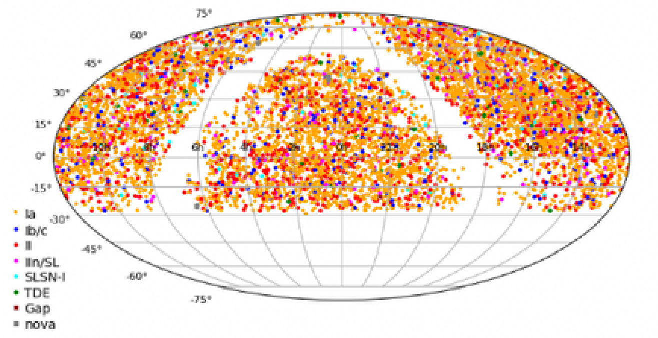


Figure 1.1 Visualization of all extragalactic transients captured by the ZTF and their categorization

Importing Data

Google Colaboratory (Colab) was chosen as the programming software with ChatGPT as a supplement. Colab codes in Python, which is frequently used among scientists, developers, and researchers, and is one of the most effective free computing programs. As someone with little coding experience, ChatGPT was also consulted. Although researchers do not recommend relying on ChatGPT for professional coding, they concede it is effective when given specific, short, detailed prompts (see Appendix C, Figure 2.1). Since the objective of the project is geared toward data manipulation and composition, not coding itself, consulting ChatGPT does not interfere with the stated purpose. When a code fulfilled the desired purpose, it was considered reliable (see Appendix C, Figure 2.2).

The data file was first converted from a .txt file to a .csv file (see Appendix C, Figure 2.3). It was then imported as a data table with the first few rows displayed to verify the code was functioning. Since this process exports one MIDI file, it must be filtered by SN type to create different instrument sounds (see Appendix C, Figure 2.4). The resulting table contained 22 columns and 5 displayed rows, but this project only considers discdate, peakmag, duration, type, and redshift (see Appendix A; see Appendix C, Figure 2.5).

Timestamp

Discdate was converted into the timestamp of each note. The discdate was first changed into a datetime object so its values could be easily manipulated, then into seconds from the start of the audio. This new column was named Timestamp (see Appendix C, Figure 3.1).

Timestamp was then normalized, or manipulated, to fit within a desired range of 600 seconds (10-minute long piece). The values were shifted down by 30 to remove a lower outlier (see Appendix C, Figure 3.2). A histogram was created that demonstrated distribution, ranges, and notes per interval to confirm normalization was occurring successfully (see Appendix C, Figure 3.3, Figure 3.4, Figure 3.5).

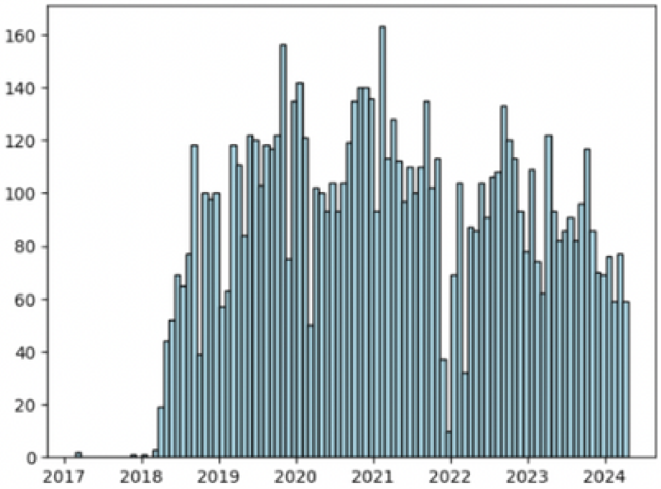


Figure 3.5 Timestamp histogram. Intervals of 20 ranging from -30 to 600. Only values between 0 and 600 were used

Pitch

Redshift was converted into the note pitch. MIDI values range from 0 to 127, or the notes C-2 to G9. But notes at the high and low ends are difficult to hear or distinguish, so the range was limited to the MIDI range 36 to 96, or the notes C2 to C7 (Inspire Acoustics, n.d.). The redshift column was changed into numeric values (see Appendix C, Figure 4.1). Using the same normalization method as discdate, the values were manipulated into the desired range and a histogram was created (see Appendix C, Figure 4.2). Upper outliers skewing the data were removed, and the new column was named Pitch (see Appendix C, Figure 4.3, Figure 4.4).

As in Parker's project, a mode was predetermined. C lydian augmented (+lyd) mode has the notes C D E F# G# A B C (see Appendix A). Whole-tone and lydian scales are known for having an ethereal, space-like quality, largely attributed to the augmented fourth also contained in +lyd scales (see Appendix A). Whole-tone scales are less settled and rooted, which can be beneficial for representing the vastness of space, however, lydian scales create more interesting and consonant harmonies, one of the previously established criteria for serenity. The +lyd mode utilizes the augmented fifth and atonality of the whole-tone scale (characteristic of space music), and the natural sixth and seventh of the lydian mode (characteristic of uplifting, harmonious sounds).

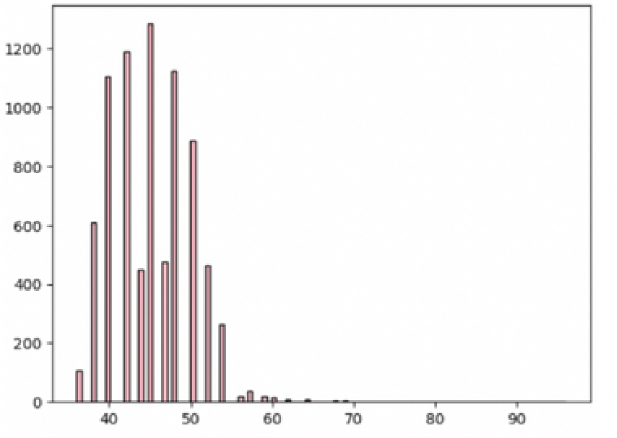


Figure 4.6 Histogram of pitches adjusted to C +lyd mode

Volume

Peakmag was converted into the volume of each note. Peakmag is the reverse logarithmic measure of peak brightness, meaning higher magnitudes correspond to fainter objects (see Appendix A). For the best musical translation, brighter objects should be reflected by louder volumes. Conversely, peakflux describes the maximum energy in SN explosions (see Appendix A). Using peakflux instead, higher brightness would correspond with higher volumes, aligning with the research goals of creating scientifically and sensorily accurate music. In Colab, peakmag values were transformed into peakflux values, defining the zero point as peakmag's max value (see Appendix C, Figure 5.1, Figure 5.2).

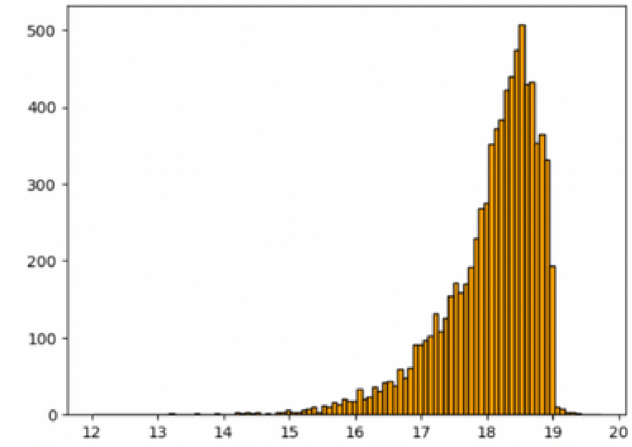


Figure 5.2 Histogram of peakmag values

The same normalization and histogram creation process was implemented. MIDI volume values range from 0-127, but anything under 10 becomes unintelligible and over 100 becomes distorted, so the range was limited to 10-100. This new column was named Volume (see Appendix C, Figure 5.3, Figure 5.4, Figure 5.5).

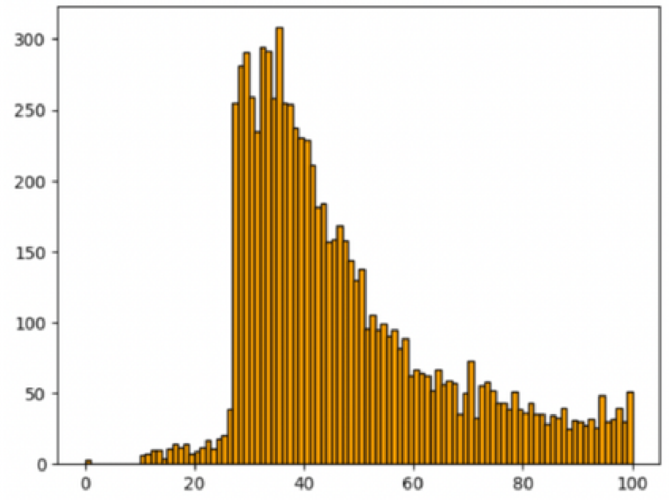


Figure 5.5 Volume histogram

Duration

Duration, or the amount of time each SN was detected for, was converted into the duration of each note. The provided data included the character “>” before each value, which Python is unable to interpret, so the character was removed (see Appendix C, Figure 6.1). All duration values of 0 were also removed as 0 duration results in no sound (see Appendix C, Figure 6.2). The same normalization and histogram functions were used, with the range set to 0.1-1 so notes will not sustain and create heavy dissonance (see Appendix C, Figure 6.3, Figure 6.4, Figure 6.5).

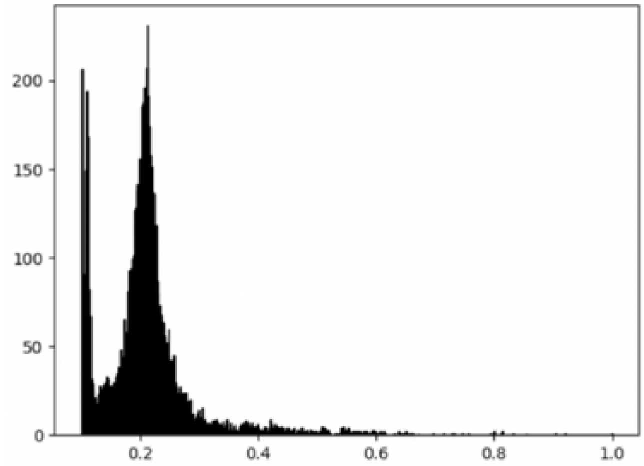


Figure 6.5 Duration histogram

Creating MIDI

A MIDI file was created with one track added to it. The table was sorted by Timestamp in ascending order and the negative values were filtered out (see Appendix C, Figure 7.1). Next, all MIDI values were defined and rounded to integers. Duration and Timestamp were converted to milliseconds, and since Python reads time as the difference between events, time_delta was defined as Timestamp - prev_time. Values were transferred into MIDI, with note on/off (duration) being based on time_delta and duration. Finally, the file was named and exported into the Colab files, which were then downloaded (see Appendix C, Figure 7.2). This process was repeated for each SN type by changing the type being filtered (refer to section: Importing Data) and rerunning all following codes. A new MIDI file was created and saved for each type: SN Ia, SN Ib, SN Ic, SN II, SN IIn, SN IIP, SLSN-I, and SLSN-II (see Appendix A).

METHOD (COMPOSING)

Instrumentation

Each of the seven downloaded MIDI files was uploaded into a Garageband MIDI track, as Garageband is a powerful, free, well-respected DAW for music creation (see Appendix C, Figure 8.1).

Tracks were then assigned instruments based on several criteria (see Appendix D):

- Range of the notes – instruments ranges matched track ranges
- Note length – short durations are played by plucked instruments, long durations by bowed or wind instruments, and mixed durations by instruments with both capabilities
- Note frequency – frequent notes are played by softer instruments to avoid cluttering the sound, rare notes are played by louder instruments that are easily distinguishable
- Combinations – instrument choices were logical in the context of other instruments, as certain timbres complement each other
- Distinction – instruments had dissimilar timbres to distinguish SN types
- Acoustic – instruments used only orchestral or acoustic instruments (electric sounds saved for compositional modification)
- Similar type – similar SN types were played by similar instruments
- Timbre – instrument timbres were qualitatively representative of their SN type

Technical Modifications

At the set tempo, the notes created a cacophony and were indistinguishable. Through trial and error, the range of 70-110 bpm was determined as an acceptable compromise between clarity and piece length. Notes held longer than five measures that sounded particularly discordant were either shifted by a step or removed. Distorted notes at the upper or lower ends of an instrument's range were also removed or shifted by an octave. Sustained notes with overpowering volumes were quieted. Finally, instruments that are naturally louder in MIDI, such as the fugue organ, received a general volume reduction.

Rhythm

The rhythms, as described earlier, were to be simple, repetitive patterns in a lo-fi style. To create the drum track, the R&B drummer Rose and Latin percussionist Isabela from the library were imported (see Appendix D). The time signature was changed to 6/8 to deviate from standard 4/4 lo-fi music. Rose's drumming style was varied to add dynamic characteristics. This was accompanied by soft Latin percussion, primarily utilizing shakers, cabassa, and congos. At the piece's climax, the gong, cymbal, and mark tree on the orchestral kit were used to create dramatic effect (see Appendix D). Lulls, or periods of low activity in the MIDI data, were highlighted by reducing or removing rhythm, a common technique in lo-fi music to break rhythmic monotony (see Appendix A; see Appendix C, Figure 9.1). The sections with minimal rhythm and data were instead filled with chords and soundscapes.

Figure 9.1 Garageband MIDI of Rhythm Modifications

Harmonization

The entire piece had a slow harmonic rhythm, or frequency of chord changes, to avoid overloading senses. Harmonizing a fast, complex chord progression to quick and inconsistently spaced data would create confusion for listeners (see Appendix A). The harmonic rhythm was increased during lulls in the data. Progressions were determined by fitting chords to the sustained notes in the data. They were also generated on a website called OneMotion, which lists all possible chords within a mode and allows users to form custom chord progressions (Eriksson, n.d.). Progressions that made musical sense were then manipulated into synthesizers and soundscapes. The "space feel" is often characterized by seventh chords, slight dissonance, and unresolved progressions (those that do not return to a harmonious chord). With these in mind, an example progression generated in OneMotion was Cmaj7#5 Bm7 Dsus2. Chords were assigned to the Outer Lands Synth, and the sounds were modified with ambiance, sustain, and reverb (see Appendix D).

Soundscapes

Soundscapes were chosen to form a space ambience. By selecting only electronic sounds for modifications and orchestral sounds for the data, a distinction in timbre was formed between the two. The soundscapes included effects such as wind and sweeps, sustained pitches, synthesizers, and eerie sounds (see Appendix D). Dreamy, airy soundscapes, whose pitches followed chords, faded in and out at constant intervals throughout the piece to create depth (see Appendix C, Figure 10.1, Figure 10.2). More distinctive effects that do not blend with other soundscapes faded in only during lulls (see Appendix D). As the piece progressed, the instrumentation, tempo, and volume increased and the sounds grew more electronic and eerie. After the climax of the piece, soundscapes and rhythm cut out, leaving only the data. They returned in an imitation of the beginning of the piece, creating a sense of continuity as the audience can detect chords and patterns they have previously heard (see Appendix C, Figure 10.3).

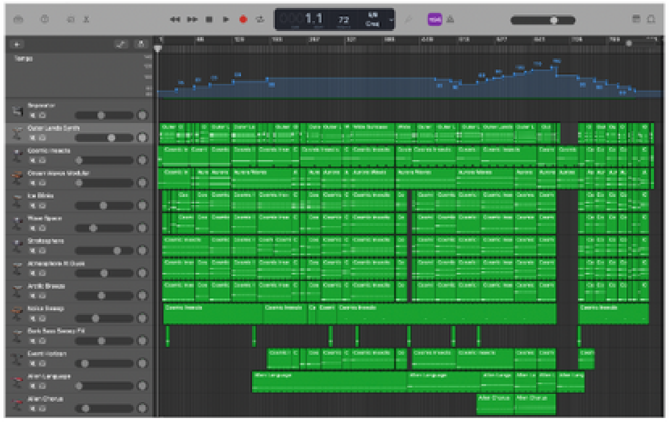


Figure 10.3 Garageband MIDI of Soundscape Modifications

Mixing

As described, automation was used on rhythms, chords, and soundscapes to create fading effects. Next, panning was used to adjust instrument resonance in each ear. To create a greater distinction between data tracks, all SN I were panned to the right, SN II to the left, and SLSN split. The busiest tracks (SN Ia and SN II) were panned to the maximum in opposite directions so their frequent notes would not blend. Other tracks were panned to varying degrees, arbitrarily decided. These reduce clutter by allowing each ear fewer inputs (see Appendix C, Figure 11.1). Synth and soundscape tracks were duplicated and panned to the max in opposite directions, a common mixing technique for creating more realistic sounds. Panning instruments in this way forms an illusion of dimension that is associated with space. Finally, audio adjustments were made by increasing master reverb to 50% and master echo to 25% on each track, with minor variations. Equalizers also reduced unwanted frequencies, improving audio quality. After mixing had been completed, the piece was exported to the disk as an mp3 file in the highest quality.

RESULTS

The piece was exported as two separate tracks: the raw MIDI data without modifications and the complete composition with modifications (see Appendix B, Link 6, Link 7). The complete code can also be found (see Appendix B, Link 5). The unmodified version includes only data directly converted to MIDI and adjusted to the assigned mode. On the other hand, the complete composition keeps data distinguishable while the compositional modifications improve musicality, meeting the project goals. Several features in the piece ensure scientific relevance: the data comprise the entire melody and utilize acoustic instruments, distinguishing them from modifications; soundscapes, chords, and rhythms undertone data patterns by highlighting changes; and enhanced musicality aids in the comprehension of data. However, despite ensuring the complete composition will strengthen scientific analysis, scientists may find the additional modifications distracting when listening for patterns or irregularities in the data. To resolve this, the data-only track was created. The piece's intended length was also 10 minutes, but because the data were more condensed than expected, the final length was stretched to 30 minutes.

As a composition, the piece resembles lo-fi, sci-fi, and ambient music. The soundscapes and data sound like ambient music, which is based on atmosphere and texture over rhythm and melody. The rhythms and chords sound like lo-fi music, which uses open harmonies and simple drum patterns. The emotions mimic sci-fi music through electronic and distorted soundscapes, chirpy sound effects, mixing strategies, and chord choice. Since these styles have similar functions as study, background, or focus music, this piece is functional for listeners, meeting the objective of introducing scientific concepts and data to everyday life. The constant influx of sound, lack of clear structure, and surrounding effect from the panning may be overwhelming for some, but it may be soothing to others in passive listening, as it mimics music used for sleeping or studying. Since the piece is still musically pleasing and functional, it achieves the project's musical goals.

Another project goal was to emulate the atmosphere of space. This was heard in the sci-fi resemblance and in the loose structure. The piece was organized to follow the life cycle of a star: stellar nebula to 8:22, main sequence massive star to 19:50, red supergiant to 23:38, SN to 23:51, and back to stellar nebula to 28:12. Tempo changes, sound effects, and increasing instrumentation highlight these changes. This subtle addition adds structure to data that does not contain intrinsic structure, improving musicality and listenability.

DISCUSSION

In practical use, the data-only track may be analyzed by researchers at Palomar Observatory and Caltech for pattern and error detection. The ZTF holds the most comprehensive SN data to date, and as sonification facilitates a new medium to observe astrophysical phenomena, this track advances modern SN research. The modified track, on the other hand, will be more beneficial for Palomar's public data releases and presentations, bringing non-expert audiences closer to sophisticated astronomy studies. Music offers a spatial perspective of the universe that cannot be quantified by data or visuals. This musical medium may therefore create a more complete understanding of the universe for both researchers and public audiences. The results confirm that the project has built on previous sonification studies with unique approaches that combine aesthetics and accuracy. Compared to the quantitative sonifications Einstein's Sonata and Parker's

Supernova Sonata, this sonification is more coherent and palatable to listeners (Parker, 2011; Valle & Korol, 2022). Compared to Alexander's qualitative solar wind sonification, this piece is more numerically accurate and beneficial to researchers (Crockett, 2019). As this project successfully created a piece that blends qualitative and quantitative methods with compositional modification to create a composed sonification, it has achieved the objective.

However, one difficulty encountered was that data created clutter in Garageband, as notes were extremely dense. There were too many notes simultaneously to apply chord progressions. To address this, the composition was expanded from 10 minutes to 30 minutes, reducing the density of the data. But this created too much music to compose beyond rhythm, slow-moving chords, and consistent soundscapes. Overall, the project could have been improved with a smaller data set, allowing for more modifications. Additionally, this is a sonification of measured values, which may vary from the actual behavior of SNe. The most significant problem was discovered post-experimentation. The method of quantifying "time-on" and "time-off" values in the code may have created inconsistencies within the result, as parts of the sonification did not line up with the video representation of the data created by Dr. Fremling. The problem was addressed and a new 2-minute sonification was created on just 1000 points, using the same composition methods. This new piece is not linked in the Appendix, as it is to be presented to the public for the first time in May 2024. For the context of this study, only the original 30-minute composition will be considered as the result.

Along with Caltech's publication of this research in May 2024, the author is working in collaboration with researchers at Palomar Observatory to create an accessible, modifiable guide to the public on sonification. Therefore, the project is achieving its purpose of public outreach – further engaging audiences by inviting them to create their own sonifications. This research has also been published on the ZTF YouTube and website and has been presented at district, county, and state science fairs (to be presented at international), reaching broader audiences. Future research should sonify different types of data to explore how musical style and emotive effect could be varied given new data sets. Sonification could become a standardized aspect of all future astronomy publications, forming both a larger audience base and a more comprehensive auditory universe. Scientists should also develop automated algorithms, where a person does not have to manually input chords and rhythms. This has the benefit of producing sonifications in real time that could be published on streaming services. Automated algorithms may also contribute to a virtual reality universe or planetarium, where people can simultaneously observe and listen to each part of the universe. Such projects would introduce sonification into commercial media, allowing astronomy studies to be relevant in daily life.

The project suggests that sonification as a medium for electronic music composition should be further developed, as it can create unique music based on science. These sonification methods utilize both algorithmic and impressionistic compositional techniques, elevating music as a technological and humanistic experience. This is particularly relevant in modern society where artists must learn to cooperate with AI rather than work against it. As shown, integrating technology with music and astrophysics mutually benefits both fields. By conveying scientific information through a platform that is accessible, recognizable, and comprehensible, composed sonifications can attract those who otherwise may not have been interested in scientific research and improve research funding. Music with scientific value has the power to merge audiences and inspire scientifically curious artists, artistically inclined scientists, and intellectually motivated individuals.

APPENDIX

Appendix A (Vocabulary Terms)

Augmented Fourth Interval: distance of a fourth that has been raised by one half-step (e.g. C to F# instead of C to F). Sounds dissonant when played in unison but can create a feeling of upward motion when played in the context of a mode, such as lydian or whole-tone

Chord Progression: sequence of chords centered around a key that form the basis of harmony

Composed Sonification: direct translation that modifies the piece with qualitative features and compositional techniques. This composition would theoretically be both euphonious and scientifically relevant.

Compositional Modification: any technique implemented to enhance musicality

Dissonance: discordant, nonharmonic notes or note combinations in music. May make listeners feel uneasy, but can create satisfying tension when resolved effectively. The opposite of consonance, or harmonious notes

Lull: a period of low activity in the rhythm or MIDI playback of the data

Lydian Augmented (+lyd) Mode: major scale with a raised fourth and fifth scale degree, or a lydian mode with an augmented fifth scale degree

Mode: a set, scale-like pattern of pitches starting on any note that compositions generally follow. Preserve general patterns, such as relative pitch, but give notes context to evoke emotions

Peakflux: peak measure of the amount of light produced by a star each second, or the maximum energy involved in a supernova explosion. High flux values correspond with high apparent brightness

Peakmag: peak apparent magnitude of supernova brightness. Reverse logarithmic measure of luminosity, meaning low magnitudes relate to high apparent brightness

Qualitative Sonification: indirect translation of data using general patterns. This type of composition is meant to be musically pleasing and evoke the feel of a phenomenon but has been interpreted beyond scientific significance.

Quantitative Sonification: direct translation of data without additional modifications. This type of audification is meant to convey data accurately but does necessarily evoke an emotional response or sound pleasant.

Redshift: objects moving at high velocities scrunch the waves they emit so waves are more frequent for objects traveling toward a viewer and less frequent for objects traveling away (Doppler Effect). As the universe expands, distant objects move away from Earth creating stretched wavelengths, known as redshift. Used to measure the distance of a moving object from Earth

Sonification: the process of converting information or data into non-speech audio

Supernova (SN or SNe): last stage in the evolution of a large star, where it creates a tremendous explosion near the end of its life. When a star uses up the last of its nuclear fuel, it collapses inwards from the force of gravity, triggering a massive release of energy

Timbre: tone color or sound quality of a musical note. Unique for every instrument and distinguishes different types of sound

SN I: supernova explosion that leaves behind no hydrogen (The Schools' Observatory, n.d.)

SN Ia: supernova in a binary (two-star) system where one star is a white dwarf. The white dwarf's gravity pulls matter from the other star until the white dwarf becomes massive and unstable enough to explode (The Schools' Observatory, n.d.)

SN Ib/Ic: supernova where a star is so massive it loses its outer layers, which contain most of the hydrogen, but explode as normal supernovae. Ic have lost both hydrogen and helium layers while Ib have lost hydrogen but may retain some helium (The Schools' Observatory, n.d.)

SN IIn/IIP: supernova where a massive star (8-200 times the mass of the sun) burns for only a few million years, much less time than low mass stars, before collapsing and exploding. IIn categorizes narrow hydrogen spectra, and IIP describes the P shape of the light curve (The Schools' Observatory, n.d.)

SLSN: super-luminous supernova with a luminosity 10x greater than standard supernovae. SLSN-I classifies events that produce little hydrogen, in contrast with SLSN: super-luminous supernova with a luminosity 10x greater than standard supernovae. SLSN-I classifies events that produce little hydrogen, in contrast with SLSN-II which classifies events that produce large amounts of hydrogen (The Schools' Observatory, n.d.)

Appendix B (Links)

(Link 1) Conversion of nebula and supernova images into music by NASA's Chandra X-ray Center:
https://news.harvard.edu/gazette/story/2021/01/harvard-scientist-turns-space-images-into-music/

(Link 2) Qualitative sonification of the NASA ACE satellite’s solar wind data: https://www.youtube.com/watch?v=2b9ykhYzw6k

(Link 3) “Supernova Sonata” sonification on Type Ia supernovae: https://vimeo.com/23927216

(Link 4) “Einstein’s Sonata” sonification on translational wave data for a LISA mission: https://www.valeriakorol.com/art-science/einsteins-sonata

(Link 5) Complete code in Colab, including all graphs and tables: https://colab.research.google.com/drive/1-8_uqABXL_Ik_nX7-WxmU8Imyimx7kYB?usp=sharing

(Link 6) Unmodified sonification from Garageband as an mp3 file (begins after 20 seconds): https://drive.google.com/file/d/1Bi6X02OrVaWRbSdiFDs8mBBqqErU_PtP/view?usp=sharing

(Link 7) Complete sonification from Garageband as an mp3 file : https://drive.google.com/file/d/1E3goaBu_b3Q98_B3chf9AaBuENBb9Ms4/view?usp=sharing

Appendix C (Figures)

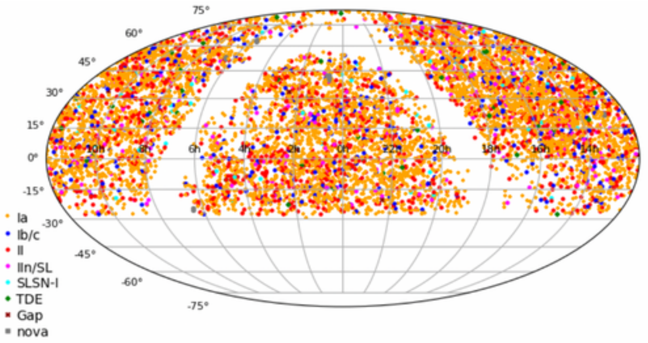
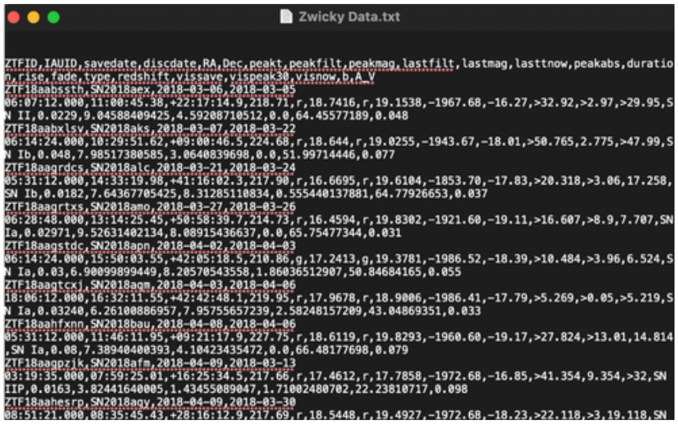
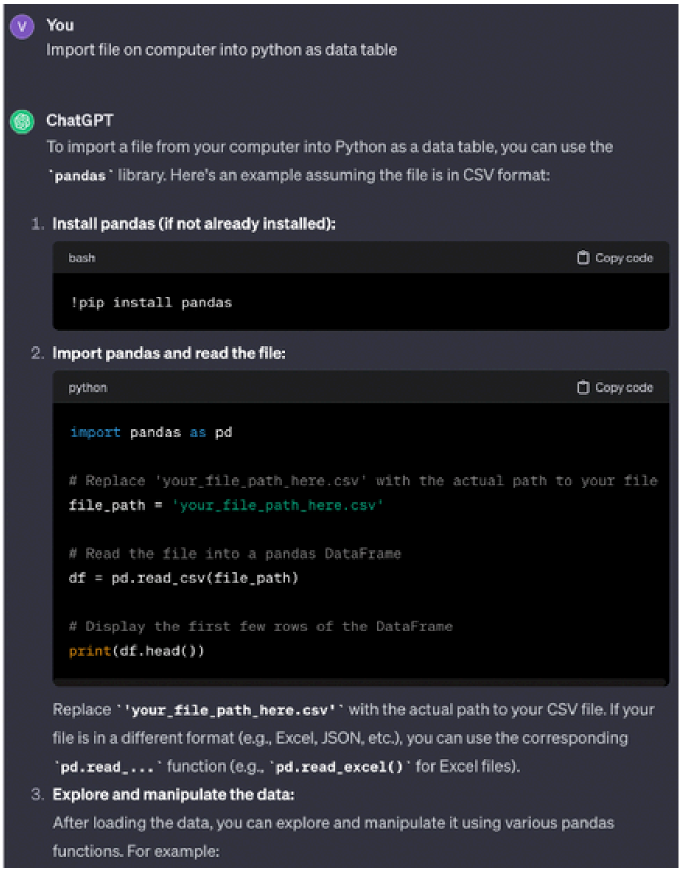


Figure 1.1 Visualization of all extragalactic transients captured by the ZTF and their categorization



```
ZTFID,IAUID,savedate,discdate,RA,Dec,peakt,peakfilt,peakmag,lastfilt,lastmag,lasttnow,peakabs,duratio
n,rise,fade,type,redshift,vissave,vispeak30,visnow,b,A_V
ZTF18aabssth,SN2018aex,2018-03-06,2018-03-05
06:07:12.000,11:00:45.38,+22:17:14.9,218.71,r,18.7416,r,19.1538,-1967.68,-16.27,>32.92,>2.97,>29.95,S
N II,0.0229,9.04588409425,4.59208710512,0.0,64.45577189,0.048
ZTF18aabxlsv,SN2018aks,2018-03-07,2018-03-22
06:14:24.000,10:29:51.62,+09:00:46.5,224.68,r,18.644,r,19.0255,-1943.67,-18.01,>50.765,2.775,>47.99,S
N Ib,0.048,7.98517380585,3.0640839698,0.0,51.99714446,0.077
ZTF18aagrdcs,SN2018alc,2018-03-21,2018-03-24
05:31:12.000,14:33:19.98,+41:16:02.3,217.90,r,16.6695,r,19.6104,-1853.70,-17.83,>20.318,>3.06,17.258,
SN Ib,0.0182,7.64367705425,8.31285110834,0.555440137881,64.77926653,0.037
ZTF18aagrtxs,SN2018amo,2018-03-27,2018-03-26
06:28:48.000,13:14:25.45,+50:58:39.7,214.73,r,16.4594,r,19.8302,-1921.60,-19.11,>16.607,>8.9,7.707,SN
Ia,0.02971,9.52631402134,8.08915436637,0.0,65.75477344,0.031
ZTF18aagstdc,SN2018apn,2018-04-02,2018-04-03
06:14:24.000,15:50:03.55,+42:05:18.5,210.86,g,17.2413,g,19.3781,-1986.52,-18.39,>10.484,>3.96,6.524,S
N Ia,0.03,6.90099899449,8.20570543558,1.86036512907,50.84684165,0.055
ZTF18aagtcxj,SN2018agm,2018-04-03,2018-04-06
18:06:12.000,16:32:11.55,+42:42:48.1,219.95,r,17.9678,r,18.9006,-1986.41,-17.79,>5.269,>0.05,>5.219,S
N Ia,0.03240,6.26100886957,7.95755657239,2.58248157209,43.04869351,0.033
ZTF18aahfxnn,SN2018bau,2018-04-08,2018-04-06
05:31:12.000,11:46:11.95,+09:21:17.9,227.75,r,18.6119,r,19.8293,-1960.60,-19.17,>27.824,>13.01,14.814
,SN Ia,0.08,7.38940400393,4.10423435472,0.0,66.48177698,0.079
ZTF18aagpzjk,SN2018afm,2018-04-09,2018-03-13
03:19:35.000,07:59:25.01,+16:25:34.5,217.66,r,17.4612,r,17.7858,-1972.68,-16.85,>41.354,9.354,>32,SN
IIP,0.0163,3.82441640005,1.43455089047,1.71002480702,22.23810717,0.098
ZTF18aahesrp,SN2018agy,2018-04-09,2018-03-30
08:51:21.000,08:35:45.43,+28:16:12.9,217.69,r,18.5448,r,19.4927,-1972.68,-18.23,>22.118,>3,19.118,SN
```

Figure 1.2 Original data received by Dr. Christoffer Fremling in .txt format



You
Import file on computer into python as data table

ChatGPT
To import a file from your computer into Python as a data table, you can use the `pandas` library. Here's an example assuming the file is in CSV format:

1. **Install pandas (if not already installed):**

```bash
!pip install pandas
```

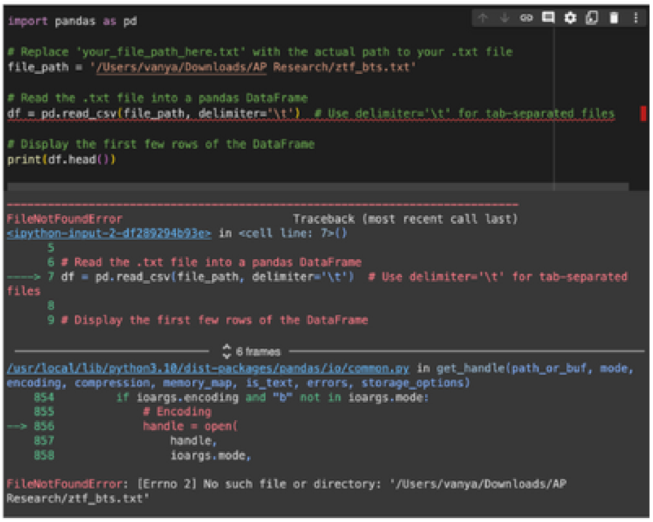
2. **Import pandas and read the file:**

```python
import pandas as pd

# Replace 'your_file_path_here.csv' with the actual path to your file
file_path = 'your_file_path_here.csv'

# Read the file into a pandas DataFrame
df = pd.read_csv(file_path)

# Display the first few rows of the DataFrame
print(df.head())
```

Replace `'your_file_path_here.csv'` with the actual path to your CSV file. If your file is in a different format (e.g., Excel, JSON, etc.), you can use the corresponding `pd.read_...` function (e.g., `pd.read_excel()` for Excel files).

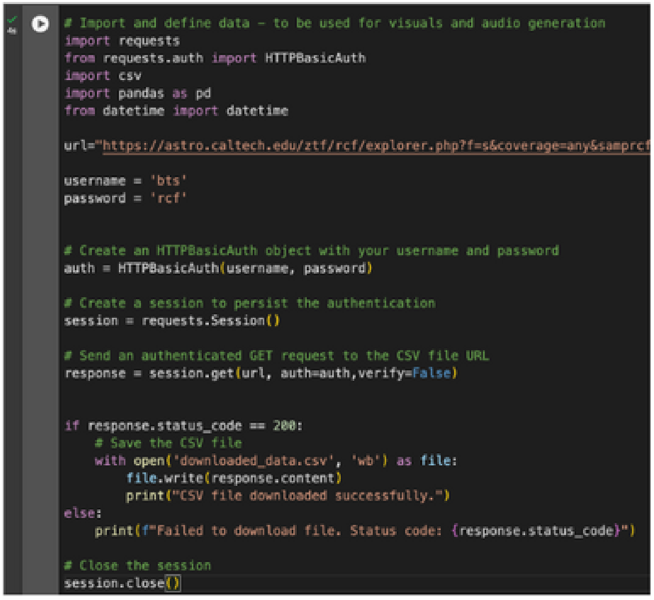
3. **Explore and manipulate the data:**
After loading the data, you can explore and manipulate it using various pandas functions. For example:

Figure 2.1 Prompt to ChatGPT on importing a data file



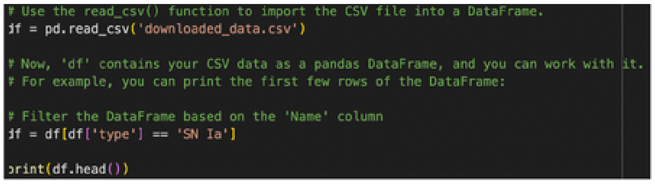
```
import pandas as pd

# Replace 'your_file_path_here.txt' with the actual path to your .txt file
file_path = '/Users/vanya/Downloads/AP Research/ztf_bts.txt'

# Read the .txt file into a pandas DataFrame
df = pd.read_csv(file_path, delimiter='\t')  # Use delimiter='\t' for tab-separated files

# Display the first few rows of the DataFrame
print(df.head())

---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-df289294b93e> in <cell line: 7>()
      5
      6 # Read the .txt file into a pandas DataFrame
----> 7 df = pd.read_csv(file_path, delimiter='\t')  # Use delimiter='\t' for tab-separated
files
      8
      9 # Display the first few rows of the DataFrame

                                  6 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode,
encoding, compression, memory_map, is_text, errors, storage_options)
    854         if ioargs.encoding and "b" not in ioargs.mode:
    855             # Encoding
--> 856             handle = open(
    857                 handle,
    858                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: '/Users/vanya/Downloads/AP
Research/ztf_bts.txt'
```

Figure 2.2 Failed code to import data file into Colab



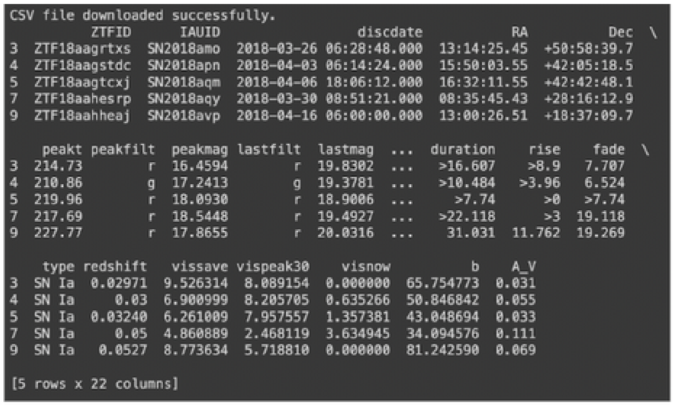
```
# Import and define data - to be used for visuals and audio generation
import requests
from requests.auth import HTTPBasicAuth
import csv
import pandas as pd
from datetime import datetime

url="https://astro.caltech.edu/ztf/rcf/explorer.php?f=s&coverage=any&samprcf

username = 'bts'
password = 'rcf'


# Create an HTTPBasicAuth object with your username and password
auth = HTTPBasicAuth(username, password)

# Create a session to persist the authentication
session = requests.Session()

# Send an authenticated GET request to the CSV file URL
response = session.get(url, auth=auth,verify=False)


if response.status_code == 200:
    # Save the CSV file
    with open('downloaded_data.csv', 'wb') as file:
        file.write(response.content)
        print("CSV file downloaded successfully.")
else:
    print(f"Failed to download file. Status code: {response.status_code}")

# Close the session
session.close()
```

Figure 2.3 Code to convert .txt file to .csv file in Colab



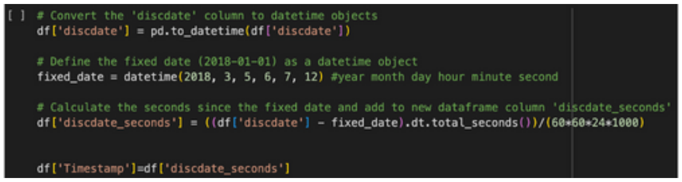
```
# Use the read_csv() function to import the CSV file into a DataFrame.
df = pd.read_csv('downloaded_data.csv')

# Now, 'df' contains your CSV data as a pandas DataFrame, and you can work with it.
# For example, you can print the first few rows of the DataFrame:

# Filter the DataFrame based on the 'Name' column
df = df[df['type'] == 'SN Ia']

print(df.head())
```

Figure 2.4 Code to import file, create data table, and filter rows



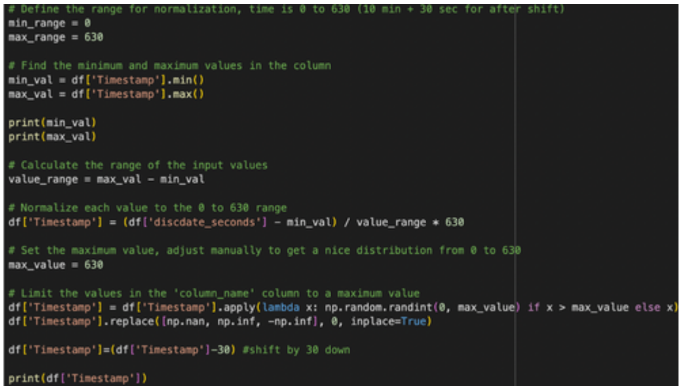
```
CSV file downloaded successfully.
         ZTFID      IAUID                 discdate           RA          Dec  \
3  ZTF18aagrtxs  SN2018amo  2018-03-26 06:28:48.000  13:14:25.45  +50:58:39.7
4  ZTF18aagstdc  SN2018apn  2018-04-03 06:14:24.000  15:50:03.55  +42:05:18.5
5  ZTF18aagtcxj  SN2018aqm  2018-04-06 18:06:12.000  16:32:11.55  +42:42:48.1
7  ZTF18aahesrp  SN2018aqy  2018-03-30 08:51:21.000  08:35:45.43  +28:16:12.9
9  ZTF18aahheaj  SN2018avp  2018-04-16 06:00:00.000  13:00:26.51  +18:37:09.7

    peakt peakfilt  peakmag lastfilt  lastmag  ...  duration    rise    fade  \
3  214.73        r  16.4594        r  19.8302  ...   >16.607    >8.9   7.707
4  210.86        g  17.2413        g  19.3781  ...   >10.484   >3.96   6.524
5  219.96        r  18.0930        r  18.9006  ...     >7.74      >0   >7.74
7  217.69        r  18.5448        r  19.4927  ...   >22.118      >3  19.118
9  227.77        r  17.8655        r  20.0316  ...    31.031  11.762  19.269

    type redshift   vissave vispeak30    visnow          b    A_V
3  SN Ia  0.02971  9.526314  8.089154  0.000000  65.754773  0.031
4  SN Ia     0.03  6.900999  8.205705  0.635266  50.846842  0.055
5  SN Ia  0.03240  6.261009  7.957557  1.357381  43.048694  0.033
7  SN Ia     0.05  4.860889  2.468119  3.634945  34.094576  0.111
9  SN Ia   0.0527  8.773634  5.718810  0.000000  81.242590  0.069

[5 rows x 22 columns]
```

Figure 2.5 First five rows of data table for SN Ia



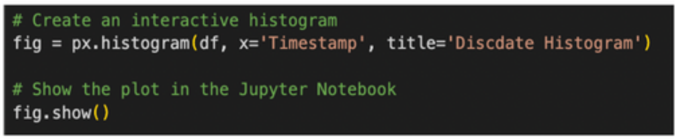
```
[ ] # Convert the 'discdate' column to datetime objects
    df['discdate'] = pd.to_datetime(df['discdate'])

    # Define the fixed date (2018-01-01) as a datetime object
    fixed_date = datetime(2018, 3, 5, 6, 7, 12) #year month day hour minute second

    # Calculate the seconds since the fixed date and add to new dataframe column 'discdate_seconds'
    df['discdate_seconds'] = ((df['discdate'] - fixed_date).dt.total_seconds())/(60*60*24*1000)


    df['Timestamp']=df['discdate_seconds']
```

Figure 3.1 Code to convert discdate into datetime object



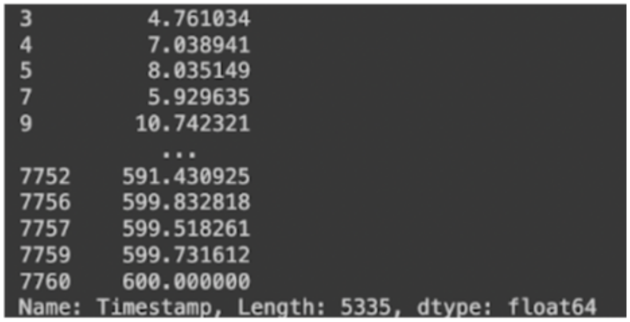
```
# Define the range for normalization, time is 0 to 630 (10 min + 30 sec for after shift)
min_range = 0
max_range = 630

# Find the minimum and maximum values in the column
min_val = df['Timestamp'].min()
max_val = df['Timestamp'].max()

print(min_val)
print(max_val)

# Calculate the range of the input values
value_range = max_val - min_val

# Normalize each value to the 0 to 630 range
df['Timestamp'] = (df['discdate_seconds'] - min_val) / value_range * 630

# Set the maximum value, adjust manually to get a nice distribution from 0 to 630
max_value = 630

# Limit the values in the 'column_name' column to a maximum value
df['Timestamp'] = df['Timestamp'].apply(lambda x: np.random.randint(0, max_value) if x > max_value else x)
df['Timestamp'].replace([np.nan, np.inf, -np.inf], 0, inplace=True)

df['Timestamp']=(df['Timestamp']-30) #shift by 30 down

print(df['Timestamp'])
```

Figure 3.2 Code to normalize Timestamp values



```
# Create an interactive histogram
fig = px.histogram(df, x='Timestamp', title='Discdate Histogram')

# Show the plot in the Jupyter Notebook
fig.show()
```

Figure 3.3 Code to create Timestamp histogram



```
3         4.761034
4         7.038941
5         8.035149
7         5.929635
9        10.742321
           ...
7752    591.430925
7756    599.832818
7757    599.518261
7759    599.731612
7760    600.000000
Name: Timestamp, Length: 5335, dtype: float64
```

Figure 3.4 Timestamp data points. Leftmost numbers indicate SN number

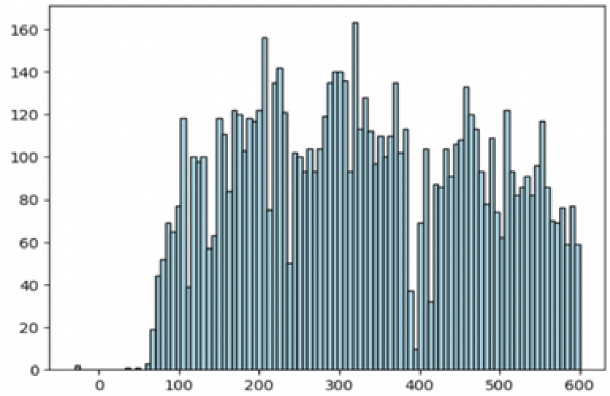


Figure 3.5 Timestamp histogram. Intervals of 20 ranging from -30 to 600

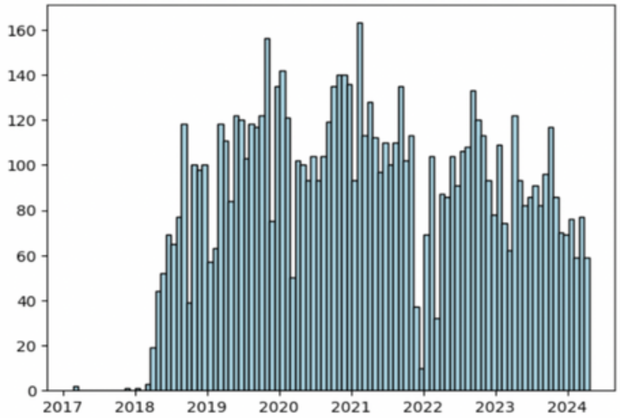
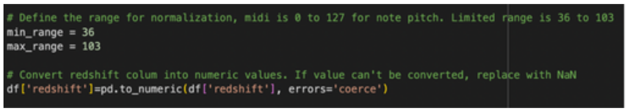
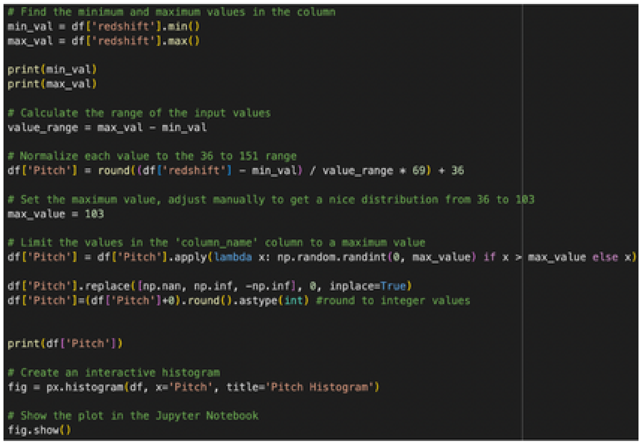


Figure 3.6 Discdate histogram, for comparison of proportionalities



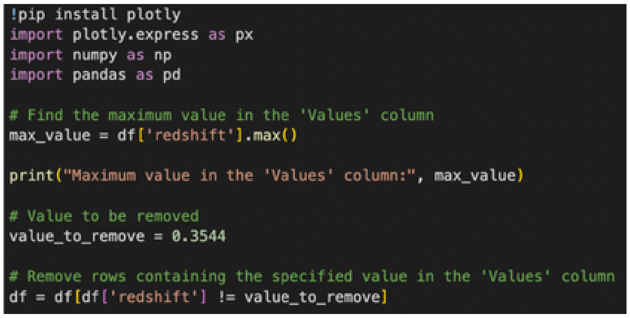
```
# Define the range for normalization, midi is 0 to 127 for note pitch. Limited range is 36 to 103
min_range = 36
max_range = 103

# Convert redshift colum into numeric values. If value can't be converted, replace with NaN
df['redshift']=pd.to_numeric(df['redshift'], errors='coerce')
```

Figure 4.1 Code to set MIDI range for pitch and change redshift into numeric values



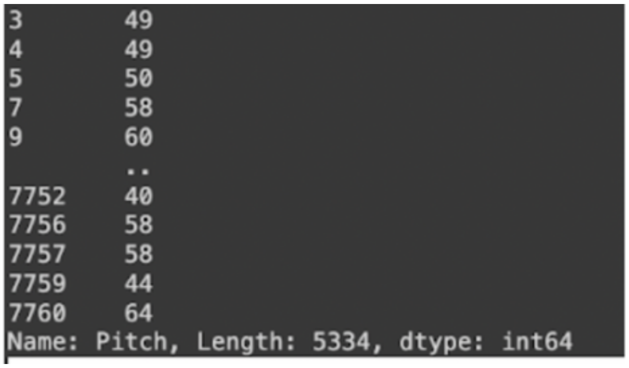
```
# Find the minimum and maximum values in the column
min_val = df['redshift'].min()
max_val = df['redshift'].max()

print(min_val)
print(max_val)

# Calculate the range of the input values
value_range = max_val - min_val

# Normalize each value to the 36 to 151 range
df['Pitch'] = round((df['redshift'] - min_val) / value_range * 69) + 36

# Set the maximum value, adjust manually to get a nice distribution from 36 to 103
max_value = 103

# Limit the values in the 'column_name' column to a maximum value
df['Pitch'] = df['Pitch'].apply(lambda x: np.random.randint(0, max_value) if x > max_value else x)

df['Pitch'].replace([np.nan, np.inf, -np.inf], 0, inplace=True)
df['Pitch']=(df['Pitch']+0).round().astype(int) #round to integer values


print(df['Pitch'])

# Create an interactive histogram
fig = px.histogram(df, x='Pitch', title='Pitch Histogram')

# Show the plot in the Jupyter Notebook
fig.show()
```

Figure 4.2 Code to normalize pitch values and create histogram



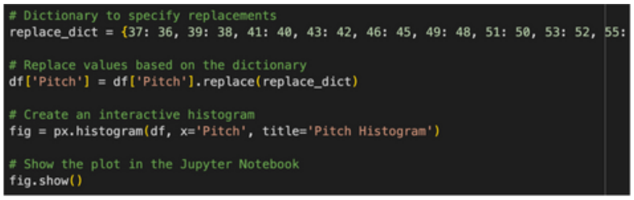
```
!pip install plotly
import plotly.express as px
import numpy as np
import pandas as pd

# Find the maximum value in the 'Values' column
max_value = df['redshift'].max()

print("Maximum value in the 'Values' column:", max_value)

# Value to be removed
value_to_remove = 0.3544

# Remove rows containing the specified value in the 'Values' column
df = df[df['redshift'] != value_to_remove]
```

Figure 4.3 Code to remove outlier from pitch data set



```
3       49
4       49
5       50
7       58
9       60
        ..
7752    40
7756    58
7757    58
7759    44
7760    64
Name: Pitch, Length: 5334, dtype: int64
```

Figure 4.4 Pitch data points



```
# Dictionary to specify replacements
replace_dict = {37: 36, 39: 38, 41: 40, 43: 42, 46: 45, 49: 48, 51: 50, 53: 52, 55:

# Replace values based on the dictionary
df['Pitch'] = df['Pitch'].replace(replace_dict)

# Create an interactive histogram
fig = px.histogram(df, x='Pitch', title='Pitch Histogram')

# Show the plot in the Jupyter Notebook
fig.show()
```

Figure 4.5 Code to adjust pitches to C +lyd mode

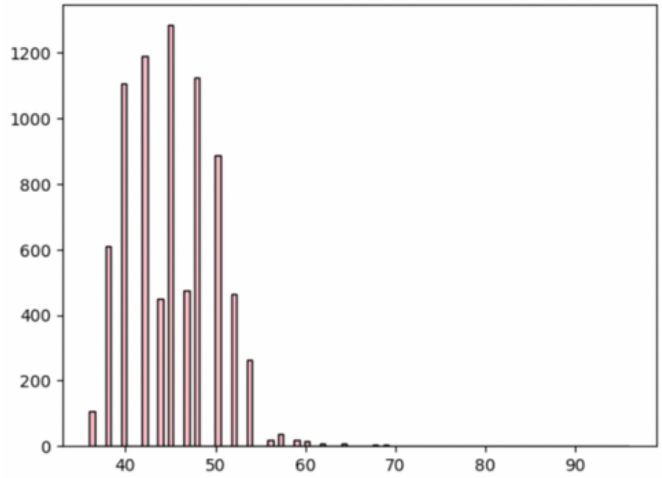


Figure 4.6 Histogram of pitches adjusted to C +lyd mode

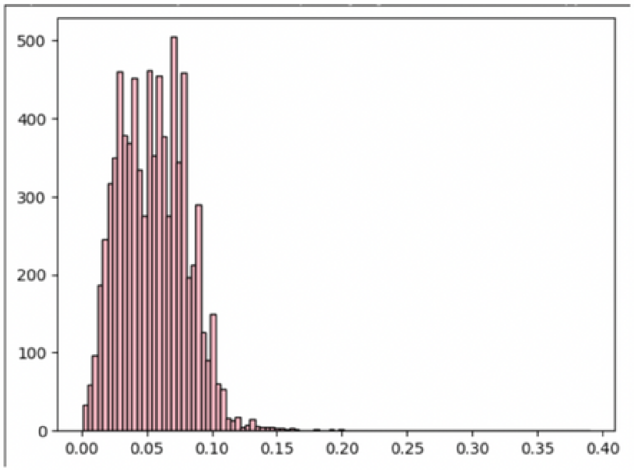
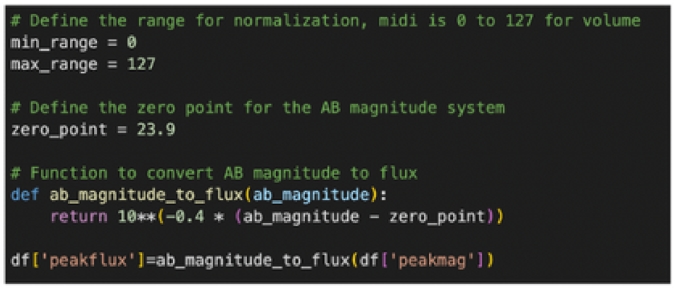


Figure 4.7 Redshift histogram, for comparison of proportionalities



```
# Define the range for normalization, midi is 0 to 127 for volume
min_range = 0
max_range = 127

# Define the zero point for the AB magnitude system
zero_point = 23.9

# Function to convert AB magnitude to flux
def ab_magnitude_to_flux(ab_magnitude):
    return 10**(-0.4 * (ab_magnitude - zero_point))

df['peakflux']=ab_magnitude_to_flux(df['peakmag'])
```

Figure 5.1 Code to transform peakmag into peakflux values

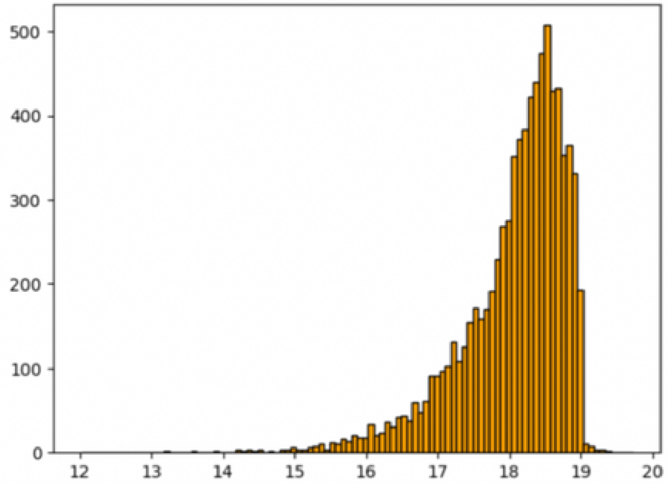
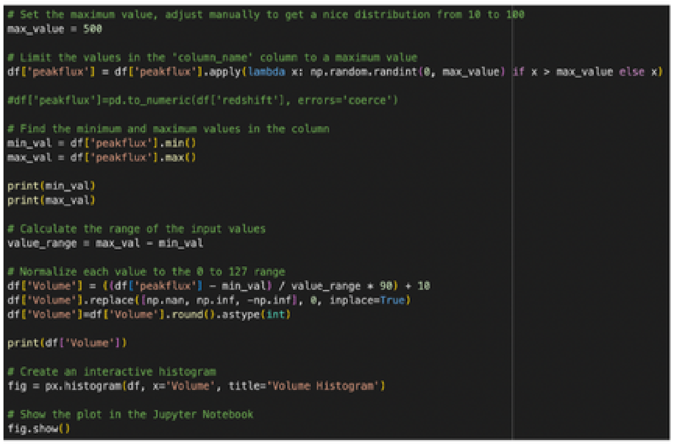
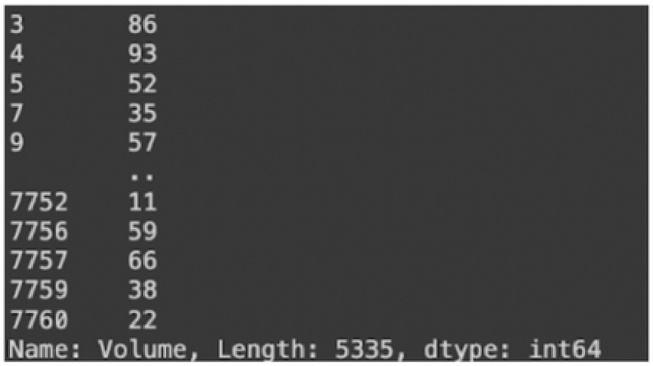


Figure 5.2 Peakmag values, for comparison of proportionalities



```
# Set the maximum value, adjust manually to get a nice distribution from 10 to 100
max_value = 500

# Limit the values in the 'column_name' column to a maximum value
df['peakflux'] = df['peakflux'].apply(lambda x: np.random.randint(0, max_value) if x > max_value else x)

#df['peakflux']=pd.to_numeric(df['redshift'], errors='coerce')

# Find the minimum and maximum values in the column
min_val = df['peakflux'].min()
max_val = df['peakflux'].max()

print(min_val)
print(max_val)

# Calculate the range of the input values
value_range = max_val - min_val

# Normalize each value to the 0 to 127 range
df['Volume'] = ((df['peakflux'] - min_val) / value_range * 90) + 10
df['Volume'].replace([np.nan, np.inf, -np.inf], 0, inplace=True)
df['Volume']=df['Volume'].round().astype(int)

print(df['Volume'])

# Create an interactive histogram
fig = px.histogram(df, x='Volume', title='Volume Histogram')

# Show the plot in the Jupyter Notebook
fig.show()
```

Figure 5.3 Code to normalize volume values and create histogram



```
3       86
4       93
5       52
7       35
9       57
        ..
7752    11
7756    59
7757    66
7759    38
7760    22
Name: Volume, Length: 5335, dtype: int64
```

Figure 5.4 Volume data points

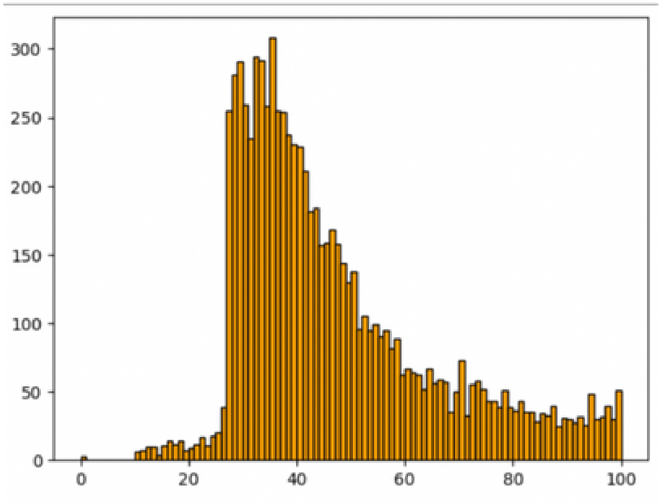
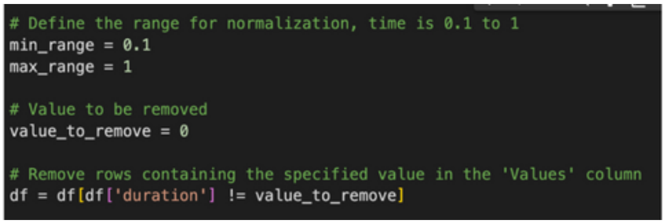


Figure 5.5 Volume histogram

```
df['duration'] = df['duration'].astype(str)
df['duration'] = df['duration'].str.replace('\W', '', regex=True)

# Convert 'duration' column to numeric
df['duration'] = pd.to_numeric(df['duration'], errors='coerce')
print(df['duration'])
```

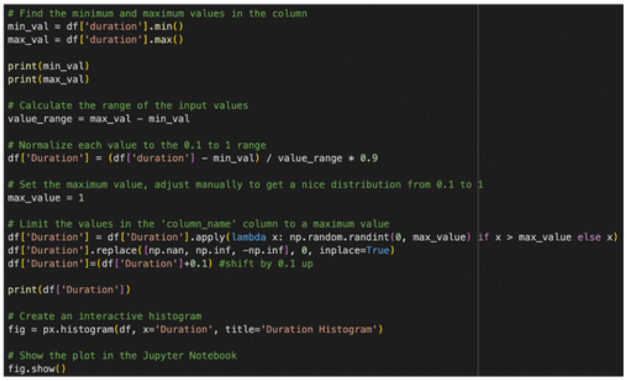
```
3       16607
4       10484
5        5269
7       22118
9       31031
        ...
7750     5717
7752      402
7756      202
7757     8394
7759    11908
Name: duration, Length: 5329, dtype: int64
```

Figure 6.1 Code to remove invalid character “>” from duration



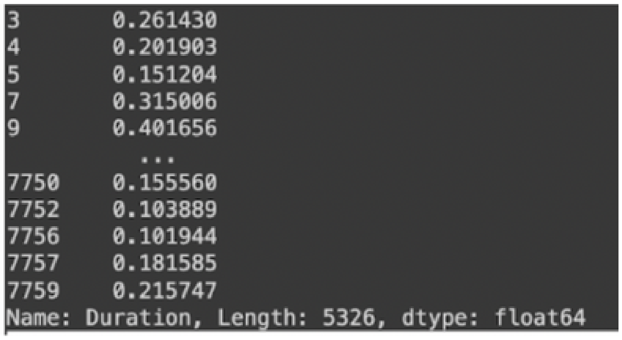
```
# Define the range for normalization, time is 0.1 to 1
min_range = 0.1
max_range = 1

# Value to be removed
value_to_remove = 0

# Remove rows containing the specified value in the 'Values' column
df = df[df['duration'] != value_to_remove]
```

Figure 6.2 Code to remove all values of 0 from duration



```
# Find the minimum and maximum values in the column
min_val = df['duration'].min()
max_val = df['duration'].max()

print(min_val)
print(max_val)

# Calculate the range of the input values
value_range = max_val - min_val

# Normalize each value to the 0.1 to 1 range
df['Duration'] = (df['duration'] - min_val) / value_range * 0.9

# Set the maximum value, adjust manually to get a nice distribution from 0.1 to 1
max_value = 1

# Limit the values in the 'column_name' column to a maximum value
df['Duration'] = df['Duration'].apply(lambda x: np.random.randint(0, max_value) if x > max_value else x)
df['Duration'].replace([np.nan, np.inf, -np.inf], 0, inplace=True)
df['Duration']=(df['Duration']+0.1) #shift by 0.1 up

print(df['Duration'])

# Create an interactive histogram
fig = px.histogram(df, x='Duration', title='Duration Histogram')

# Show the plot in the Jupyter Notebook
fig.show()
```

Figure 6.3 Code to duration volume values and create histogram



```
3       0.261430
4       0.201903
5       0.151204
7       0.315006
9       0.401656
          ...
7750    0.155560
7752    0.103889
7756    0.101944
7757    0.181585
7759    0.215747
Name: Duration, Length: 5326, dtype: float64
```

Figure 6.4 Duration data points

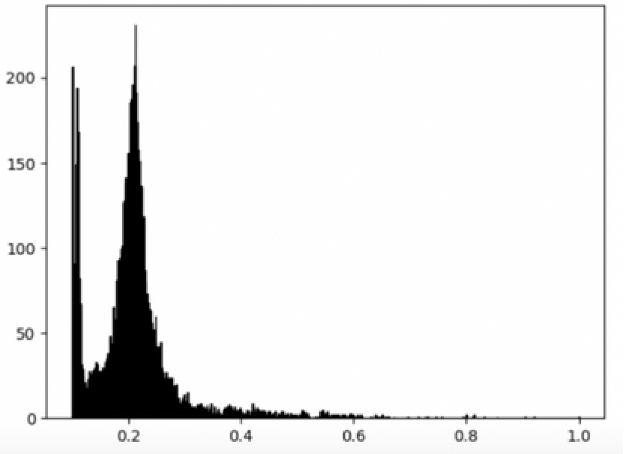
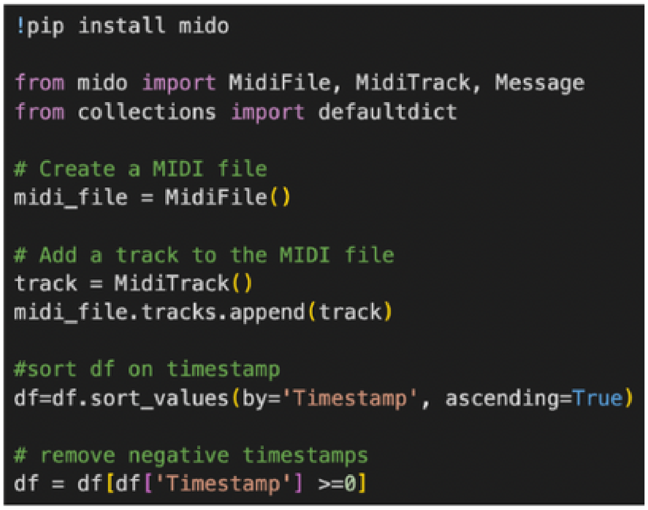


Figure 6.5 Duration histogram



```
!pip install mido

from mido import MidiFile, MidiTrack, Message
from collections import defaultdict

# Create a MIDI file
midi_file = MidiFile()

# Add a track to the MIDI file
track = MidiTrack()
midi_file.tracks.append(track)

#sort df on timestamp
df=df.sort_values(by='Timestamp', ascending=True)

# remove negative timestamps
df = df[df['Timestamp'] >=0]
```

Figure 7.1 Code to create MIDI file and sort table by Timestamp

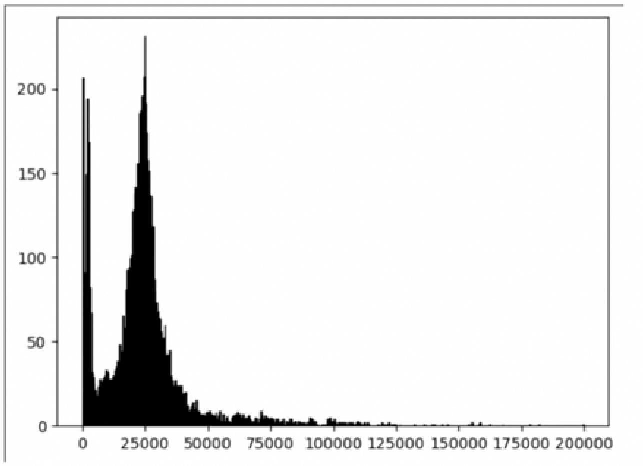


Figure 4.7 Unmodified duration histogram, for comparison of proportionalities

```
# Add notes to the MIDI track with overlapping timestamps
prev_time = 0
for index, row in df.iterrows():
   duration = int(float(row['Duration']))
   pitch = int(row['Pitch'])
   volume = int(row['Volume'])
   time_delta = int((float(row['Timestamp']) - prev_time) *1000) # Convert to milliseconds

   track.append(Message('note_on', note=pitch, velocity=volume, time=time_delta))
   track.append(Message('note_off', note=pitch, velocity=0, time=int(time_delta+duration*1000))) # Convert duration to milliseconds
   prev_time = float(row['Timestamp'])


# Specify the output file path and name
output_file = "ZTF.mid"

# Save the MIDI file
midi_file.save(output_file)
```

Figure 7.2 Code to define MIDI values and export MIDI file

Figure 8.1 Raw MIDI file from Colab uploaded directly into Garageband

Figure 9.1 Garageband MIDI of Rhythm Modifications

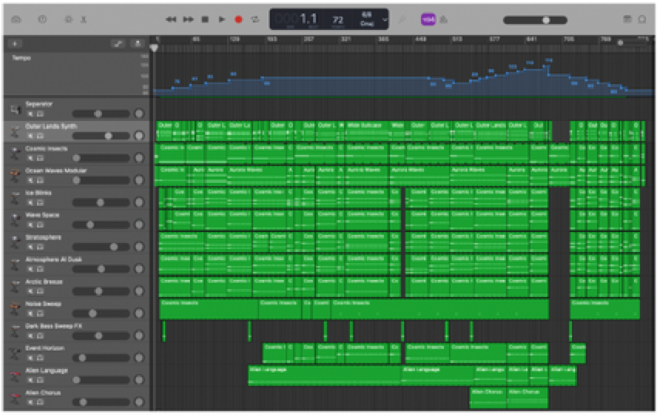


Figure 10.3 Garageband MIDI of Soundscape Modifications

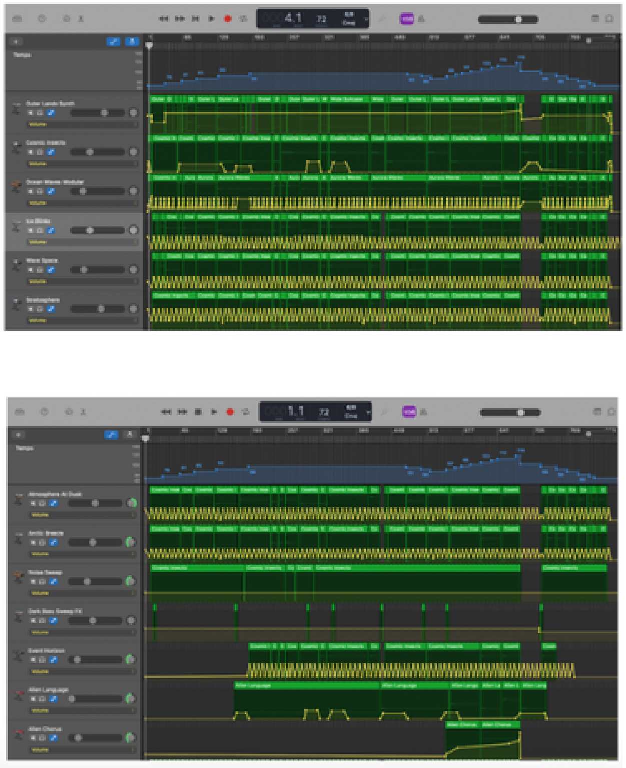


Figure 10.1, Figure 10.2 Garageband Soundscape Automation

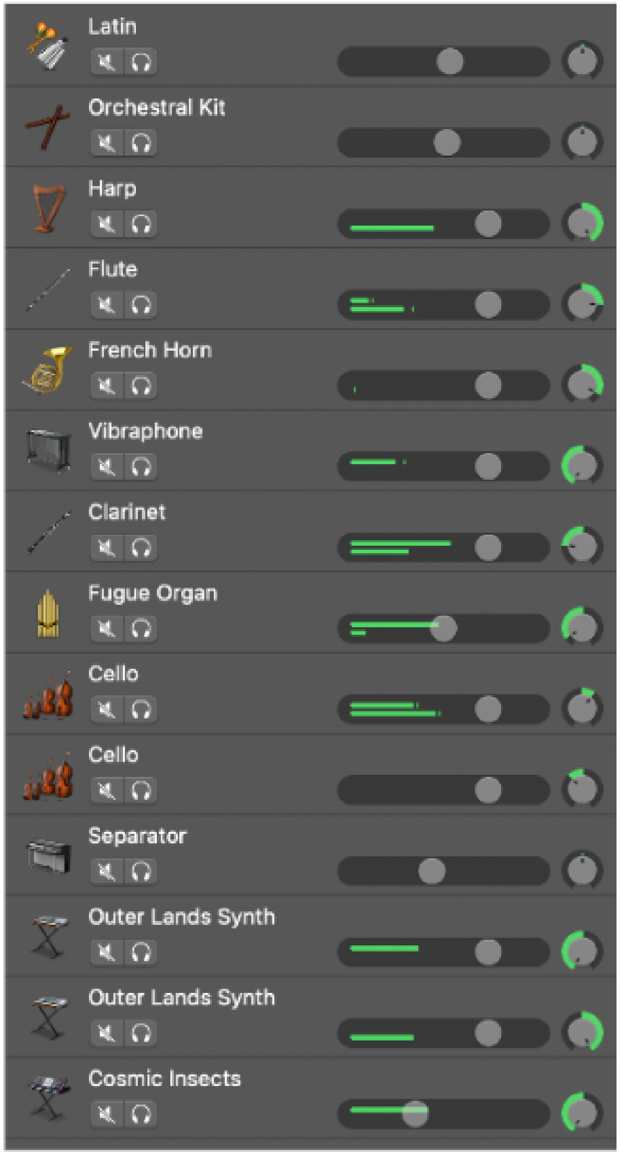
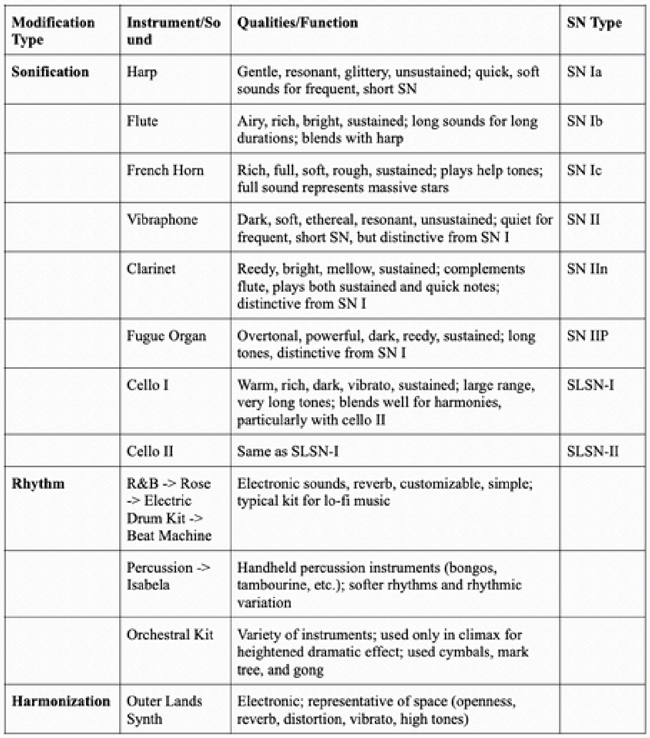


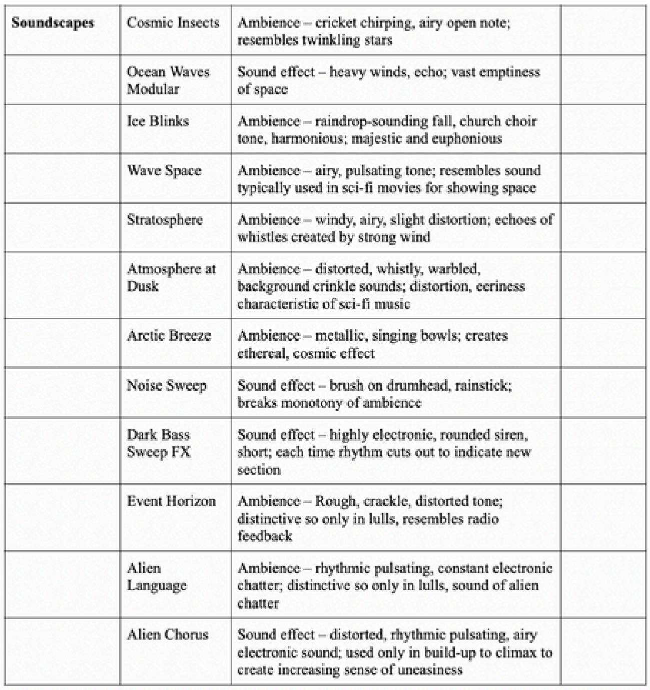
Figure 11.1 Garageband Panning Sample

Appendix D (Instrumentation Chart)

All instrumentation used, type of modification, qualitative characteristics, function in the piece, and SN Type if applicable



Modification Type	**Instrument/Sound**	**Qualities/Function**	**SN Type**
Sonification	Harp	Gentle, resonant, glittery, unsustained; quick, soft sounds for frequent, short SN	SN Ia
	Flute	Airy, rich, bright, sustained; long sounds for long durations; blends with harp	SN Ib
	French Horn	Rich, full, soft, rough, sustained; plays help tones; full sound represents massive stars	SN Ic
	Vibraphone	Dark, soft, ethereal, resonant, unsustained; quiet for frequent, short SN, but distinctive from SN I	SN II
	Clarinet	Reedy, bright, mellow, sustained; complements flute, plays both sustained and quick notes; distinctive from SN I	SN IIn
	Fugue Organ	Overtonal, powerful, dark, reedy, sustained; long tones, distinctive from SN I	SN IIP
	Cello I	Warm, rich, dark, vibrato, sustained; large range, very long tones; blends well for harmonies, particularly with cello II	SLSN-I
	Cello II	Same as SLSN-I	SLSN-II
Rhythm	R&B -> Rose -> Electric Drum Kit -> Beat Machine	Electronic sounds, reverb, customizable, simple; typical kit for lo-fi music	
	Percussion -> Isabela	Handheld percussion instruments (bongos, tambourine, etc.); softer rhythms and rhythmic variation	
	Orchestral Kit	Variety of instruments; used only in climax for heightened dramatic effect; used cymbals, mark tree, and gong	
Harmonization	Outer Lands Synth	Electronic; representative of space (openness, reverb, distortion, vibrato, high tones)	



Soundscapes	Cosmic Insects	Ambience – cricket chirping, airy open note; resembles twinkling stars	
	Ocean Waves Modular	Sound effect – heavy winds, echo; vast emptiness of space	
	Ice Blinks	Ambience – raindrop-sounding fall, church choir tone, harmonious; majestic and euphonious	
	Wave Space	Ambience – airy, pulsating tone; resembles sound typically used in sci-fi movies for showing space	
	Stratosphere	Ambience – windy, airy, slight distortion; echoes of whistles created by strong wind	
	Atmosphere at Dusk	Ambience – distorted, whistly, warbled, background crinkle sounds; distortion, eeriness characteristic of sci-fi music	
	Arctic Breeze	Ambience – metallic, singing bowls; creates ethereal, cosmic effect	
	Noise Sweep	Sound effect – brush on drumhead, rainstick; breaks monotony of ambience	
	Dark Bass Sweep FX	Sound effect – highly electronic, rounded siren, short; each time rhythm cuts out to indicate new section	
	Event Horizon	Ambience – Rough, crackle, distorted tone; distinctive so only in lulls, resembles radio feedback	
	Alien Language	Ambience – rhythmic pulsating, constant electronic chatter; distinctive so only in lulls, sound of alien chatter	
	Alien Chorus	Sound effect – distorted, rhythmic pulsating, airy electronic sound; used only in build-up to climax to create increasing sense of uneasiness	

REFERENCES

- Alexander, R. L., O'Modhrain, S., Roberts, D. A., Gilbert, J. A., & Zurbuchen, T. H. (2014). The bird's ear view of space physics: Audification as a tool for the spectral analysis of time series data. Journal of Geophysical Research: Space Physics, 119(7), 5259–5271. https://doi.org/10.1002/2014ja020025
- Bartkowiak, M. J. (2010). *Sounds of the future: Essays on music in science fiction film*. McFarland & Co. https://books.google.com/books?hl=en&lr=&id=i5mKZT5D3TMC&oi=fnd&pg=PA22&dq=lisa+schmidt+sci-fi+music&ots=pHUple_2MI&sig=uEzNkoCxO7_w3bgGdY_t73uOW-c#v=onepage&q=lisa%20schmidt%20sci-fi%20music&f=false
- Crockett, C. (2019, August 8). *A musician composes a solar soundtrack*. Science News. https://www.sciencenews.org/article/musician-composes-solar-soundtrack
- Eriksson, T. (n.d.). *Chord player*. OneMotion.com. https://www.onemotion.com/chord-player/
- Hevner, K. (1936). Experimental studies of the elements of expression in music. *The American Journal of Psychology*, *48*(2), 246. https://doi.org/10.2307/1415746
- Inspire Acoustics. (n.d.). *MIDI note numbers and center frequencies*. Inspire Acoustics. https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies
- Kramer, G. (1995). Auditory display: Sonification, Audification, and auditory interfaces. *Computer Music Journal*, *19*(2), 110. https://doi.org/10.2307/3680606
- Parker, A. (2011). *Alex Harrison Parker*. Alex Harrison Parker | Astronomy | Supernova Sonata. https://www.astro.uvic.ca/~alexhp/new/supernova_sonata.html
- Samuel, D. (n.d.). Sensory studies. The Music of the Spheres. https://www.sensorystudies.org/picture-gallery/spheres_image/

- Schiller, J. (2012, October 5). Astronomer creates Hubble image mashup of Starry night and composes cosmic conciertos. Wired. https://www.wired.com/2012/10/hubble-image-mashup-of-starry-night/
- Stefani, W. H. (1981). The psycho-physiological effects of volume, pitch, harmony and rhythm ... The Psycho-physiological Effects of Volume, Pitch, Harmony and Rhythm in the Development of Western Art Music Implications for a Philosophy of Music History. https://www.semanticscholar.org/paper/The-Psycho-physiological-Effects-of-Volume%2C-Pitch%2C-Stefani/88f9cc53ce5880c373b679d8faf3d53e2ab57401
- Temming, M. (2020, October 16). Turning space images into music makes astronomy more accessible. Science News. https://www.sciencenews.org/article/space-images-music-astronomy-accessible-sonification
- The Chromatics. (2023). Astrocappella. AstroCappella. https://www.astrocappella.com/
- The Schools' Observatory. (n.d.). The Schools' Observatory. Types of Supernovae | The Schools' Observatory. https://www.schoolsobservatory.org/discover/projects/supernovae/typeI
- Valle, A., & Korol, V. (2022, February 9). For Lisa. A piano-based sonification project of Gravitational Waves. arXiv. https://arxiv.org/abs/2202.04621
- Zwicky Transient Facility. (2023, March 10). Classifying supernovae with the Zwicky Transient Facility. YouTube. https://www.youtube.com/watch?v=D56q-PnHheo