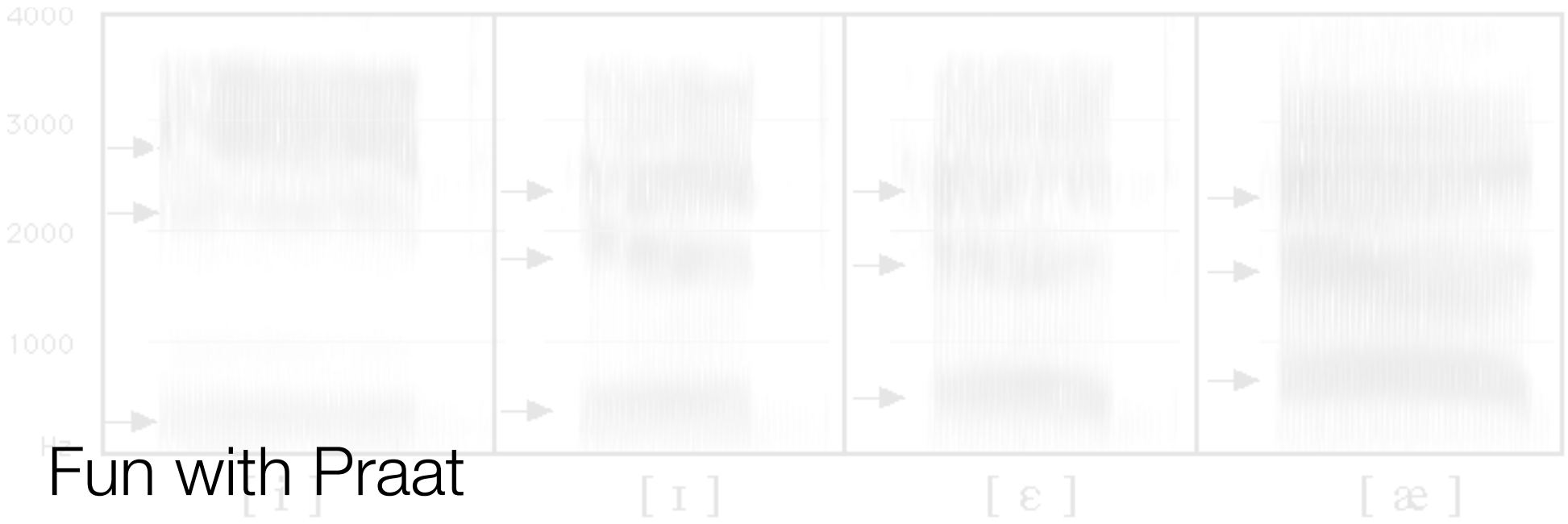
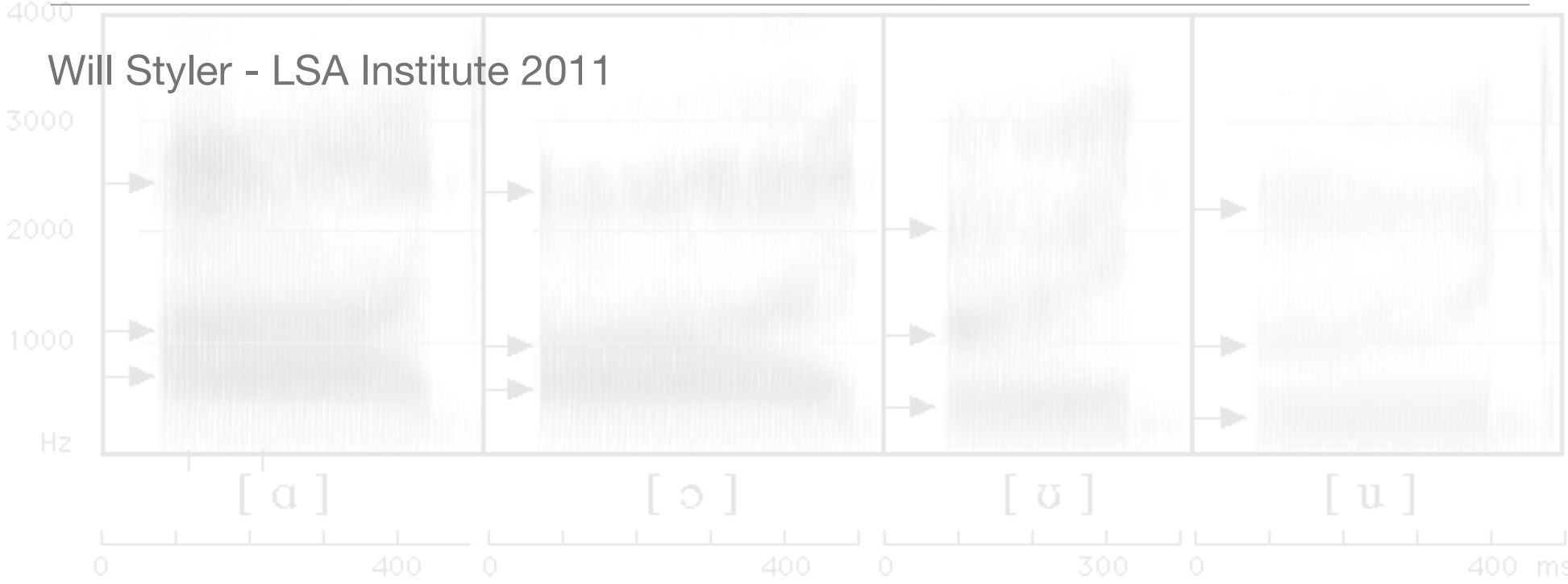


Fun with Praat

Will Styler - LSA Institute 2011



Some introductory acoustics and digital audio information

What is sound?

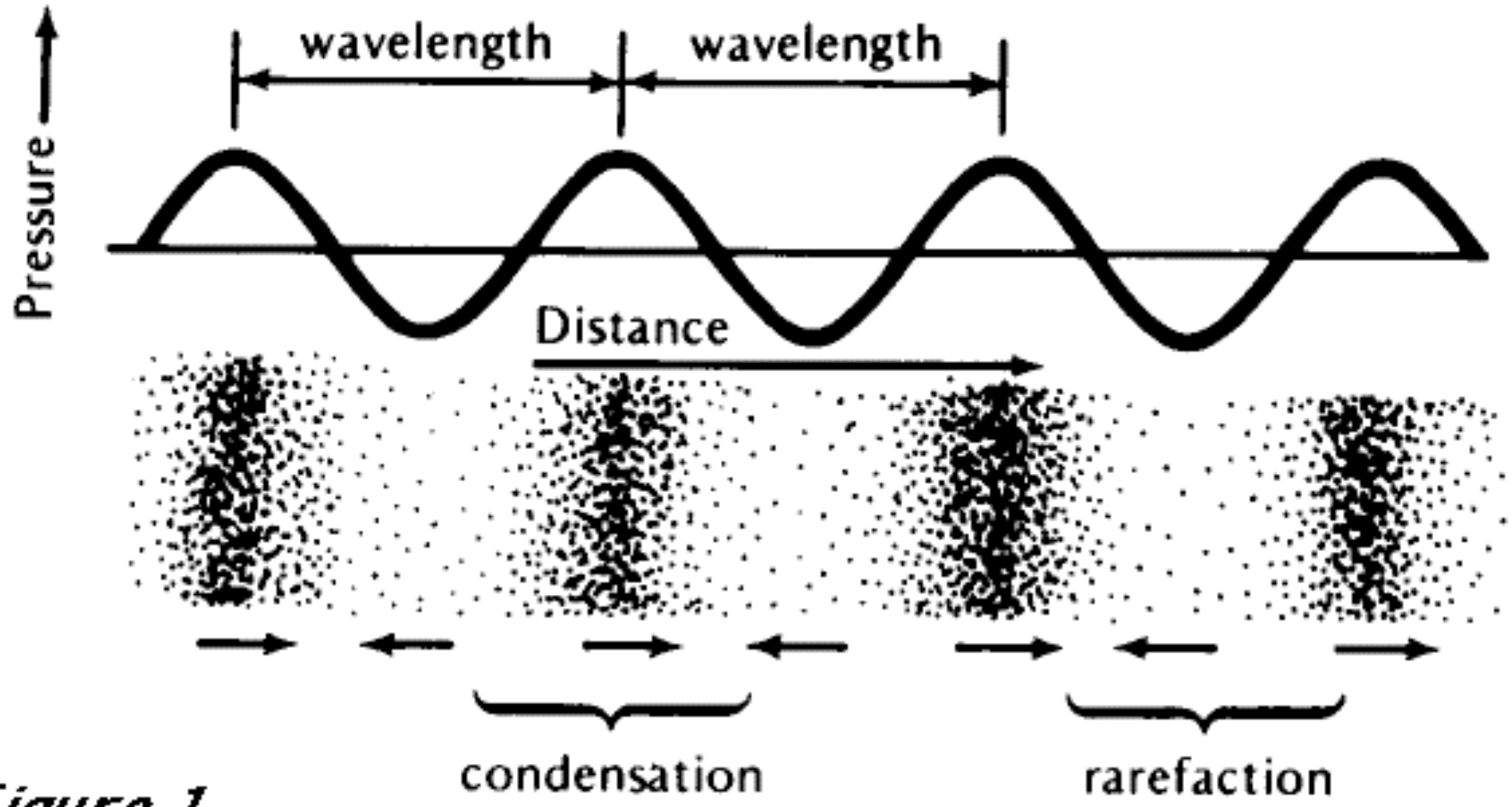
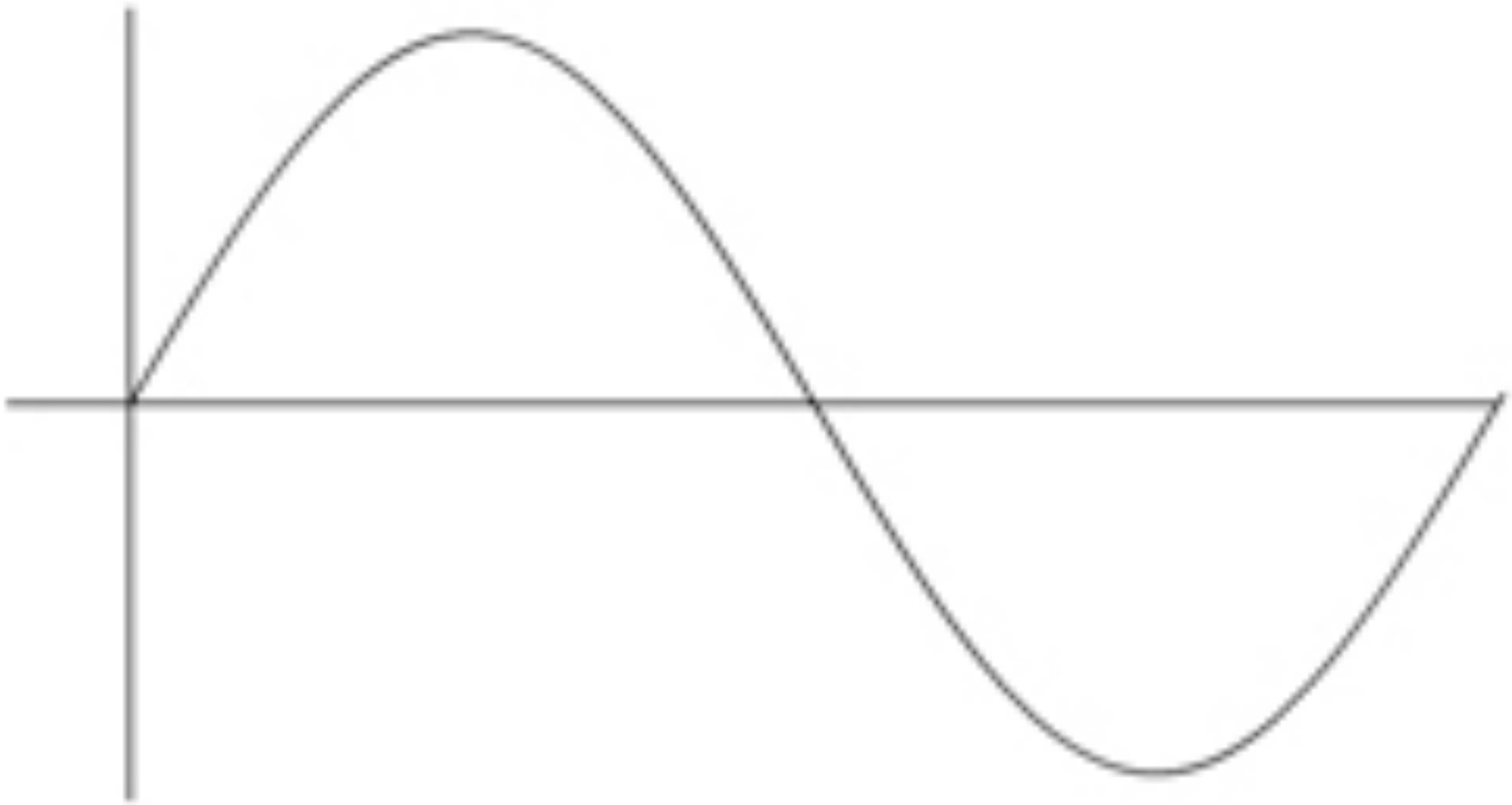

wavelength
wavelength
Pressure
Distance
condensation
rarefaction

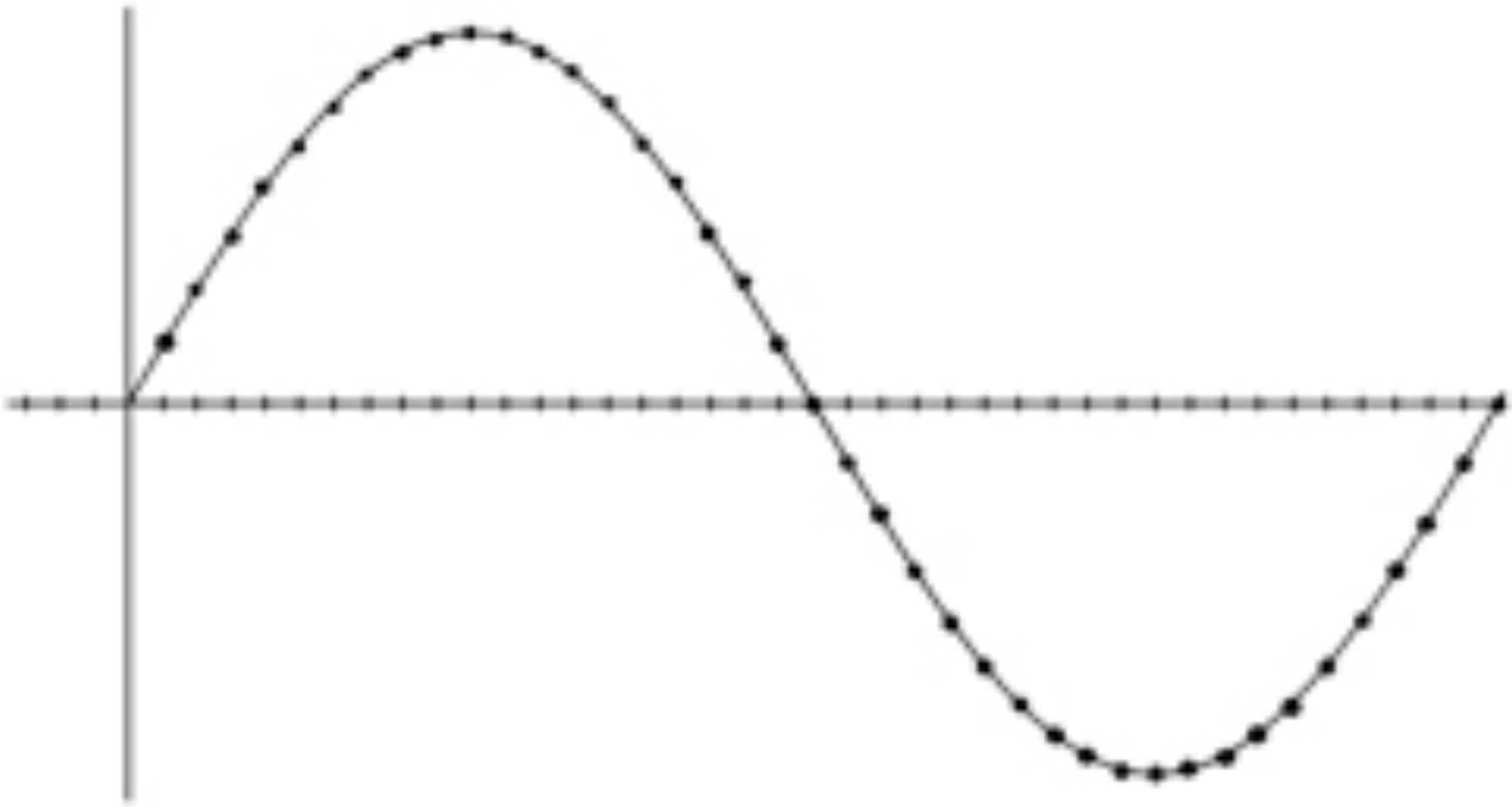
Figure 1

Digitization of Sound

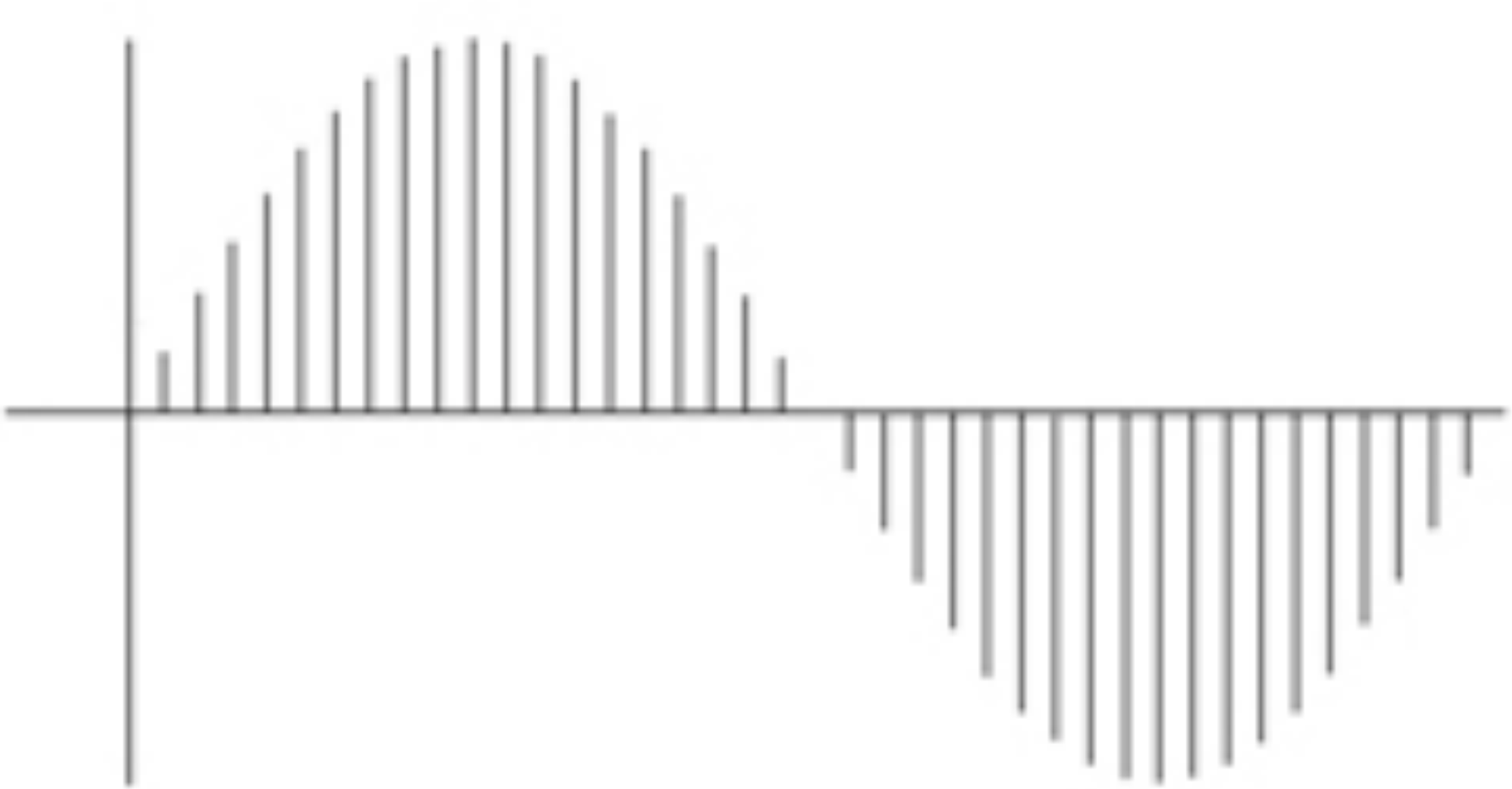
Computers don’t understand waves, just 0 and 1



d)

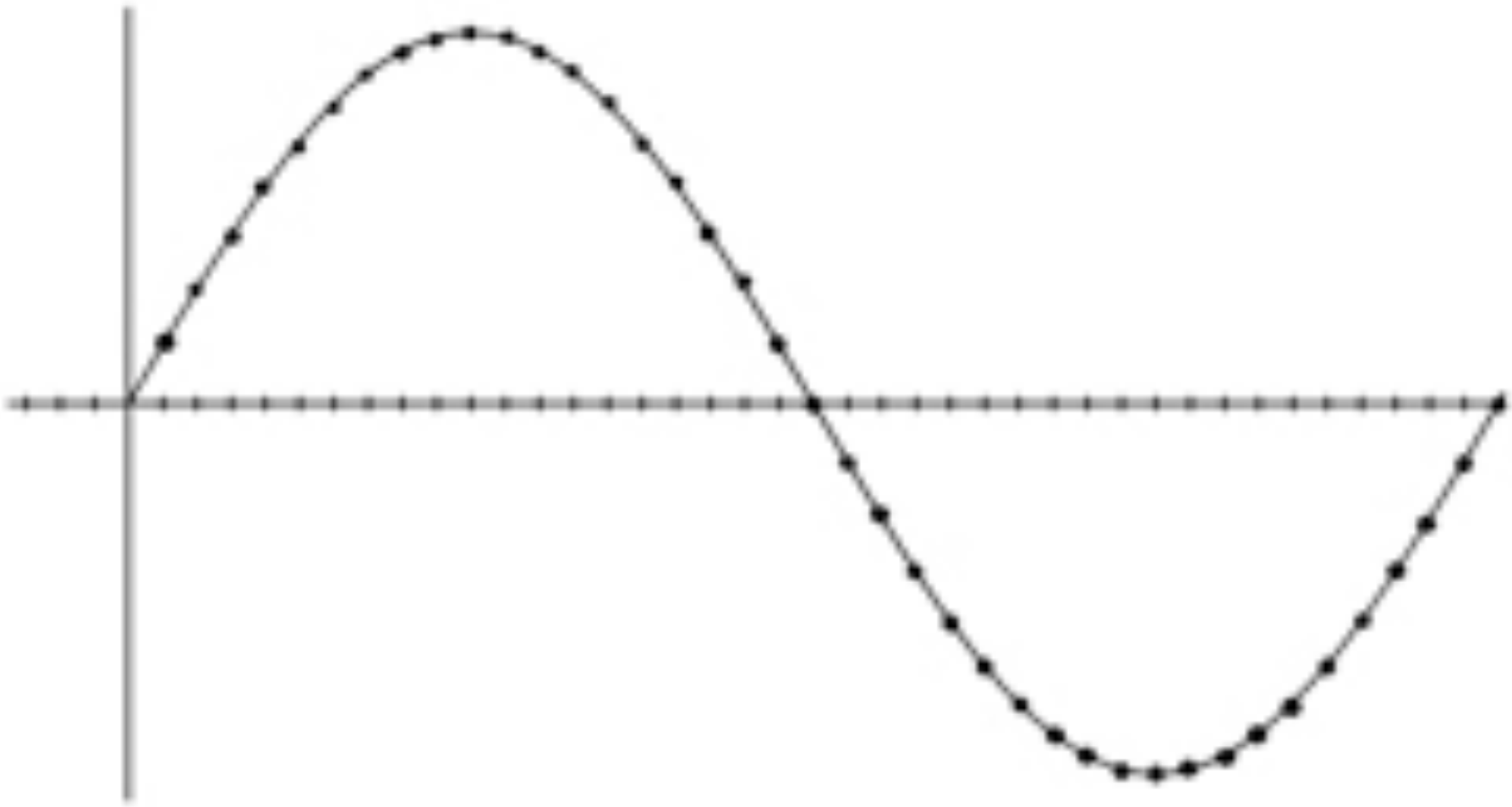


a)

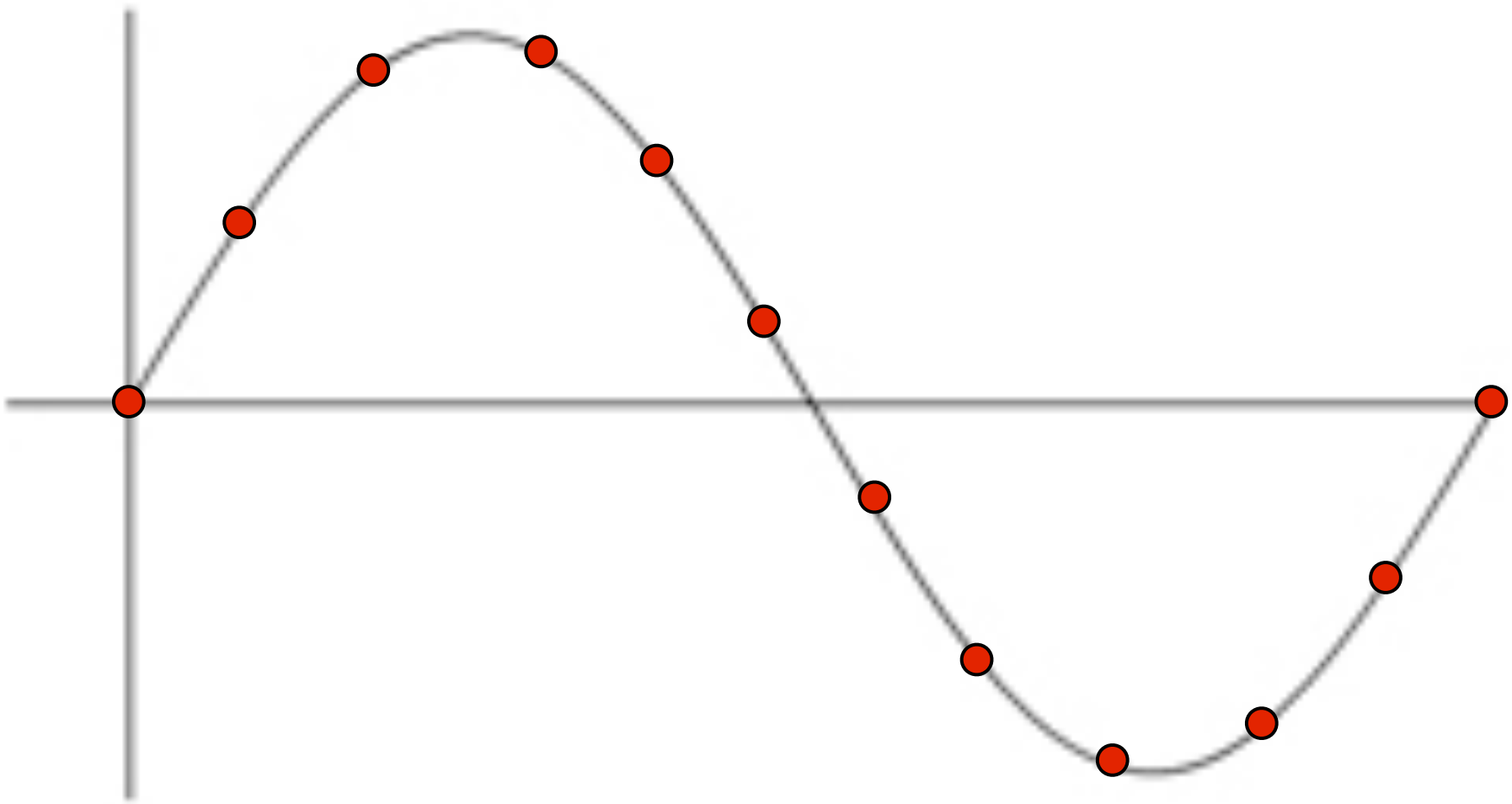


b)

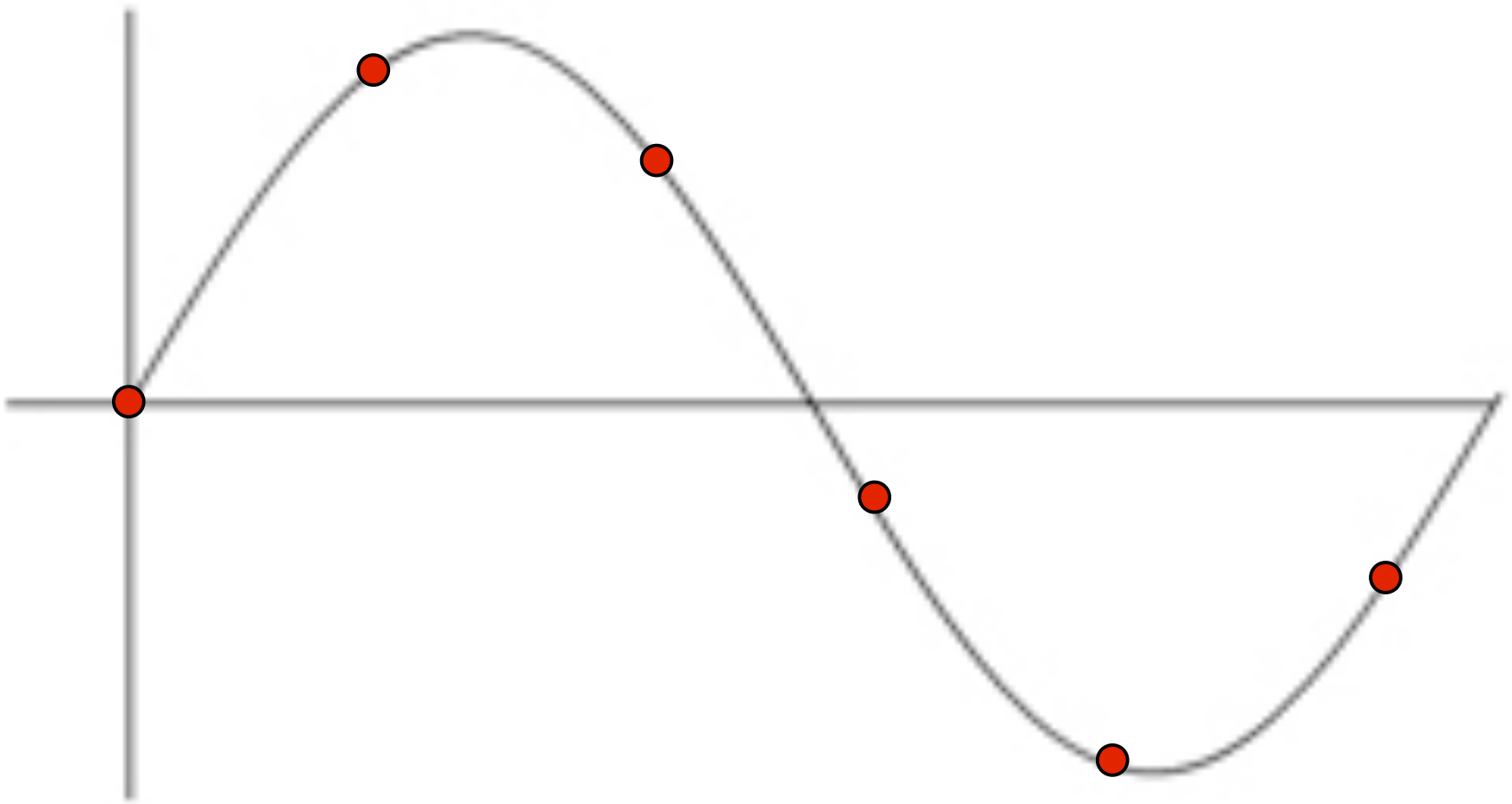
Sampling rate



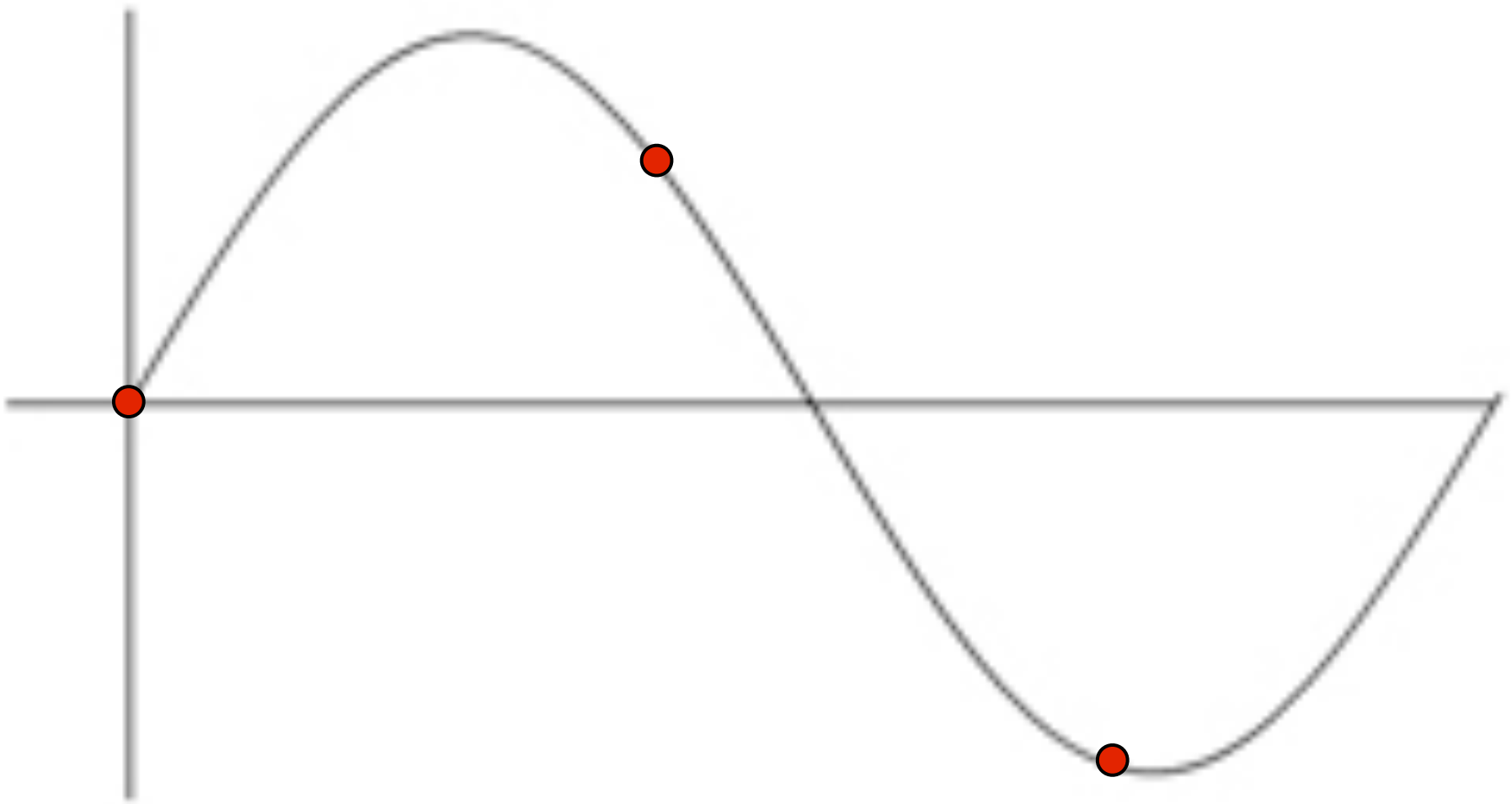
a)



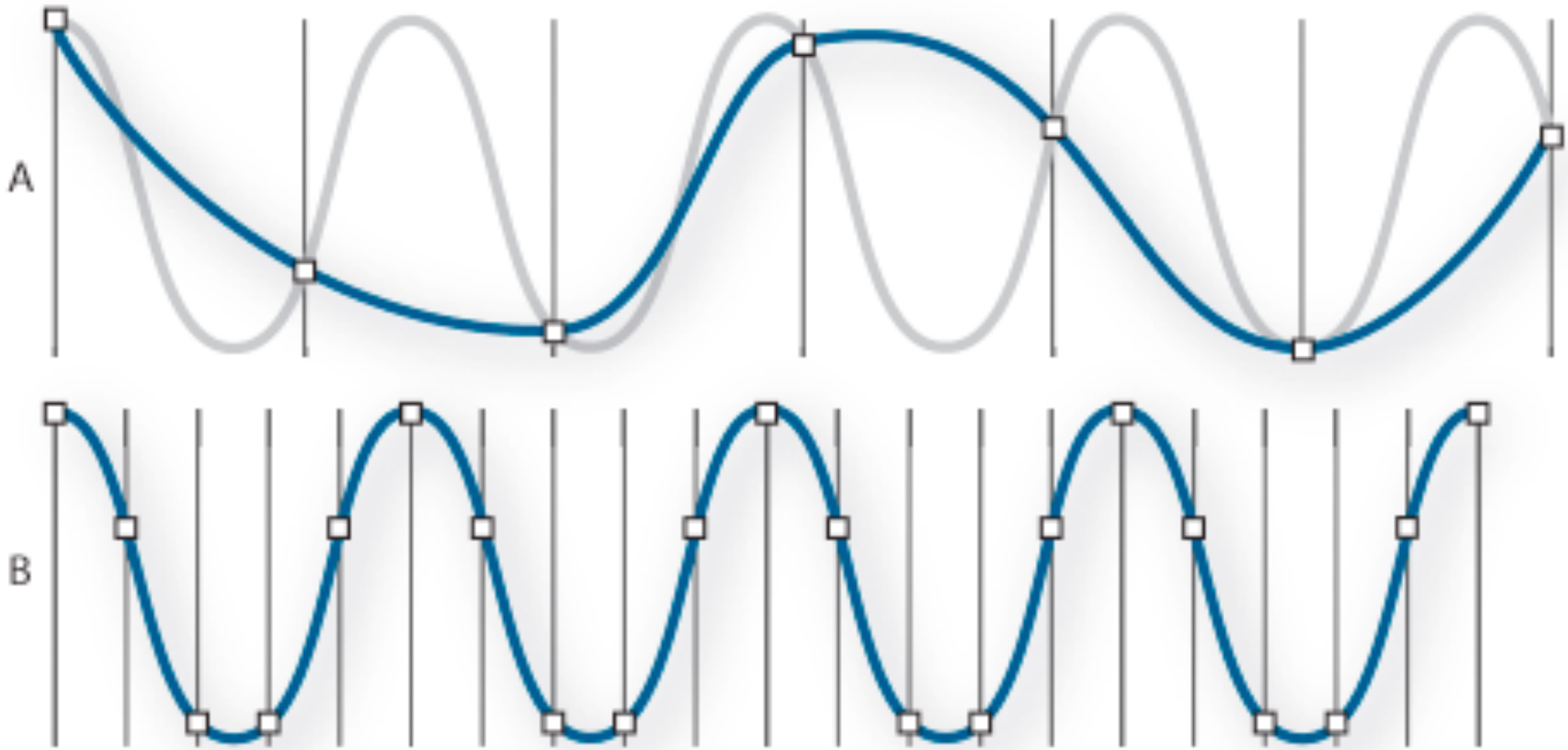
d)



d)



d)


A
B

The Nyquist Frequency: The highest frequency that a given sampling rate can resolve

Sample rate	Quality level	Frequency range
11,025 Hz	Poor AM radio (low-end multimedia)	0–5,512 Hz
22,050 Hz	Near FM radio (high-end multimedia)	0–11,025 Hz
32,000 Hz	Better than FM radio (standard broadcast rate)	0–16,000 Hz
44,100 Hz	CD	0–22,050 Hz
48,000 Hz	Standard DVD	0–24,000 Hz
96,000 Hz	High-end DVD	0–48,000 Hz

From: http://help.adobe.com/en_US/Soundbooth/2.0/WS41FBD92E-39EA-4eda-B490-EDE8EA1175C8a.html

Praat's default setting (44,100) is just fine.

File Types and Compression in Digital Audio


Lossless
(TIFF)


GIF 256


GIF 64


GIF 10


GIF 4


GIF 2


GIF 2 image
converted back to
Lossless

Once you compress sound, you will never get detail back.

Lossless File Types:

.wav - “waveform audio file format”

.aiff - “audio interchange file format”

.flac - “free lossless audio codec”

Apple Lossless (.m4a)

.shn - “Shorten”. You’ll never see this.

Compressed File Types:

.mp3 - “mpeg-3 audio”

.m4a - “AAC (Advanced Audio Codec”

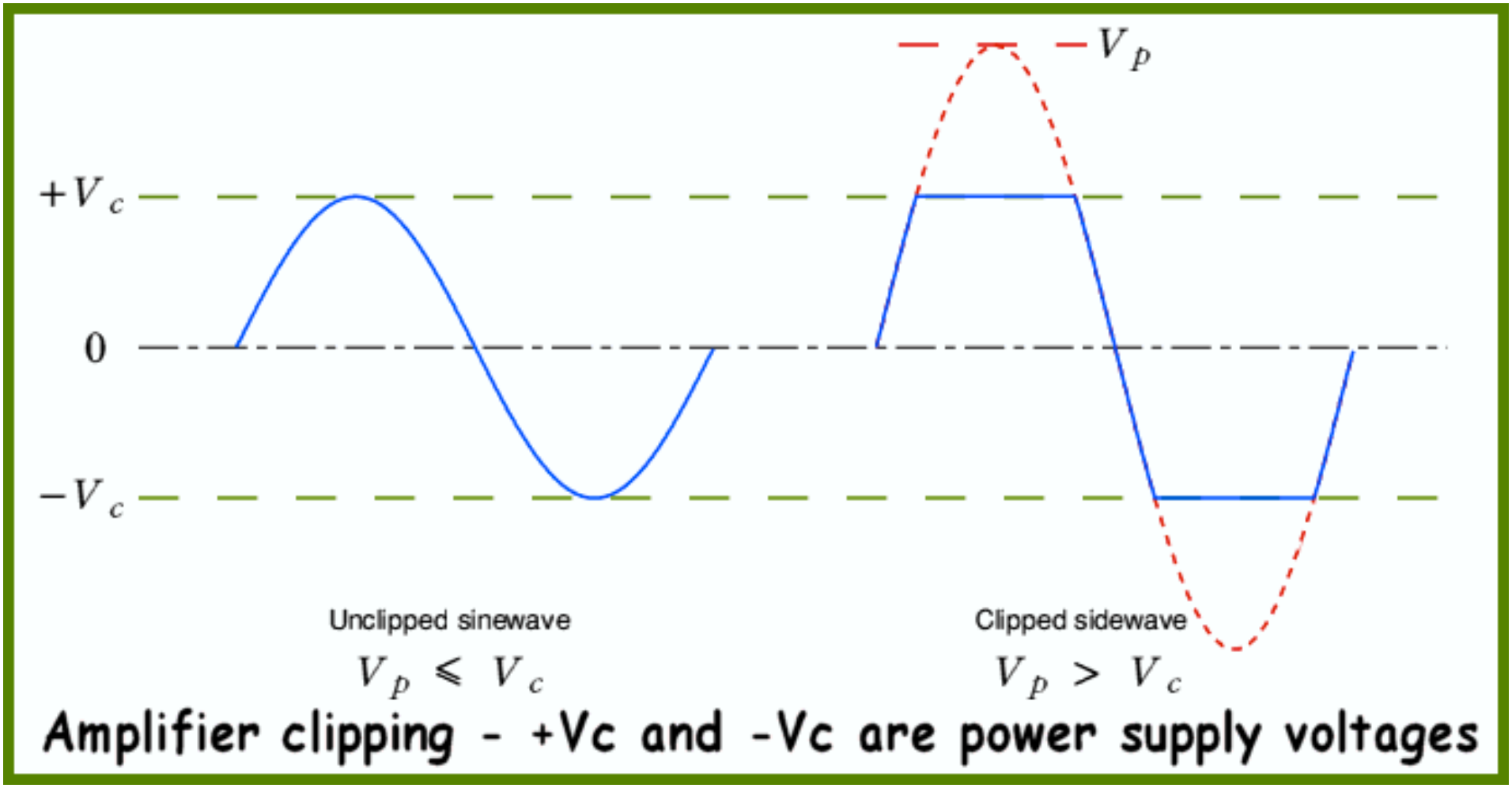
.wma - “Windows Media Audio”

Repeat after me:

Lossless good. Compressed bad.

Use .wav or .aiff

Clipping



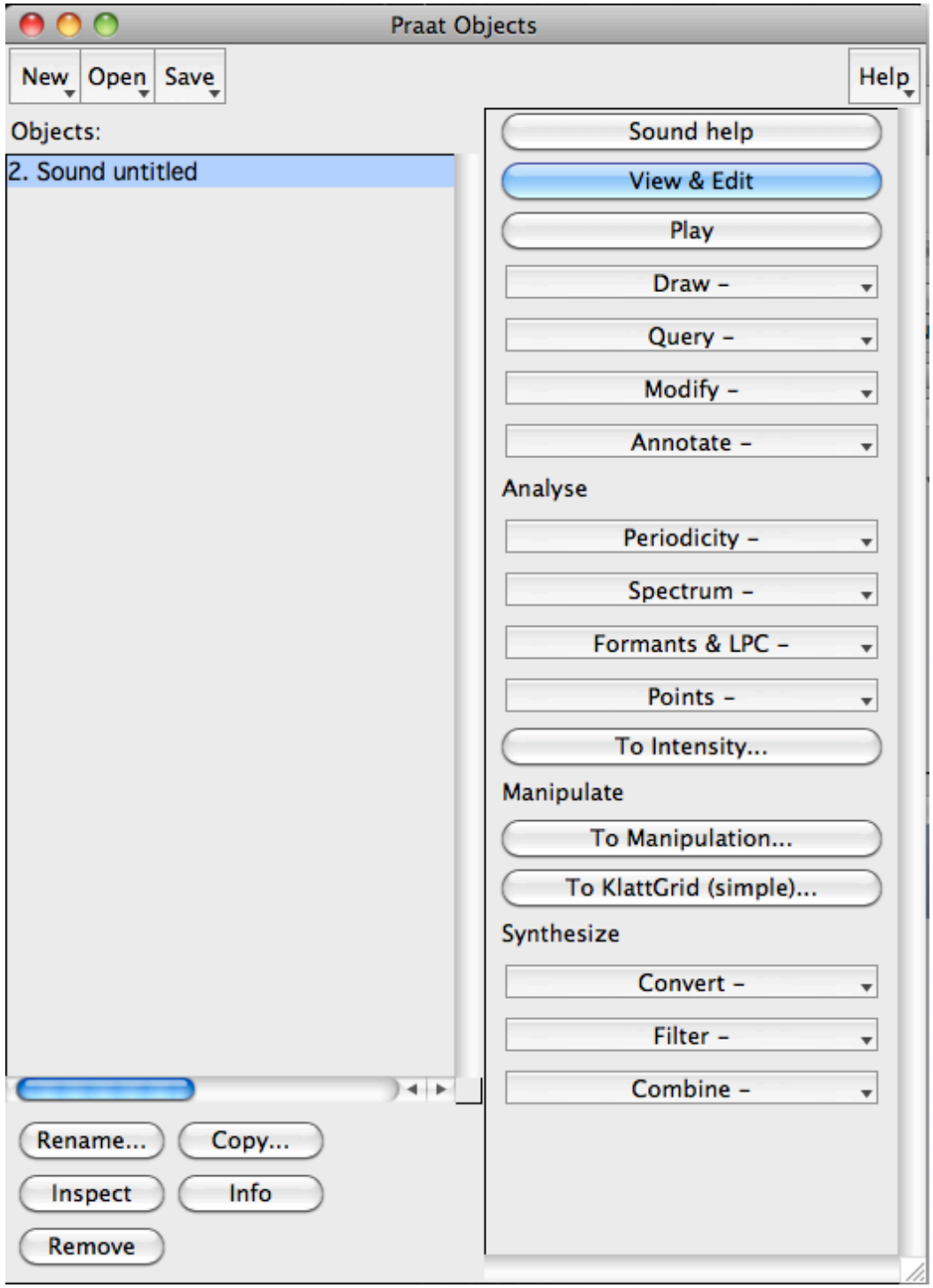
Amplifier clipping - +Vc and -Vc are power supply voltages

Clipping is not a good thing.

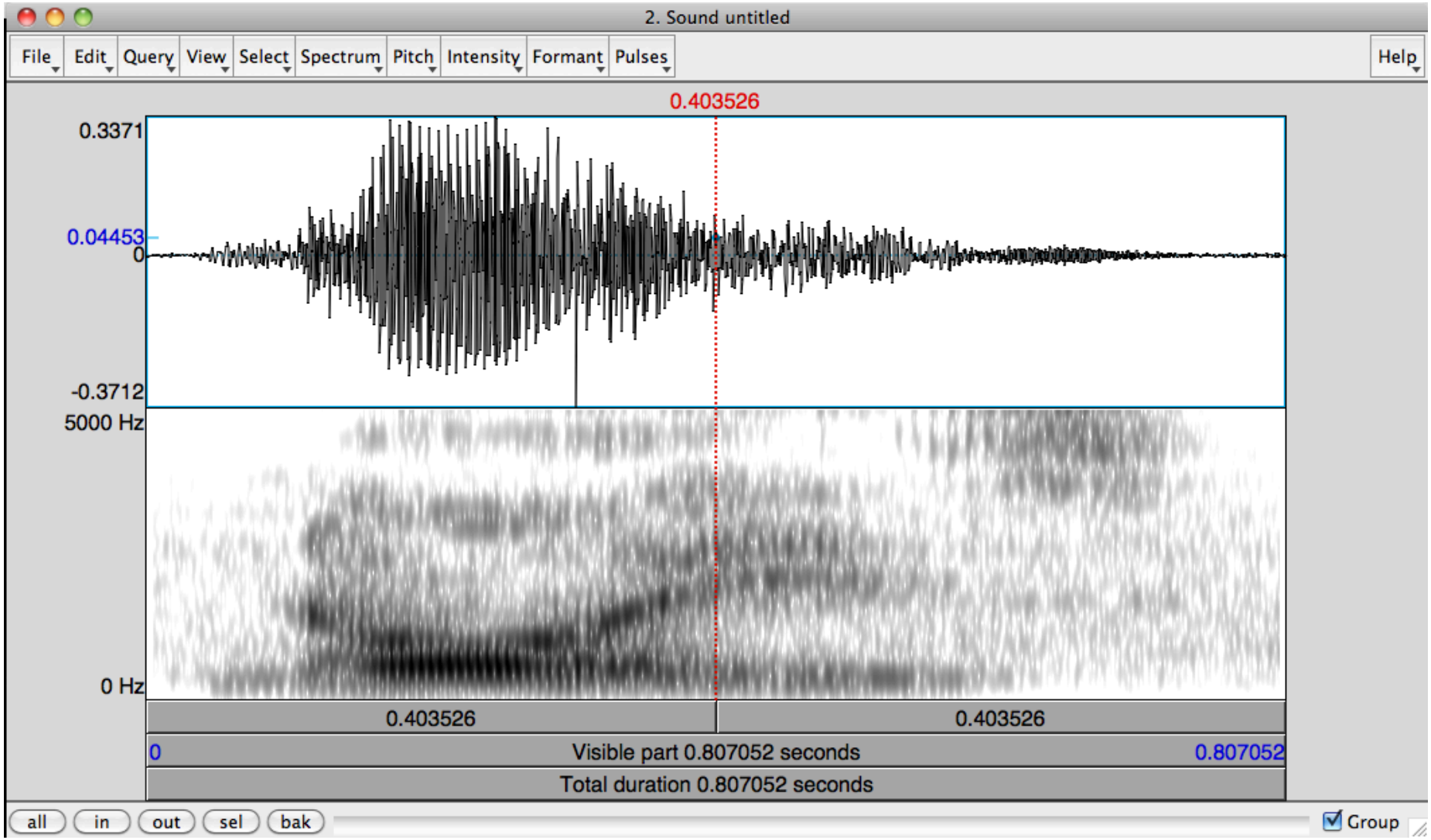
(but Praat will warn you)

Now, it’s Praat time!

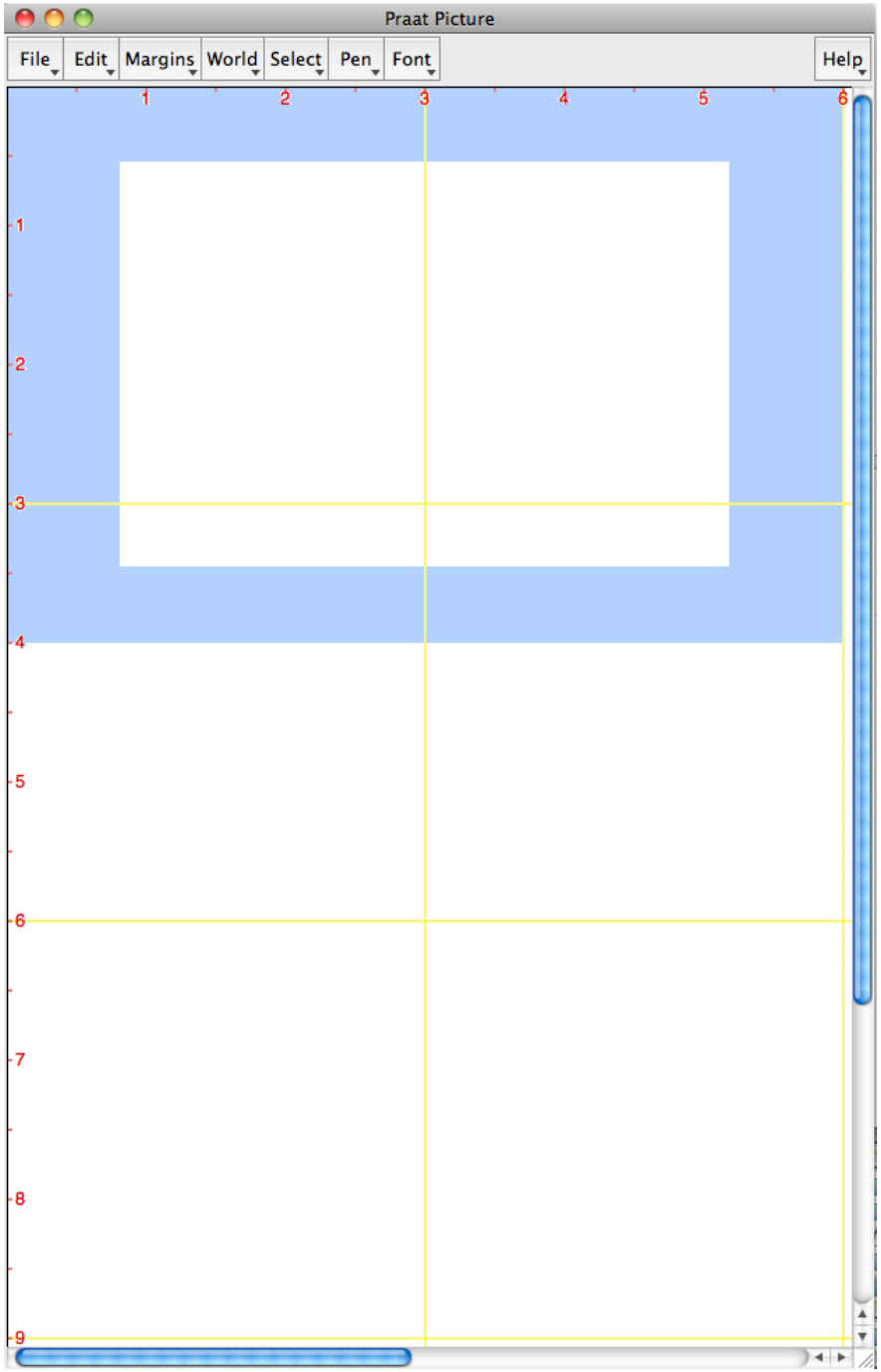
So, what parts of Praat do we need to worry about?



The “Objects” window

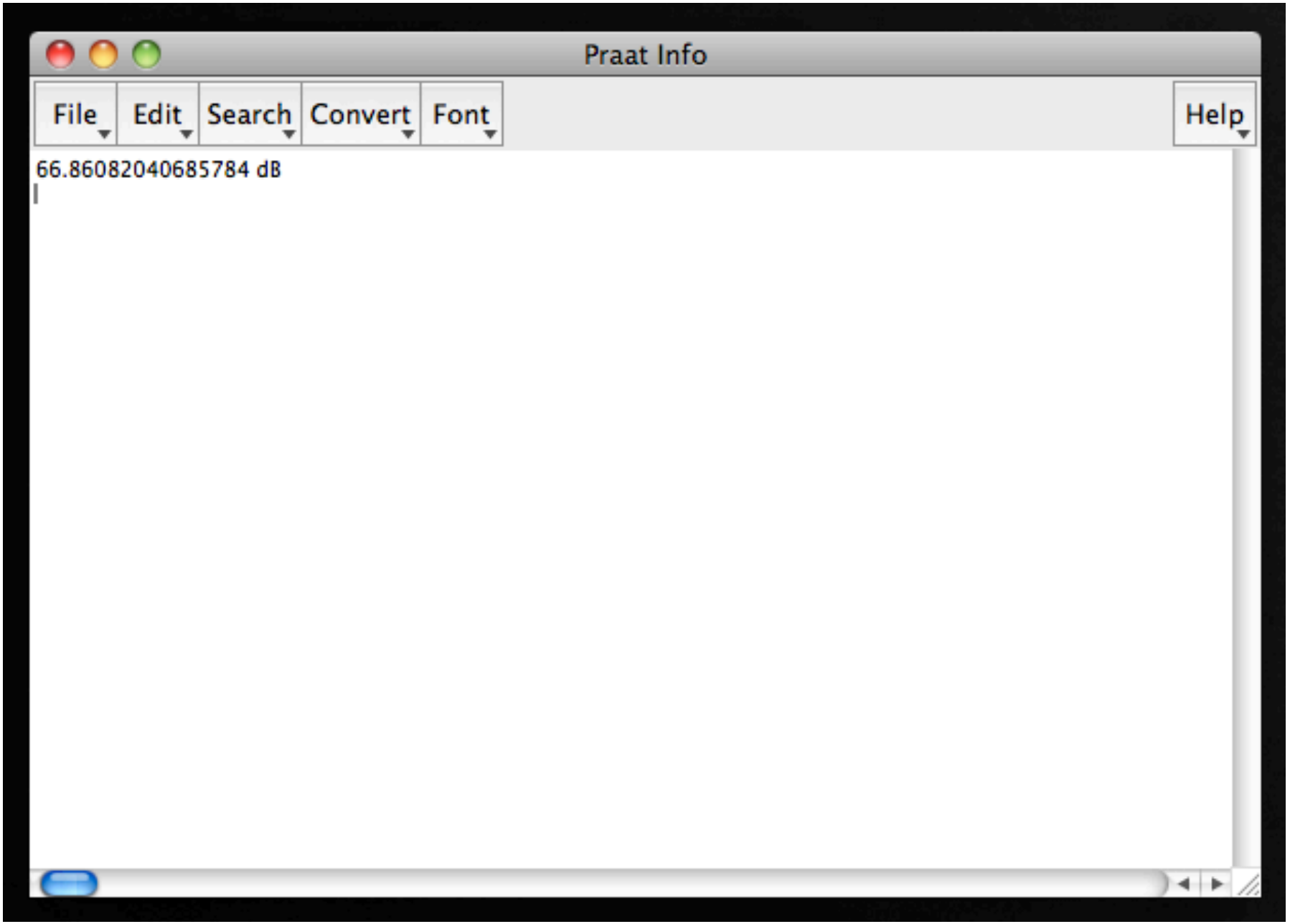


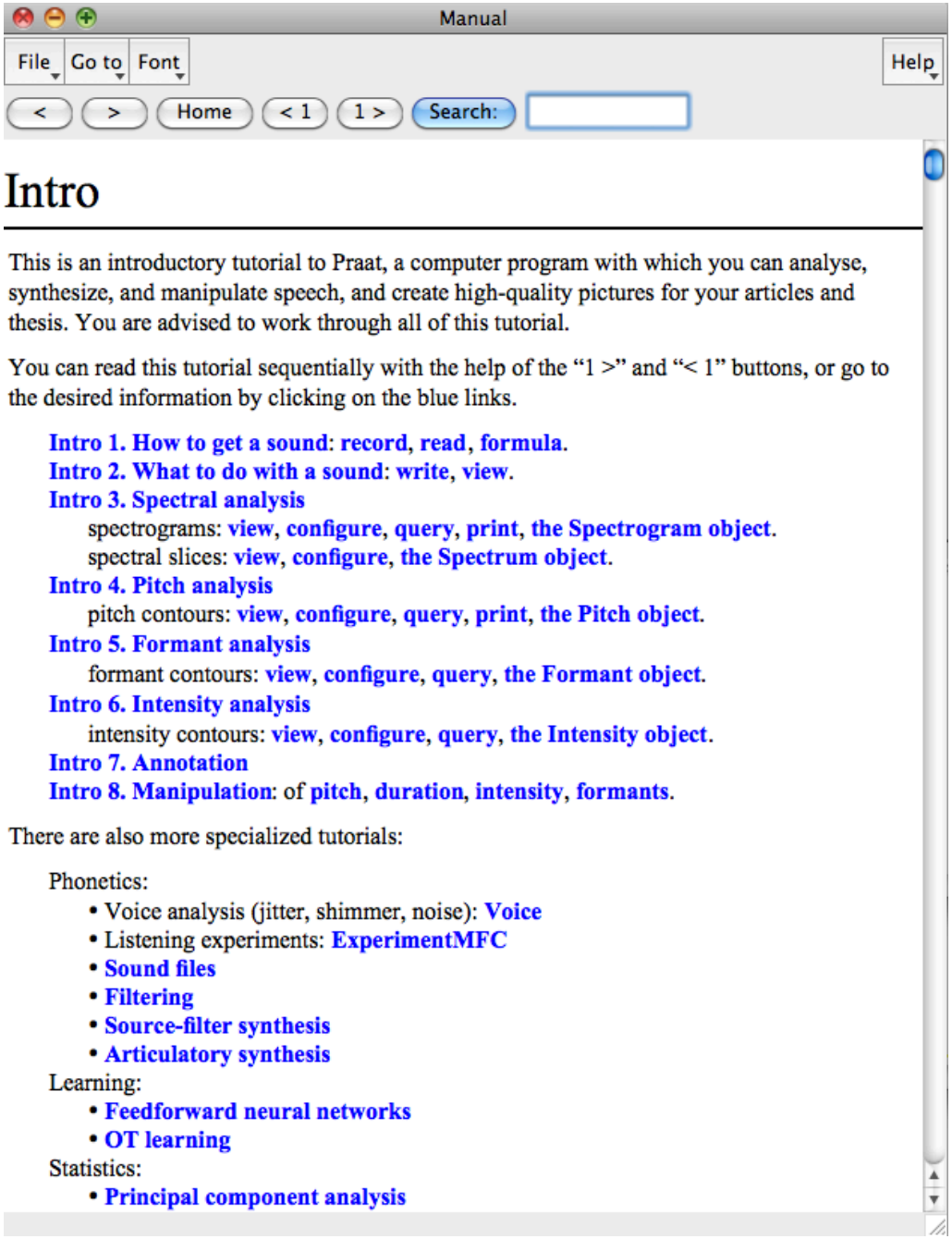
The “Editor” window



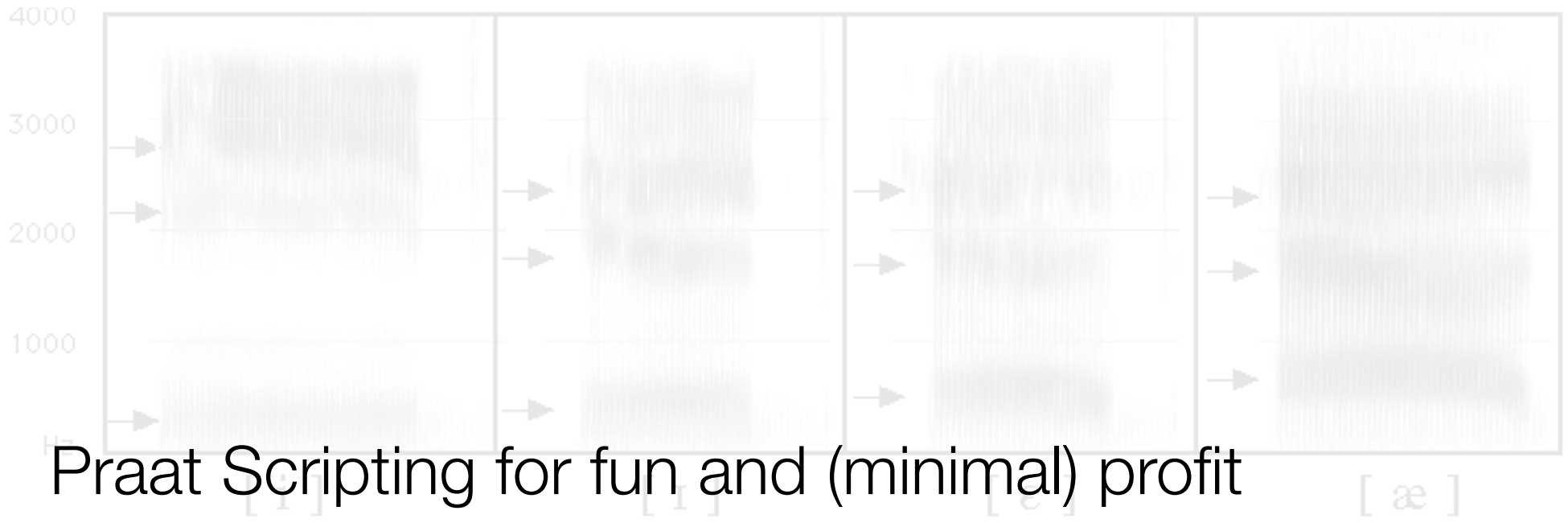
The “Picture” window

The “Info” window

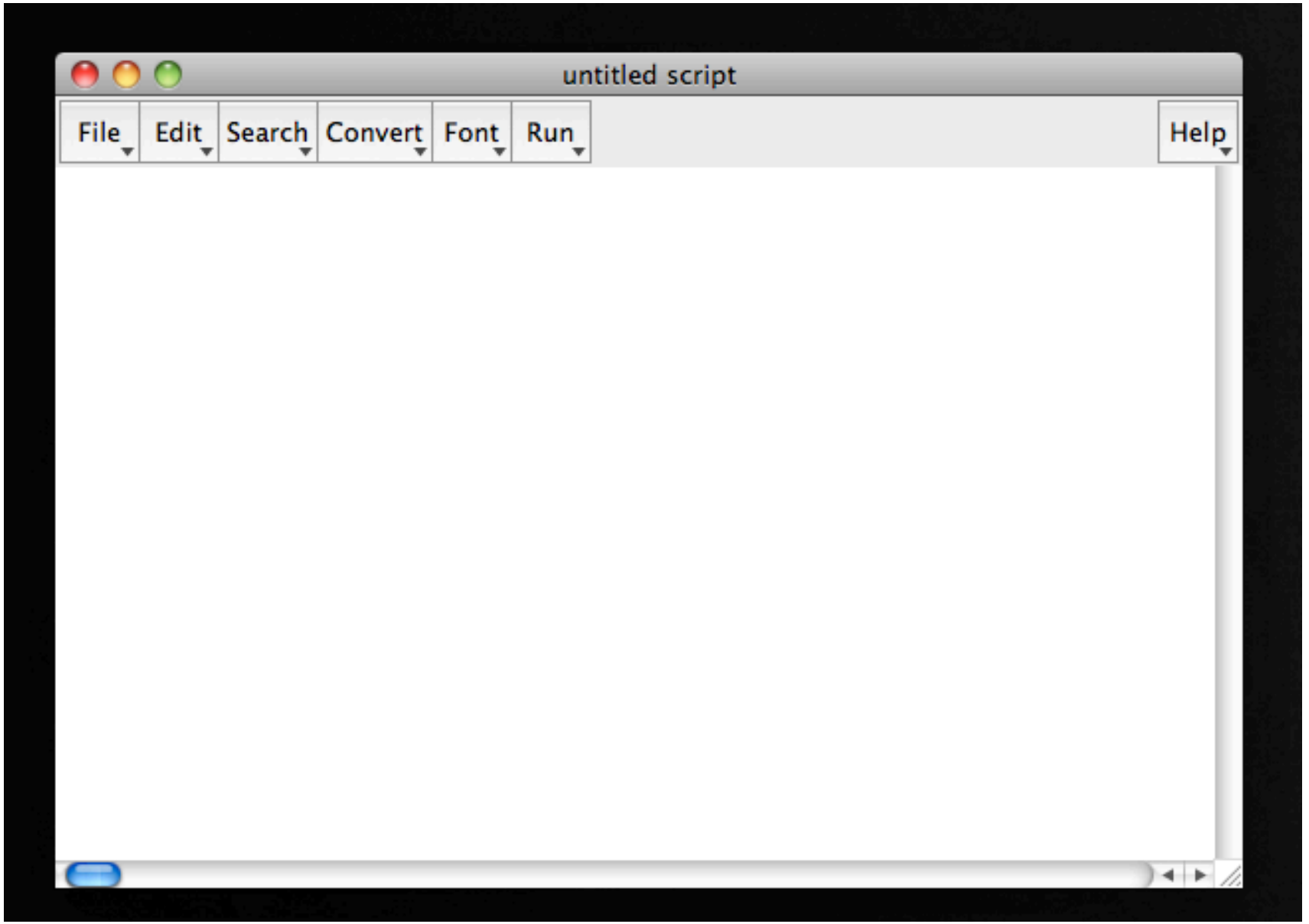




The "Help" window

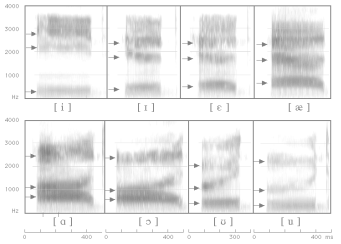


Praat Scripting for fun and (minimal) profit



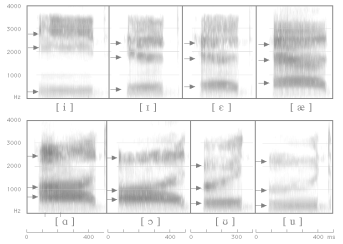
The Script Editor Window

Why script in Praat?



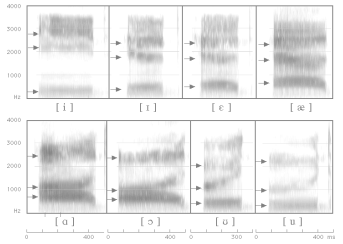
- Let the computer do the boring parts of your measurements
- Be more consistent with your measurements over large amounts of data
- Get your work done faster with a computer co-pilot
- Doing the same thing more than a few times is silly
- You can do anything that you can do through menus in Praat
- **You** are the bottleneck in your data measurement speed

What can’t you do with Praat scripting



- Label your data/speech recognition
- Work with programs that aren’t Praat
- Generate measurements as consistent as hand measurements
- Sanity-check your data and measurements
- Run statistics on your data
- Anything that you can’t, eventually, do through the Praat UI
    - But you can definitely do some things more easily!

What's the catch?



- Humans make sane measurements, computers don't
  - You know that his F0 is not 3000 Hz. Praat doesn't.
  - You can find H1 and H2 easily. It can't
  - You can figure out easily if the formant track is right. It definitely can't.
  - Praat is a black box sometimes
- A single indent can be your worst enemy
- You need to double-check everything, or your data could be subtly wrong

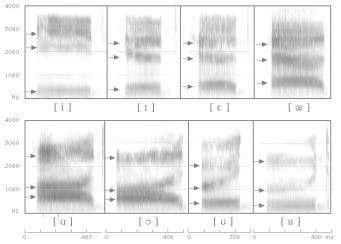
What is a Praat script?

A file with a series of commands for Praat and comments for humans

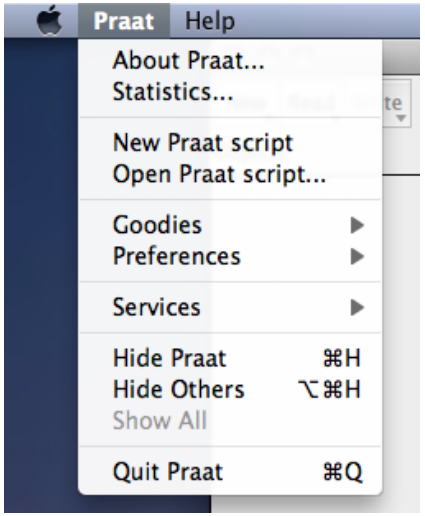
```
# This is a comment, Praat ignores lines that start with #

select Sound untitled
# Plays the sound
Play
# Gets the duration
Get total duration
# Gets the amplitude
Get intensity (dB)
# Renames it
Rename... My_awesome_sound
# Prints the message into the Praat information window
print “Script Finished”
```

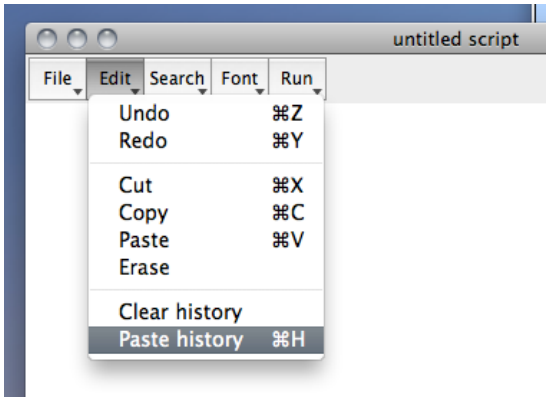
You can just save that to a file, then run it later

Your very first script

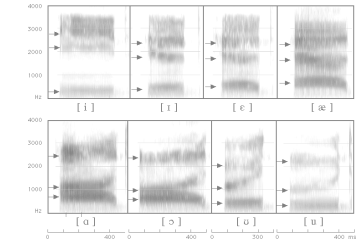


- Do something that you do all the time, in the exact same way...
  - Like making a broadband vs. narrowband spectrogram
- Open Praat, go to “New Praat Script”
- Now, open a sound, and narrow the spectrogram
- Then, go to the script window and “Paste History”
- History saves **everything** you do in Praat



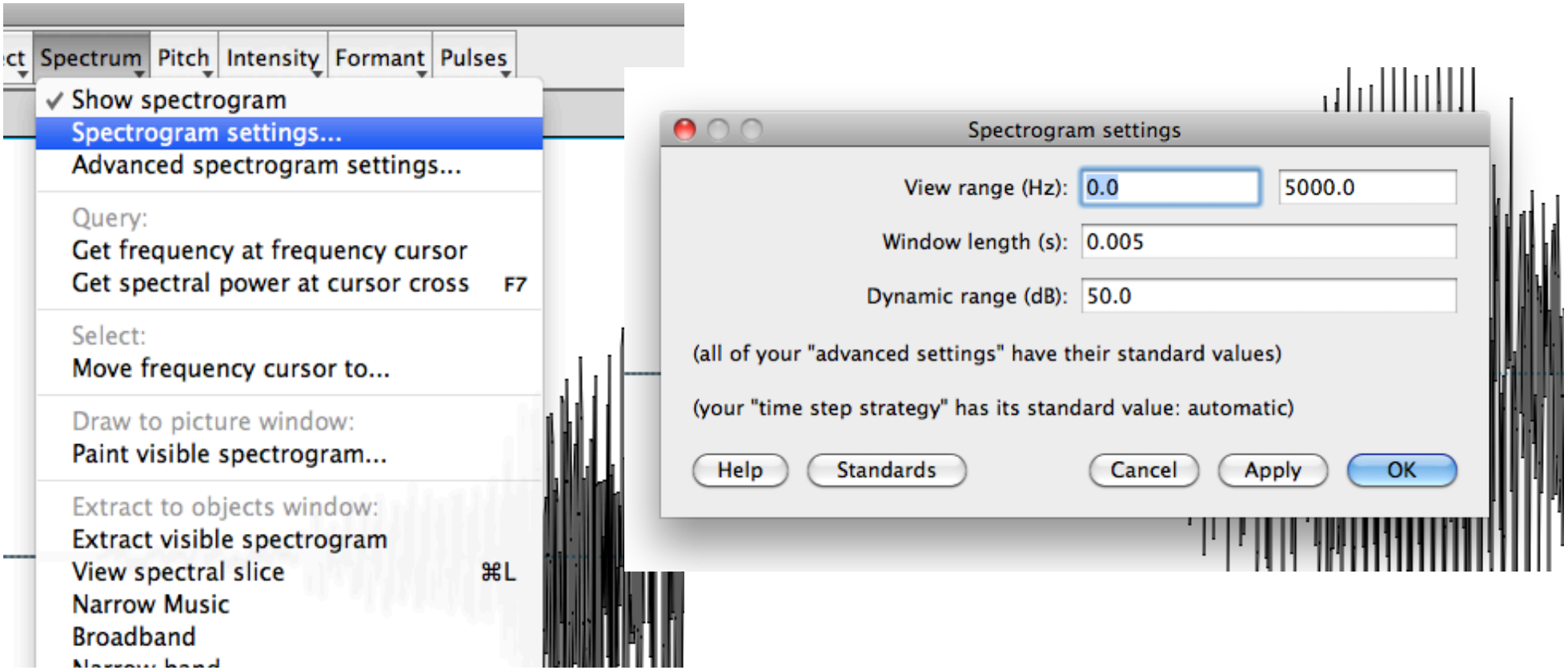


Almost done...



- You now have a few lines, one of which looks like ...
    - `Spectrogram settings... 0 5000 0.05 50`
- That's your first script. Make that line the only thing in the script window
- Now save the script someplace handy, narrowband.praat
- Now, you can add it to any menu as described in Rebecca's handout (Section 3)
- There you go. You just Praat scripted. That was easy, no?

Praat scripting commands are similar to the commands in the UI



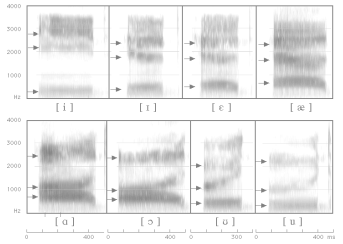
Scripting Command to change these settings:

```
Spectrogram settings... 0 5000 0.005 50
```

Script to make the spectrogram broad again

```
Spectrogram settings... 0 5000 0.025 50
```

Getting a bit fancier

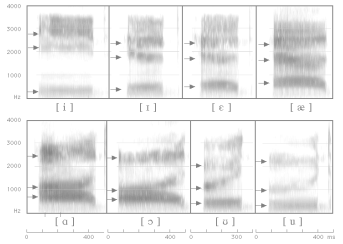


- Sometimes, you want to automate more than one command
- You can put things in a sequence:

```
soundname$ = selected$ ("Sound")
select Sound 'soundname$'
Rename... 'soundname$'_resampled
Resample... 10000 50
Write to WAV file... `soundname$`_10000.wav
```

- Then you just put those lines into a file, and Praat runs through them
- That entire operation (resampling and renaming a sound) can be done with a single command
- But you can get fancier still...

Variables

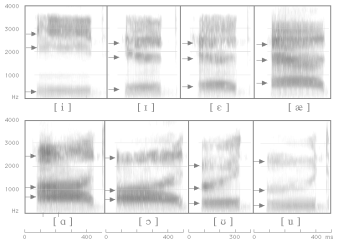


- Assigning variables (`variable = [whatever you want it to be]`)
- Always start with lowercase letter, strings end with $
  - `variable = 1000`
  - `variable = Get number of intervals... '1'`
  - `start = Get starting point... 1 2`
  - `end = Get end point... 1 2`
  - `midpoint = start + ((end - start) / 2)`

Get vocal tract length from F3 at cursor

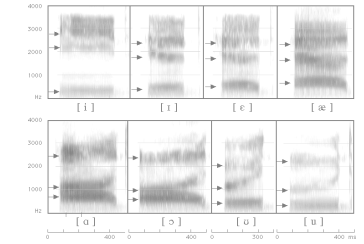
```
f3h = Get third formant
length = (1715/(4 * f3h))
lcm = length * 100
print Your vocal tract length is  'lcm:1'  cm
```

Running your scripts



- Two options:
  - Add it to a menu and use it like another menu option/command as described on the handout in your folders
    - Best for quick scripts you use often
    - Beware, scripts which work in the editor window need to be added to the editor menus
  - Just open the script in Praat and click “Run” in the Script editor window

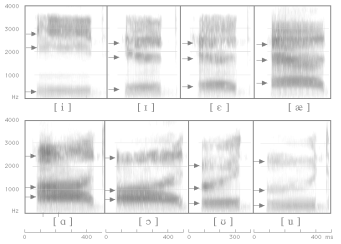
The Button/Skynet continuum



- You can write scripts of various degrees of complexity
  - Some scripts run a single command quickly, and act as a new button
    - Here, the user is completely in control, but it’s relatively slow
  - Some scripts run a large process, but keep the user involved
    - Here, the user yields some control to gain speed
  - Some scripts go through a large process on their own, without your help
    - Fastest, but you don’t control any of the process while it’s running

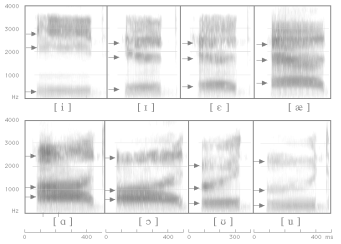
... but sometimes, you need more than just the file(s) you have open

Giving Praat files and information

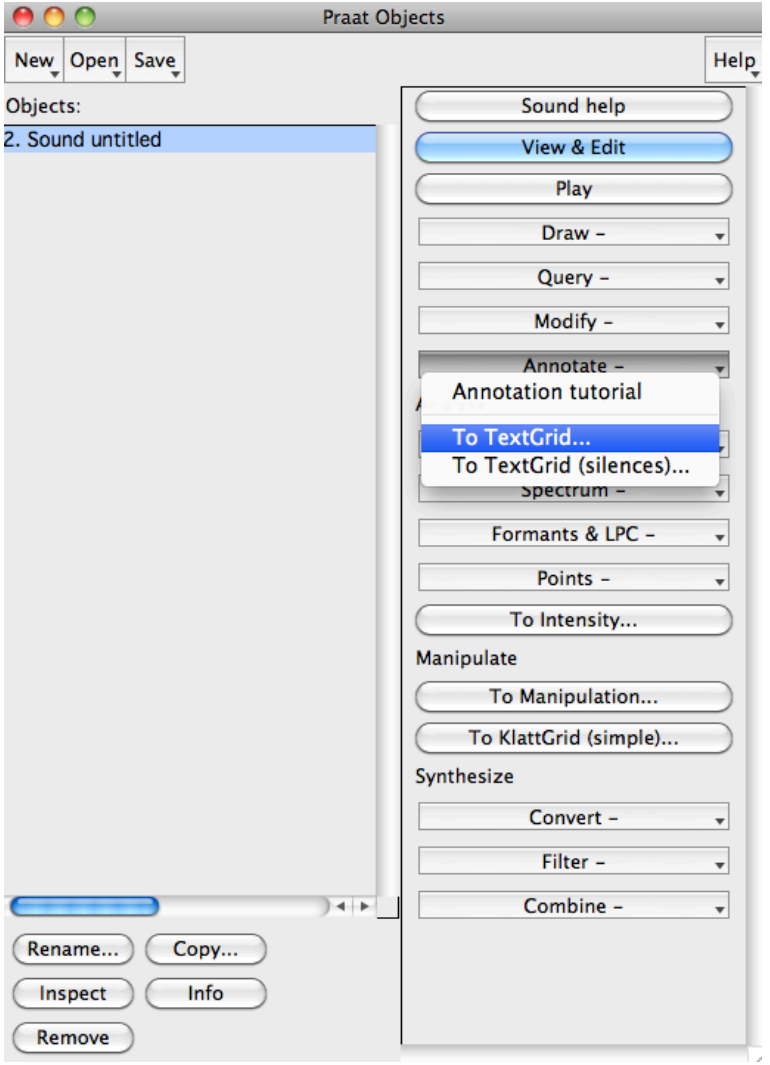


- So, Praat doesn't know what a vowel is, and can't find them natively
- Sound files are annotated for Praat by making "TextGrids". These are companion files that list segments or points, along with their labels
    - Textgrids are saved as different files
- Textgrids have "Tiers", as many as you want, which can be used to annotate different levels of detail
    - You might have a "Word" tier and a "Vowel" tier.
    - Interval tiers mark spans (like vowels), point tiers mark moments (like releases)

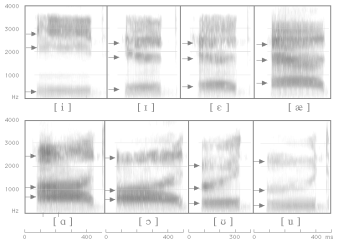
Creating a TextGrid



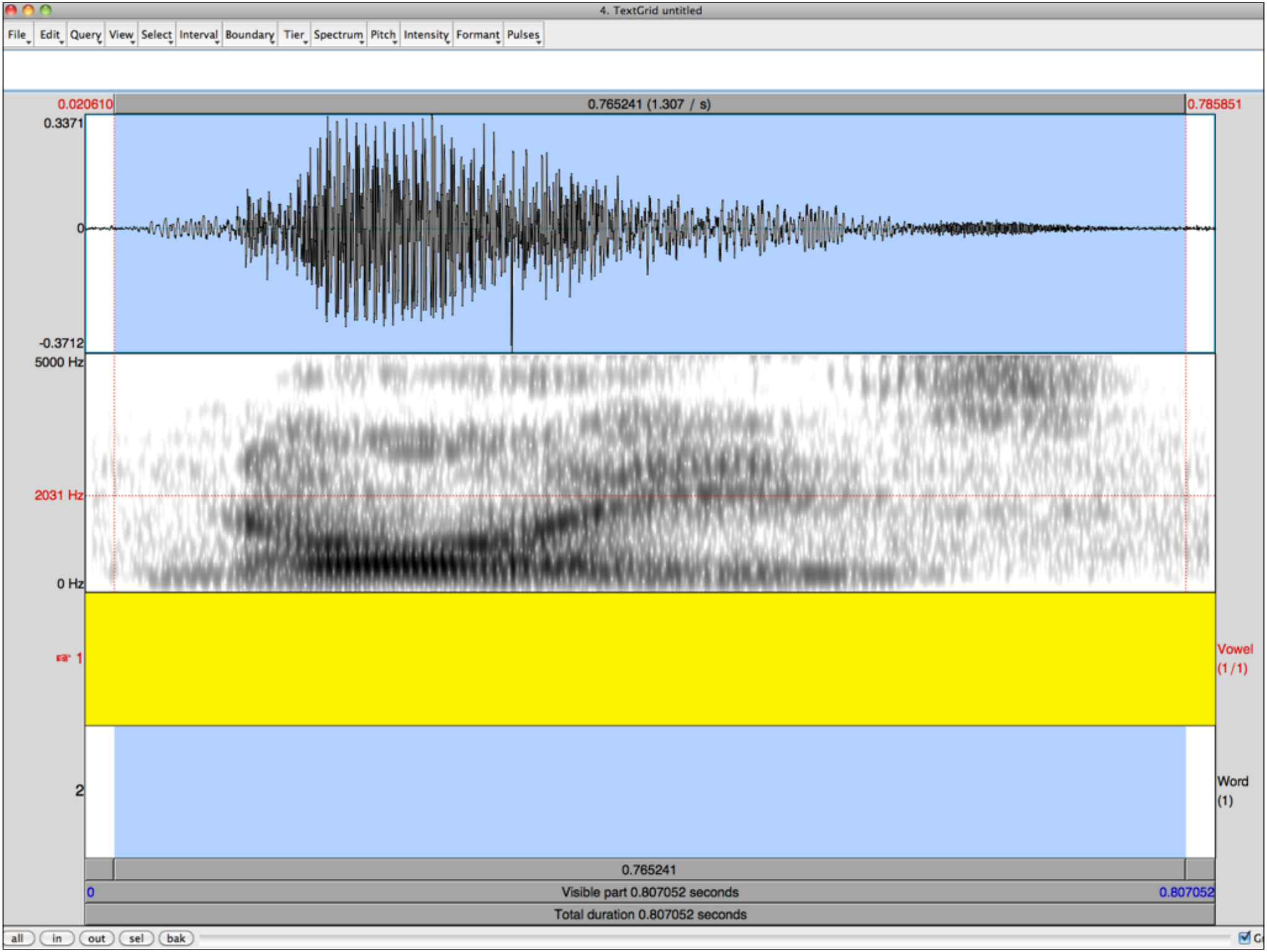
- Open or Record a Sound
- Select the sound in the Objects window
- Annotate -> “To TextGrid...”
- Give names to all tiers you want
- Specify which, if any, are point tiers
- Tiers will be ordered in the order named



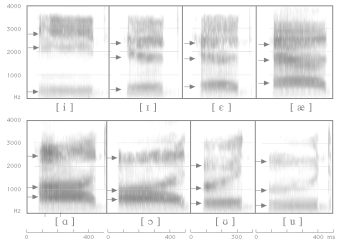
Textgridding, continued



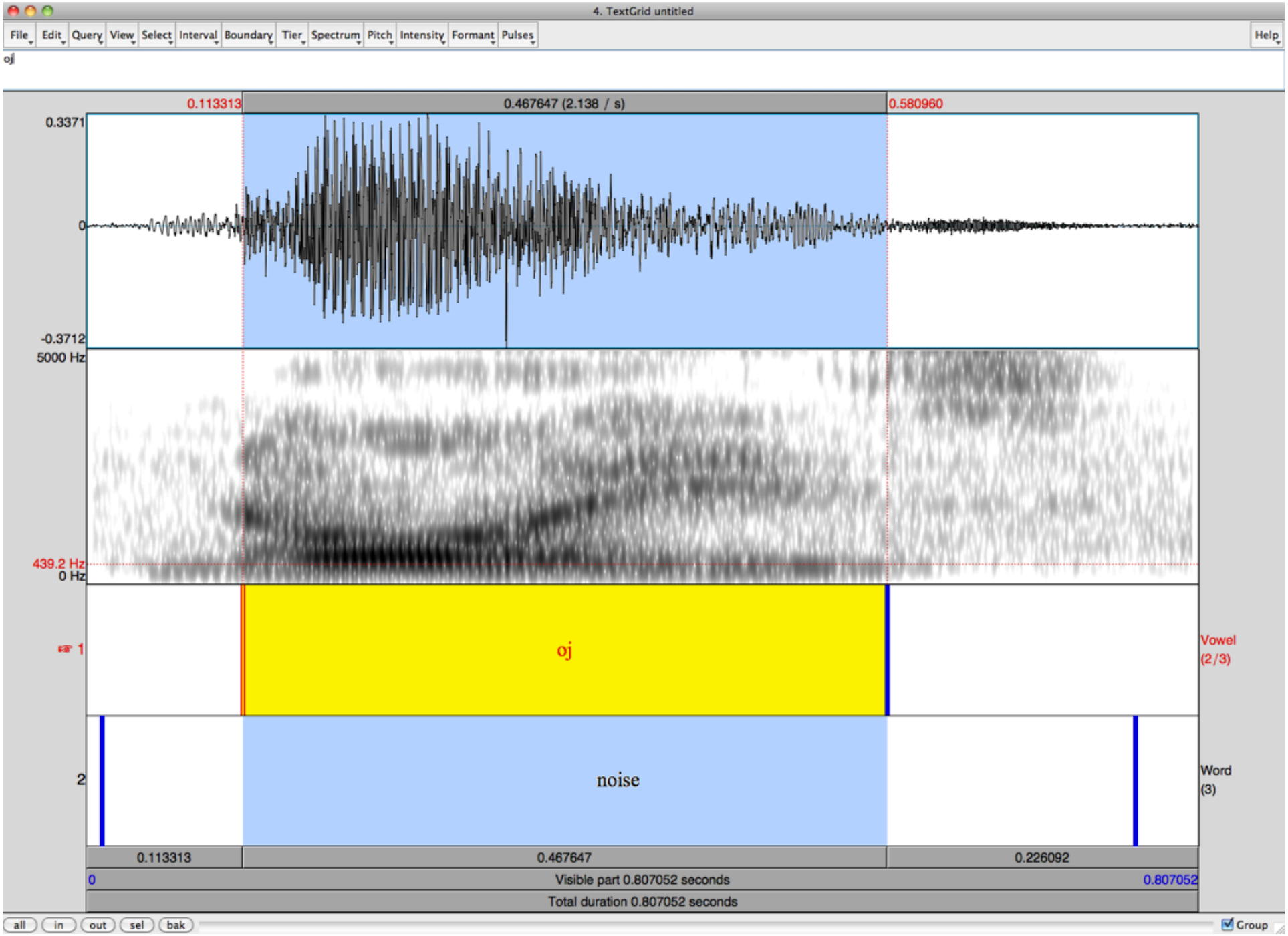
- Select both Sound whatever and TextGrid whatever in the Objects window, and click “View and Edit”


4. TextGrid untitled
File
Edit
Query
View
Select
Interval
Boundary
Tier
Spectrum
Pitch
Intensity
Formant
Pulses
0.020610
0.765241 (1.307 / s)
0.785851
0.3371
0
-0.3712
5000 Hz
2031 Hz
0 Hz
1
Vowel
(1 / 1)
2
Word
(1)
0.765241
0
Visible part 0.807052 seconds
0.807052
Total duration 0.807052 seconds
all
in
out
sel
bak

Textgridding, continued



- Now, click on the tier you'd like to make an interval on
- Select the part of the word you'd like the interval to contain
- Hit "Return"
- Now click your interval and name it
    - You can use IPA here, but you probably shouldn't...
- Repeat for all your tiers and all your words
- Then, if you'd like, you can split your textgrids/sounds into words later
    - Use file_segmenter.praat

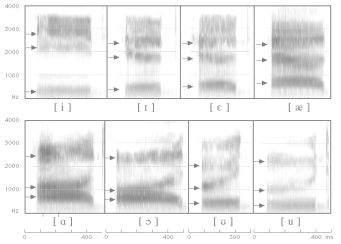

4. TextGrid untitled
File
Edit
Query
View
Select
Interval
Boundary
Tier
Spectrum
Pitch
Intensity
Formant
Pulses
Help
oj
0.113313
0.467647 (2.138 / s)
0.580960
0.3371
0
-0.3712
5000 Hz
439.2 Hz
0 Hz
1
oj
Vowel
(2/3)
2
noise
Word
(3)
0.113313
0.467647
0.226092
0
Visible part 0.807052 seconds
0.807052
Total duration 0.807052 seconds
all
in
out
sel
bak
Group

There are scripts and ways to make this easier/faster.
Talk to me.

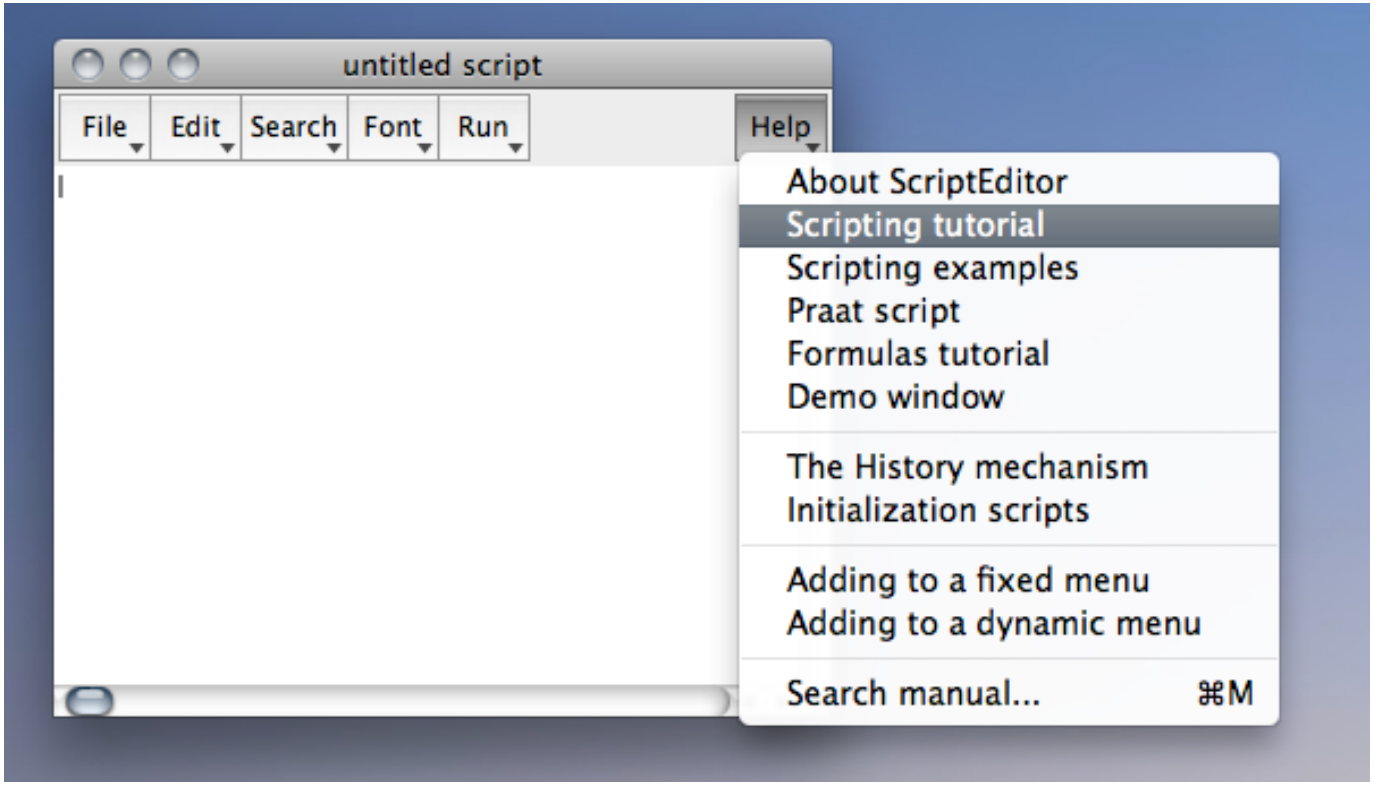
Once you've got files and textgrids, you can then pull in a batch of files and have Praat do something to every one of those files.

Praat Programming Basics

Praat's Scripting Help

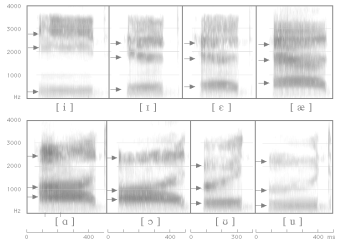


- It's actually really, really good



- ... and all of the following is covered there, more thoroughly

For loops



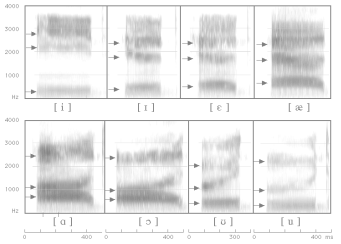
- For loops iterate through large amounts of data, doing the same thing many times over and over again.

- They always have the format `for [var] from 1 to [other var]`, followed by an indented block, ended with an `endfor`

```
select TextGrid 'sn$'
number_intervals = Get number of intervals... 2
for k from 1 to number_intervals
  Set interval text... 2 k Vowel
endfor
```

(This chunk of code changes the text for each interval in tier two of a textgrid to “Vowel”)

if statements

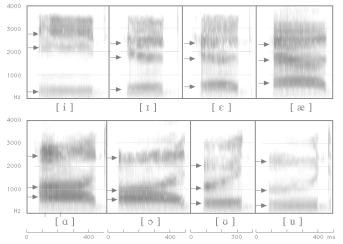


- Sometimes, you want to make sure a given action happens only sometimes
- if statements let you say “Do this only if X”
- Usually take the form `if [var] = [value]`, indented block, `endif`

```
vowel_label$ = Get label of interval... 1 2
if vowel_label = “i”
  Set interval text... 2 2 HighFrontVowel
endif

vowel_label$ = Get label of interval... 1 2
if vowel_label <> “i”
  Set interval text... 2 2 AnotherVowel
endif
```

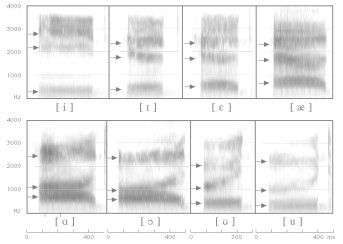
(If the vowel’s label is “i”, set the interval text to “HighFrontVowel”, but if not, set it to “Another vowel”)

Other Residents of Script-land



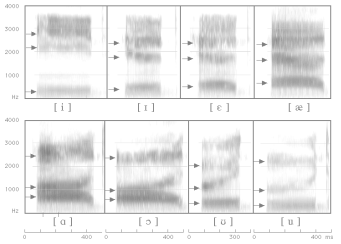
- # at the start of the line indicates a comment, nothing on this line affects how the script runs
- Adding a colon followed by a number to a variable name (`variable:2`) will round the number to that number of decimal places
- The `print` command will print whatever follows it into the info window
  - `print duration` will print the contents of the variable `duration`
  - `print “duration”` will print the word “duration”
  - Making Praat print variables can sometimes help you find where a script is crashing in a horrible mess of code and failure

Cannibalism isn’t so bad

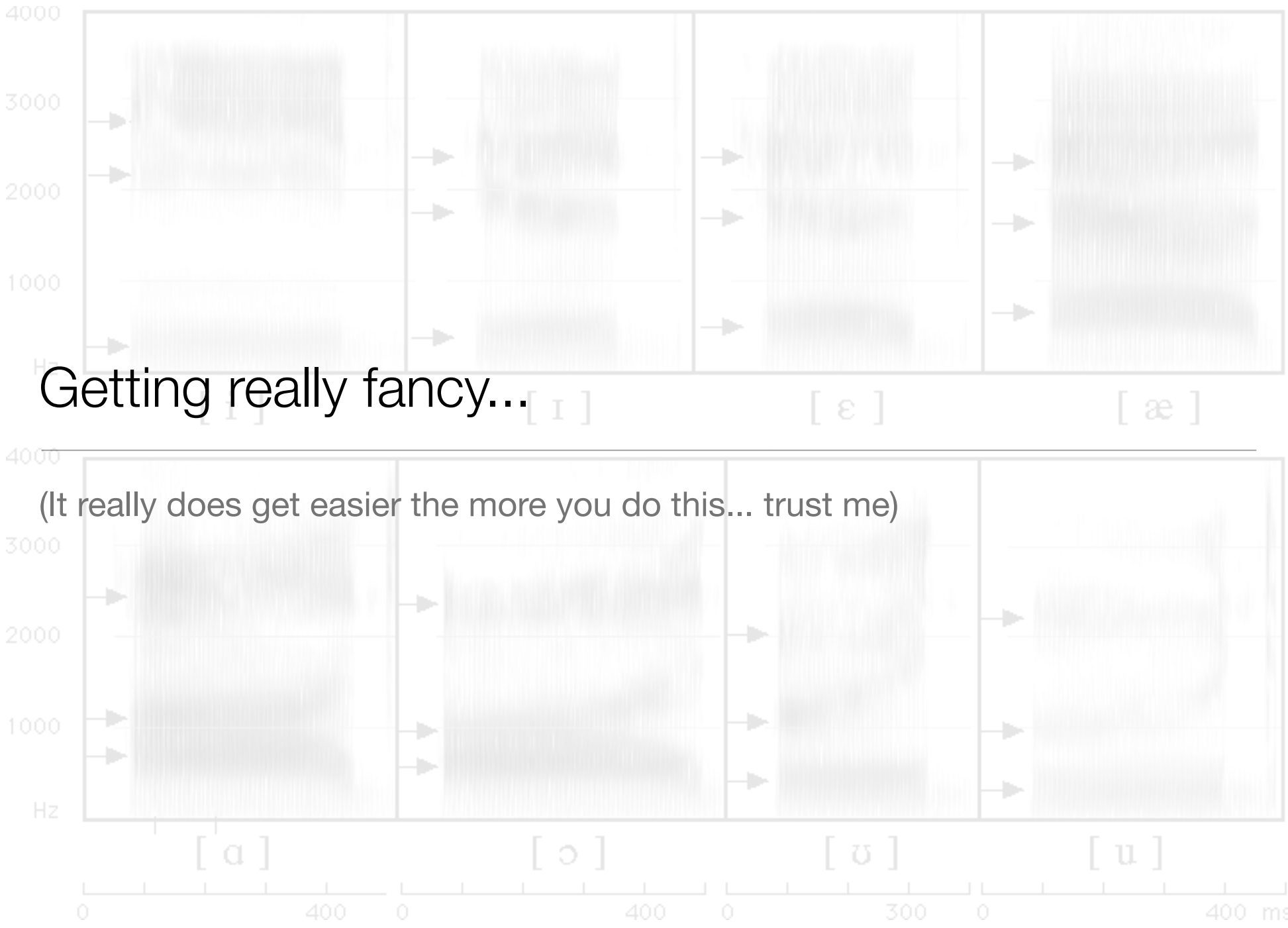


- If you find a script that does 90% of what you want, add the 10% yourself
- Make sure the script does what it says it does, well
- Don’t trust anybody else’s math
    - Especially not mine
- Give them credit with a comment

Praat is odd...



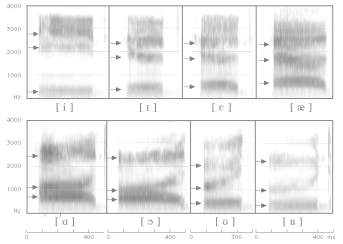
- Avoid long (~20 character) filenames
  - Don’t have decimals in filenames
- The symbol for “not equal to” is <> in Praat, rather than != or something sane
- You’ll have to describe folder/file locations differently on a Mac vs. Windows
  - `directory$ = "c:\Documents\test data\" (Windows)`
  - `directory$ = "/Users/will/Documents/test data/" (Mac)`
- Praat won’t always agree with itself from window to window



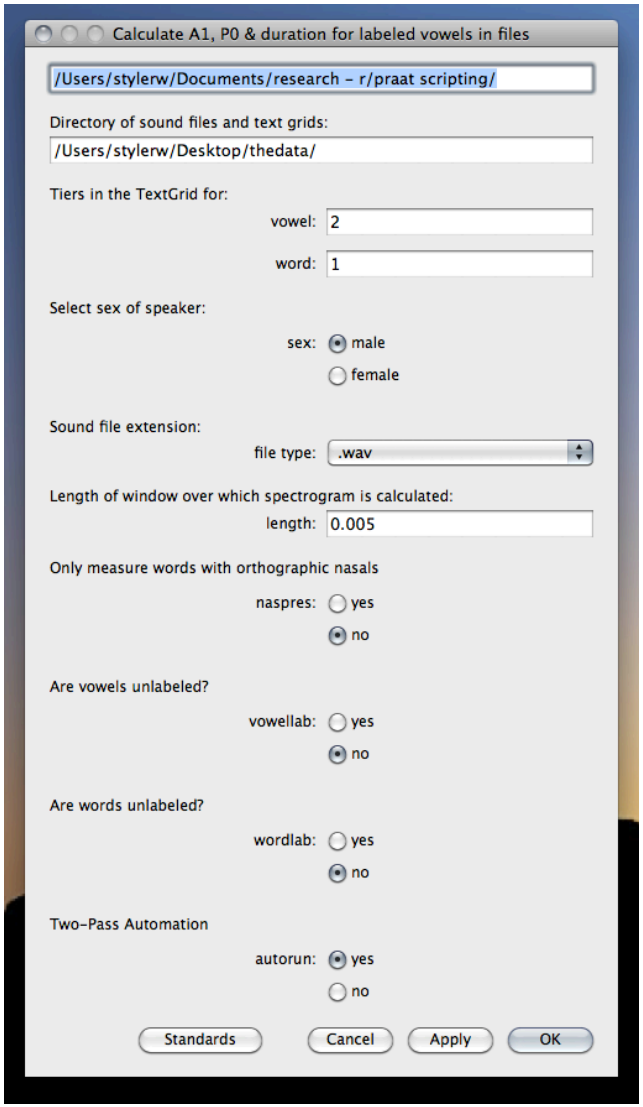
Getting really fancy...

(It really does get easier the more you do this... trust me)

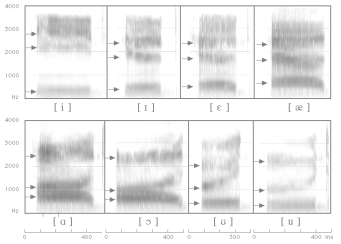
Forms



- Forms show up at the very start of your script
- They allow you to enter information at the start of every script run
- They allow your user not to need to enter paths and variable changes into the script per-se
- They let you make decisions about how the script is run
- More info in the tutorial



Procedures



- If you're doing the same thing over and over in a script, make a procedure of it, and just call that procedure

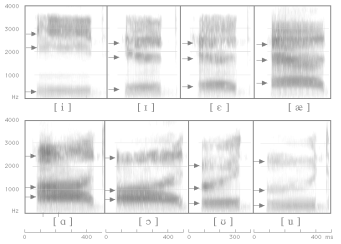
- You can define the procedure anywhere...

```
procedure f0check
   if h1 < 90
      error = 1
   endif
endproc
```

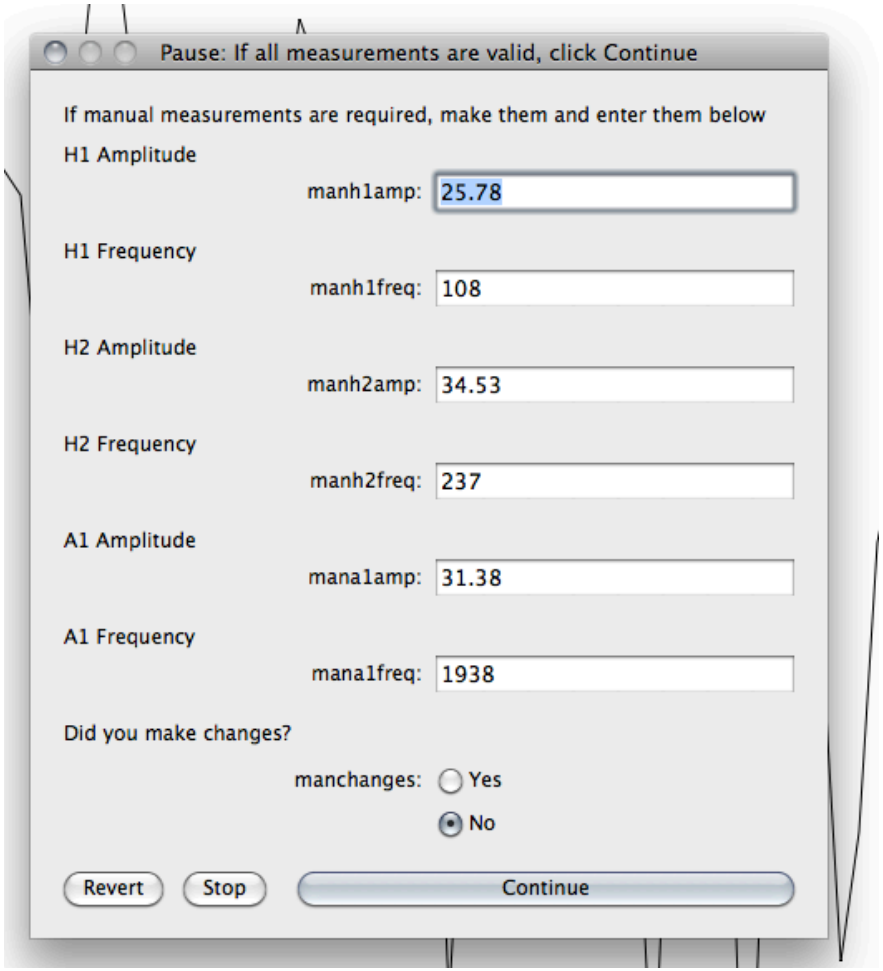
- Then just use it later in the script, even in other procedures

```
h1 = [some variable]
call f0check
if error = 1
  pause
endif
```

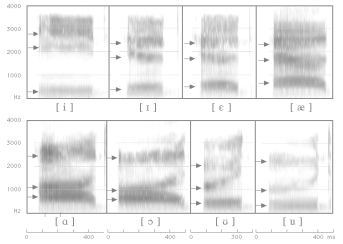
Pause forms



- These allow the script to pause so the user can fill in or confirm data
- This also gives the user a chance to change the course of the script while it's already running
- The script will stop until the user restarts it
- Also listed in the tutorial ("Controlling the User")



Editor Scripting

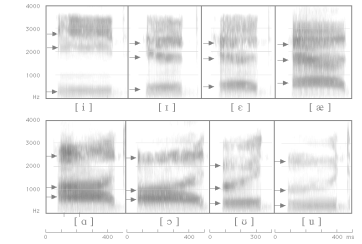


- If you need access to things done in the editor window, you'll need to do editor scripting

- This allows you to do things like cutting and pasting chunks of sound, and using the Formant and Pitch trackers within the editor window

```
select Sound 'soundname$'
Edit
editor Sound 'soundname$'
Select... startvowel endvowel
Cut
endeditor
```

Cleanup

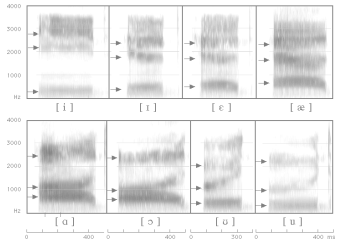


- If you run a large `for` loop, you can end up with thousands of objects in your Praat window if you don't clean up between words

- Throw something like the following in at the end of each file's for loop

```
select all
minus Strings list
Remove
```

Data checks



- Praat doesn’t know what sane measurements are
  - Take the data more times than you need, and only save data that matches
  - Ask it to present you with the data to hand confirm
  - Figure out how Praat fails, and have it detect the failure mode, try again
  - Sometimes, you just need to back up 10 ms or something
  - If you’re feeling dirty, use averaging
  - Post-Hoc checking. Lots of it. Over, and over, and over.

Enjoy a demo, then go forth and script joyously!

... and please feel free to email me when you're having trouble!