Blue CoLab

Sonification Workshop

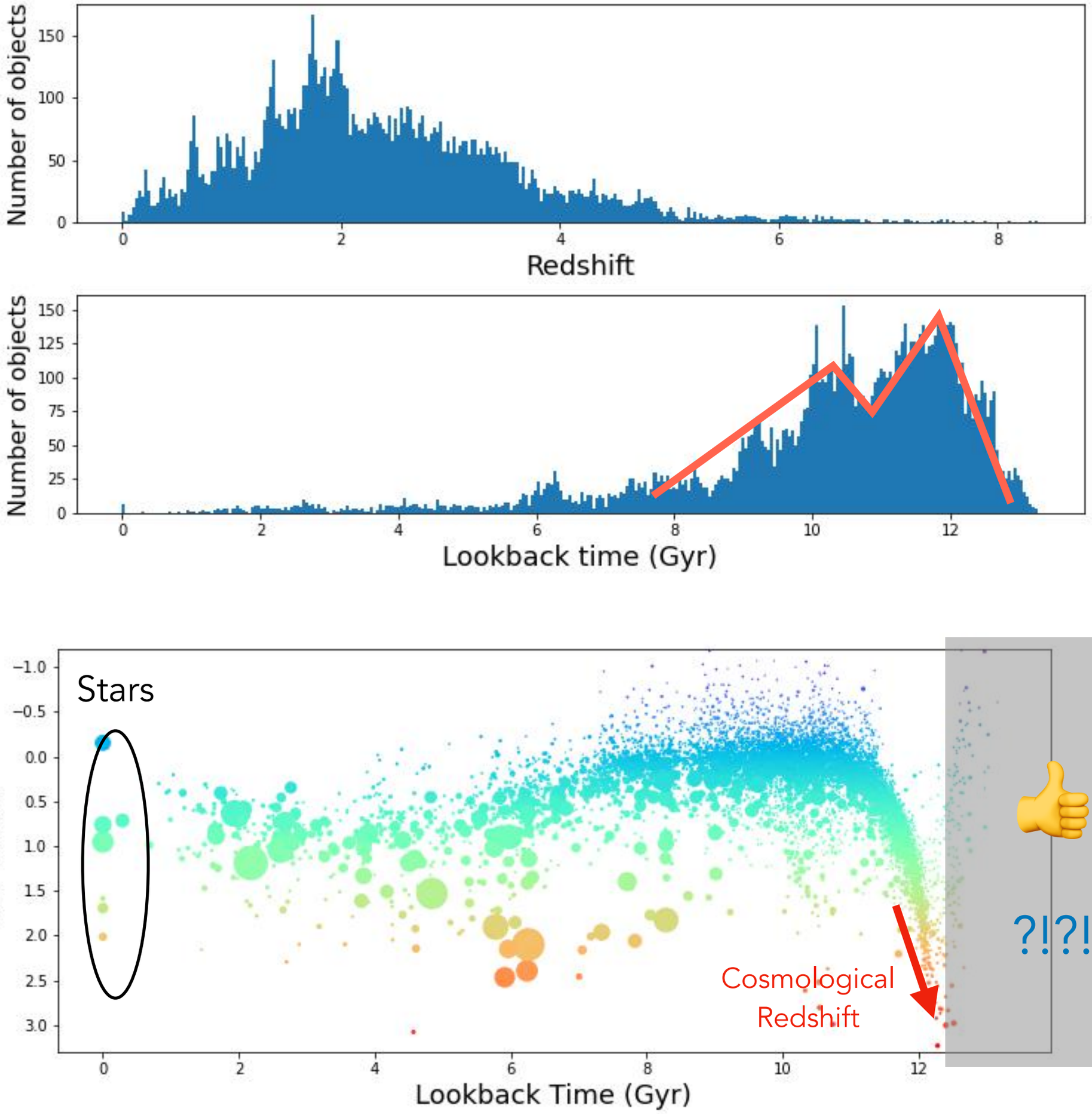
Dr. Matt Russo

Feb. 25. 2023

Looking Back 13.35 Billion Years

Table 5
Column Description of UDF Catalog

Column No.	Column Title	Description
1	ID	Object identification number
2	COE_ID	Object identification number from Coe et al. (2006)[a] (-99 if no match)
3	RA	Right ascension (J2000 in units of decimal degrees)
4	DEC	Declination (J2000 in units of decimal degrees)
5	X	Image X pixel coordinate in the UDF mosaic
6	Y	Image Y pixel coordinate in the UDF mosaic
7-17	MAG_*	Total AB magnitude of each filter[b]
18-28	MAGERR_*	Total AB magnitude uncertainty of each filter[c]
29-39	FLUX_*	Total flux of each filter in units of μJy[d]
40-50	FLUXERR_*	Total flux uncertainty of each filter in units of μJy[d]
51-61	FLUX_ISO_*	`SExtractor` isophotal Flux of each filter in units of e^-/s[d]
62-72	FLUXERR_ISO_*	`SExtractor` isophotal Flux uncertainty of each filter in units of e^-/s[d]
73	APCOR	Aperture correction to convert from isophotal magnitude to total magnitude[e]
74	FWHM	`SExtractor` F775W FWHM in units of pixel (1 pixel = 0.03″)
75	AREAF	`SExtractor` F775W isophotal area (filtered) above detection threshold in units of pixel^2
76	STELLARITY	`SExtractor` F775W stellarity
77	ELLIPTICITY	`SExtractor` F775W ellipticity
78	THETA	`SExtractor` position angle in units of degrees
79	UVUDF_COVERAGE	1 = covered by NUV data, 0 = not covered by NUV data
80	UVUDF_EDGEFLG	1 = close to NUV edge or chip gap, 0 = not close to edge or chip gap
81	Z_BPZ	Bayesian photometric redshift (BPZ)
82	ZMIN_BPZ	BPZ 95% lower limit to redshift
83	ZMAX_BPZ	BPZ 95% upper limit to redshift
84	ODDS_BPZ	BPZ integrated $P(z)$ contained within $2 * 0.03 * (1 + z)$
85	CHISQ2_BPZ	BPZ modified reduced chi square ($\chi^2_{\rm mod}$)
86	TEMPLATE_BPZ	BPZ template number[f]
87	Z_EAZY	EAZY photometric redshift
88	ZMIN_EAZY	EAZY 95% lower limit to redshift
89	ZMAX_EAZY	EAZY 95% upper limit to redshift
90	ODDS_EAZY	EAZY integrated $P(z)$ contained within $0.2(1 + z)$
91	CHISQ_EAZY	EAZY reduced chi square (χ^2_ν)
92	NFOBS	Number of filters available for photometric redshift
93	NF5SIG	Number of filters with signal to noise above 5
94	SPECZ	Spectroscopic redshift (-99 if no value)
95	SPECZ_REF	Reference for spectroscopic redshift[g]
96	GRISMZ	Grism + EAZY redshift from 3D-HST team (Brammer et al. 2012) (-99 if no value)
97	STAR	Stars identified by GRAPES program (Pirzkal et al. 2005) (1=star)


Number of objects
Redshift
Number of objects
Lookback time (Gyr)
Stars
B-V Colour
Lookback Time (Gyr)
Cosmological
Redshift
?!?!

Data

What does your data ‘look like’?
(parameters, ranges, trends…,background research)

What is the central idea you’d like to communicate?
What story do you want to tell?

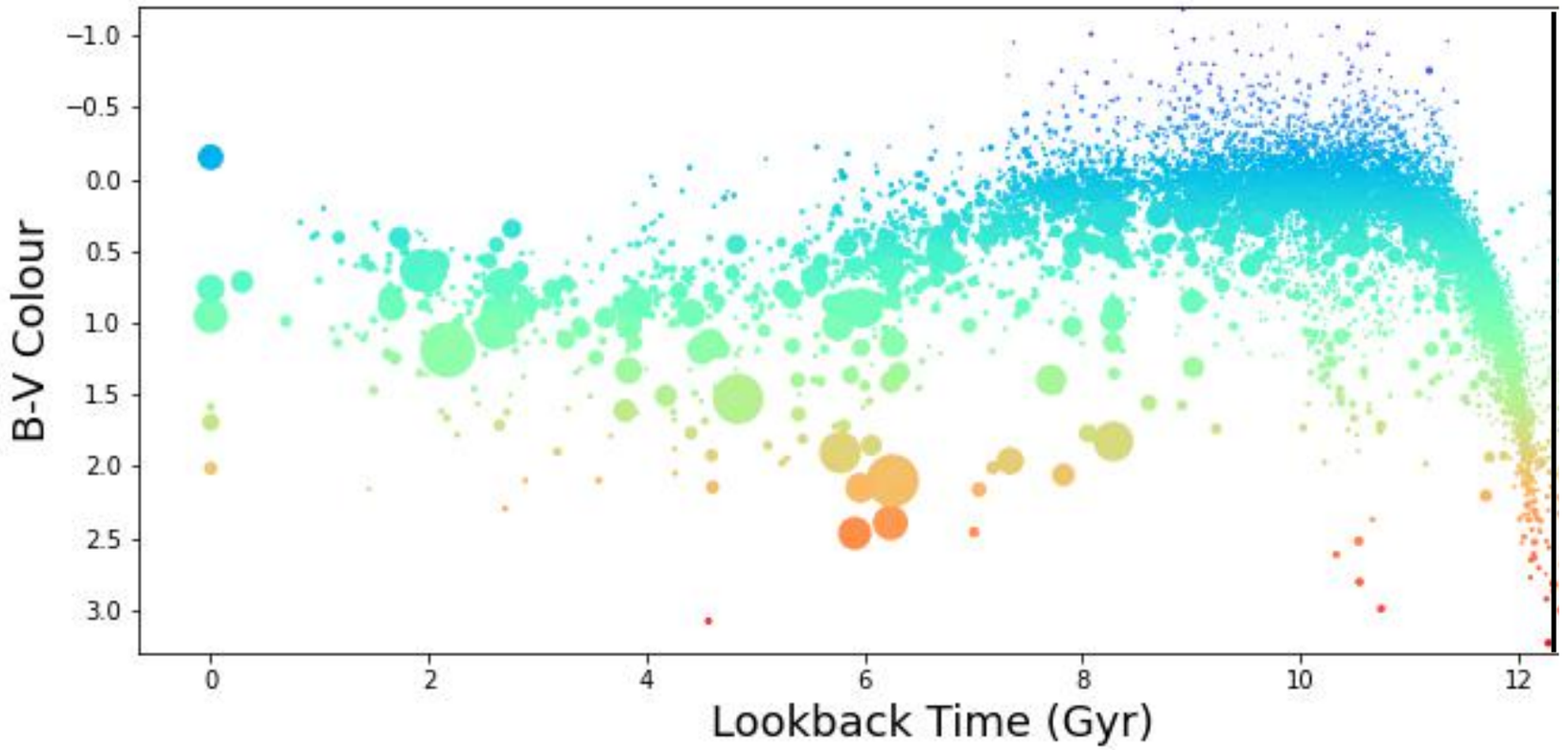
Your Turn (10min)

Add answers to the following questions on the shared Google Doc:

1. What data will you be sonifying?
2. What is the central idea you’d like to communicate through sound?

High Notes

Low Notes



Lookback time → **Time** (greater lb time is later)

Color → **Musical Pitch** (bluer is higher)

Apparent Size → **MIDI Velocity** (bigger is higher)

X Coordinate → **Stereo Pan** (right is right, left is left)

Mapping

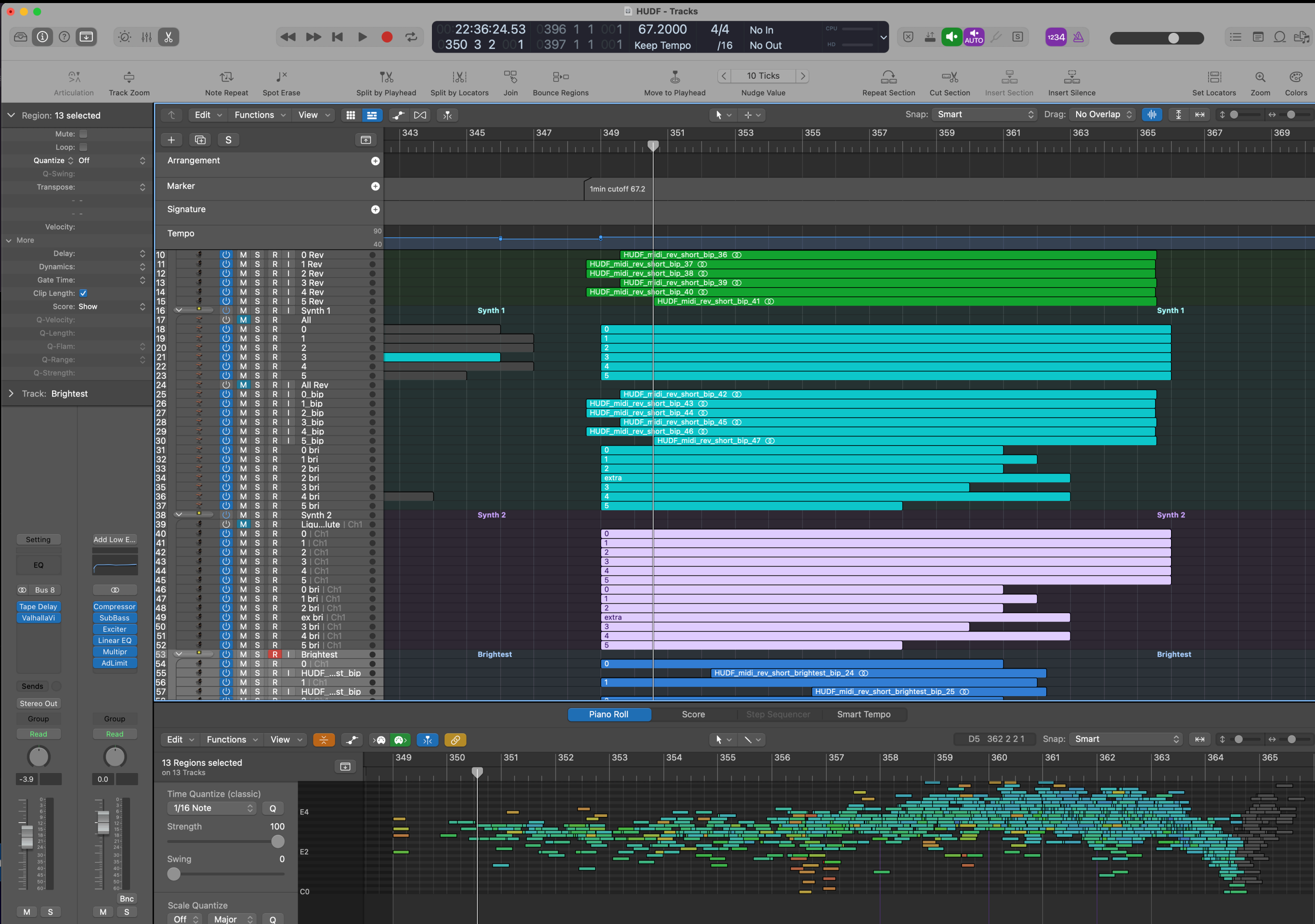
Which data parameters will control which sonic parameters?
Simple/intuitive mappings or complex/opaque?

Discrete or continuous?
Range of each parameter?
Polarity of each parameter?
Musical considerations (instrument ranges, timbres, etc)

Your Turn (10min)

Describe your **sonification mapping** choices on the shared Google Doc

Map data to MIDI with python, import into Logic…



Technique

MIDI? Digital signal processing (synthesis)? Performance? Interactive? ….

Software:

Python (or any other general programming language)
Supercollider (very powerful synthesis)
WebAudio (html/css/javascript)
Pre-existing sonification tool (ex. TwoTone)
DAW (Logic, Ableton, etc)

Visualization

Direct or abstract connection between data and visuals?
Static, animated, interactive?
Will visuals help explain the sonification mapping?

Software:

Python, Processing, P5 (or any other general programming language)
Animation software: After Effects, Blender, etc
Video Editing: Final Cut Pro, Premiere, etc
Web (html/css/javascript, three.js, d3)

Prototyping

data2midi-BC.ipynb

- maps discrete data to MIDI pitch and velocity
- maps continuous data to MIDI control changes

1. Use the notebook to load and map your data to MIDI
2. Import the MIDI file(s) into Reaper (or Logic)
3. We'll regroup at the end to share your progress

Assigning CC to a parameter in Reaper

1. save CC Data in MIDI file in python (remember which CC number you use)
2. Drag and drop the MIDI file onto a new track in Reaper
3. To choose an instrument sound: press FX on the track, then ADD (Reaper's default synth is ReaSynth)
4. Press TRIM on the track
5. Select a parameter (ex Volume)
6. Press MOD beside that parameter
7. Select LINK FROM MIDI OR FX PARAMETER
8. From the dropdown, choose MIDI->CC-> CC number

Assigning CC to a parameter in Logic

1. save CC Data in MIDI file in python
2. Drag and drop the MIDI file onto a new track in Logic
3. Choose an instrument from the Library, add effects to the track
4. Press A to see automation, press ‘Track’ on the track so that it switches to Region
5. Select the current automation channel from the dropdown
6. Right click on the track, choose automation->Move region automation to track
7. Enter the automation channel drop down and press Option while choosing the parameter you’d like to assign the CC data to (chose convert to move, copy and convert to copy and move)