Praat Workshop
-Intermediate-

Yuka Tatsumi

Penn State CLS Lab Manager

Overview of Today

1. Introducing Praat Scripting--what is it?
2. Basic syntax
3. Hands-on Practice!!
4. Other Implementations you can use with scripting knowledge

Part 1

Introducing Praat Scripting

What is Praat Scripting?

- Allows you to **automate** any processing you can do with GUI

- It is a **programming language** (a human-friendly one!) just for Praat

- you could use others' script online...but learning how to script by yourself can really help you understand others' scripts,
and apply others' scripts to your data

What can you do with Praat Scripting?

Answer: you can do anything you can do with Praat GUI -- and can handle **a lot of data**!

Analyze speech

- spectrogram
- formants
- pitch
- intensity
- voice quality
- labelling by
    - phonemes
    - words
    - turns...etc.

Manipulate speech

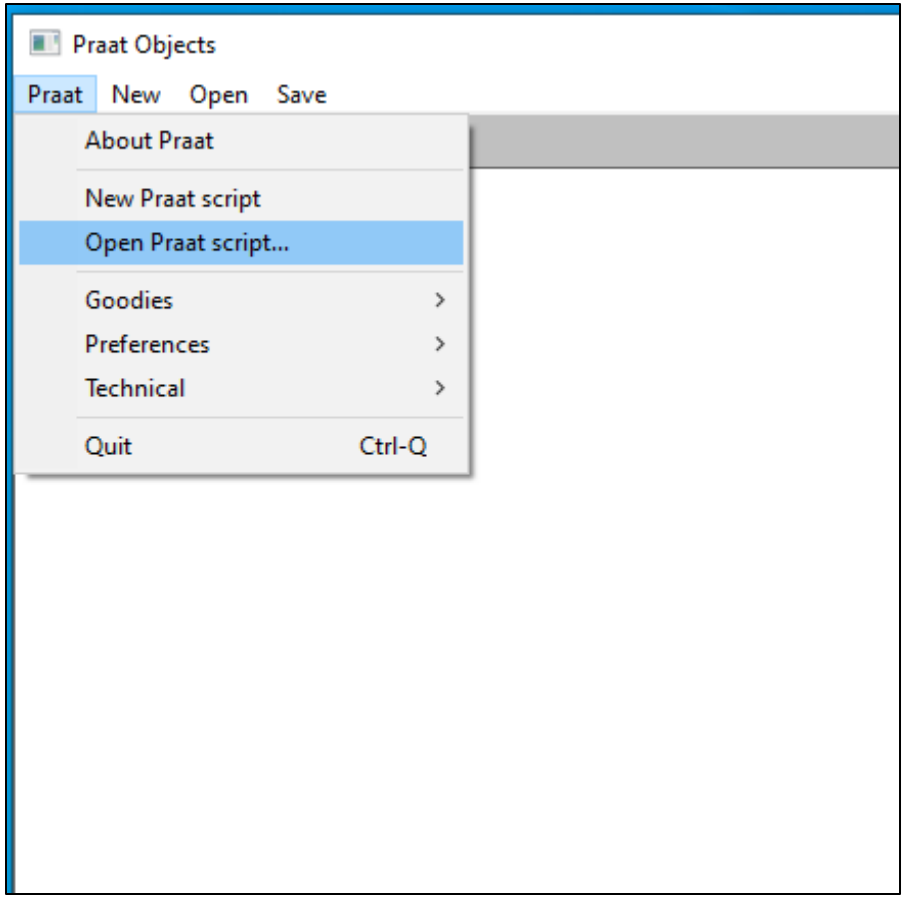
- copy and paste sounds
- change
    - intensity,
    - pitch
    - duration...etc.
- filtering
- synthesize speech

Speech data processing

- Get a table of acoustic measures
- Get graphics of
    - vowel space
    - spectrum slices
    - LPC slices...etc.
- Stats

Let's look at an example script

1. Open an **existing** script



0. Download/copy&paste the script from here

https://github.com/yukat237/Praat_Workshop/blob/main/Saving%20sound%20objects%20as%20wav%20files

praat script to save sound objects.

Script "C:\Users\yzt5262\OneDrive - The Pennsylvania State University\Desktop\BigC annotation\praat script to save extracted sounds" (modified)

File Edit Search Convert Font Run Help

```
#####################################################

# Saving many sound files   Updated: 11/18/2022 Yuka

#####################################################

###READ ME###
#- this script saves sound objects listed in the object window all at once
#- name of each sound is the same as the one appeared on the object window
#- change your directory for your purpose
#- the meaning of objFROM...from which sound in the object window you want to save
#- the meaning of objTO...to which sound in the object window you want to save


#-----------------Form-----------------#

form
        comment Participant Info
                word folderID workshop
        comment Object ID Range
                positive objFROM
                positive objTO
endform

#-----------------Setting Directory-----------------#

directoryInt$ = "C:\Users\yzt5262\OneDrive - The Pennsylvania State University\BIGC_annotations\Round1 annotation\" + folderID$
directoryInt$ = replace$(directoryInt$,"\","/",0)


#-------------------EXTRACTION----------------------#

for i from objFROM to objTO
        selectObject: i
        turnNum$ = selected$ ("Sound")
        nowarn Save as WAV file: directoryInt$ + "/" + turnNum$ + ".wav"
endfor

exitScript: "Done!"

####====================END=======================####
```

General Script Structure

1. Title, Readme, Dates, Rights

2. Body script

2-1. Set up a "**form**" window

2-2. Set up the **directory**

2-3. **Loops**
- "I will repeat the same action below, for this input to that output"
- "import this sound, grab click on that object, do this function"
- "I save that

2-4. Producing products (**Writing** out)

*None of these is obligatory!

praat script to save extracted sounds.

Script "C:\Users\yzt5262\OneDrive - The Pennsylvania State University\Desktop\BigC annotation\praat script to save extracted sounds" (modified)

File Edit Search Convert Font Run Help

1

2.1

2.2

2.3 + 2.4

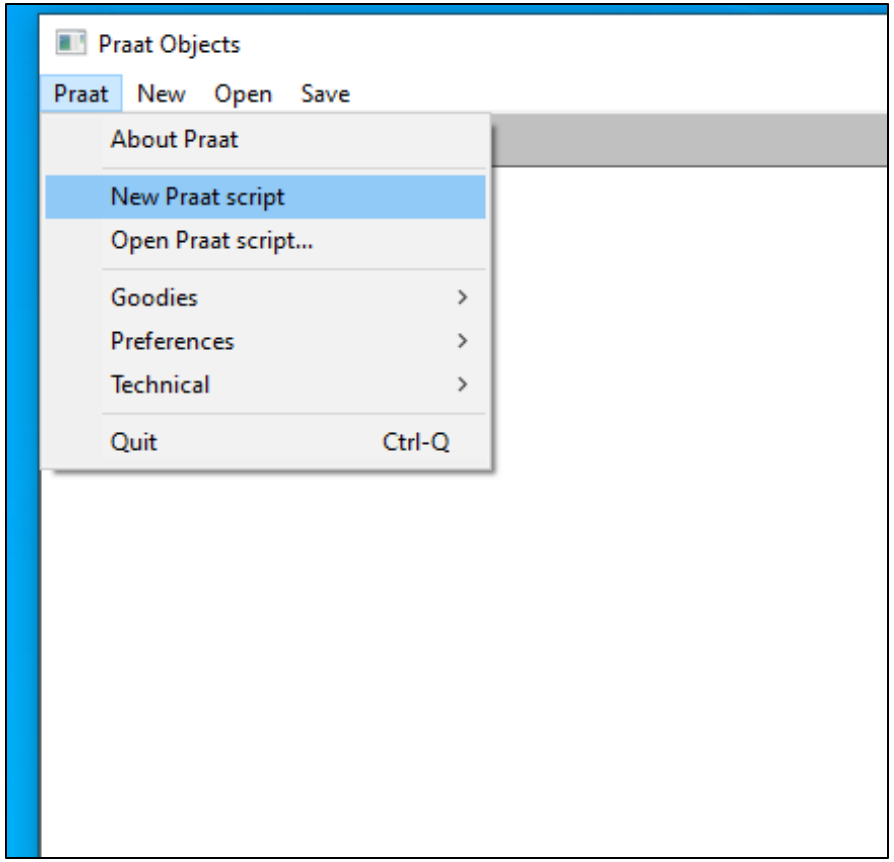
```
#####################################################

# Saving many sound files   Updated: 11/18/2022 Yuka

#####################################################

###READ ME###
#- this script saves sound objects listed in the object window all at once
#- name of each sound is the same as the one appeared on the object window
#- change your directory for your purpose
#- the meaning of objFROM...from which sound in the object window you want to save
#- the meaning of objTO...to which sound in the object window you want to save


#-----------------Form-----------------#

form
        comment Participant Info
                word folderID workshop
        comment Object ID Range
                positive objFROM
                positive objTO
endform

#-----------------Setting Directory-----------------#

directoryInt$ = "C:\Users\yzt5262\OneDrive - The Pennsylvania State University\BIGC_annotations\Round1 annotation\" + folderID$
directoryInt$ = replace$(directoryInt$,"\","/",0)


#-------------------EXTRACTION----------------------#

for i from objFROM to objTO
        selectObject: i
        turnNum$ = selected$ ("Sound")
        nowarn Save as WAV file: directoryInt$ + "/" + turnNum$ + ".wav"
endfor

exitScript: "Done!"

####====================END=======================####
```

Part 2

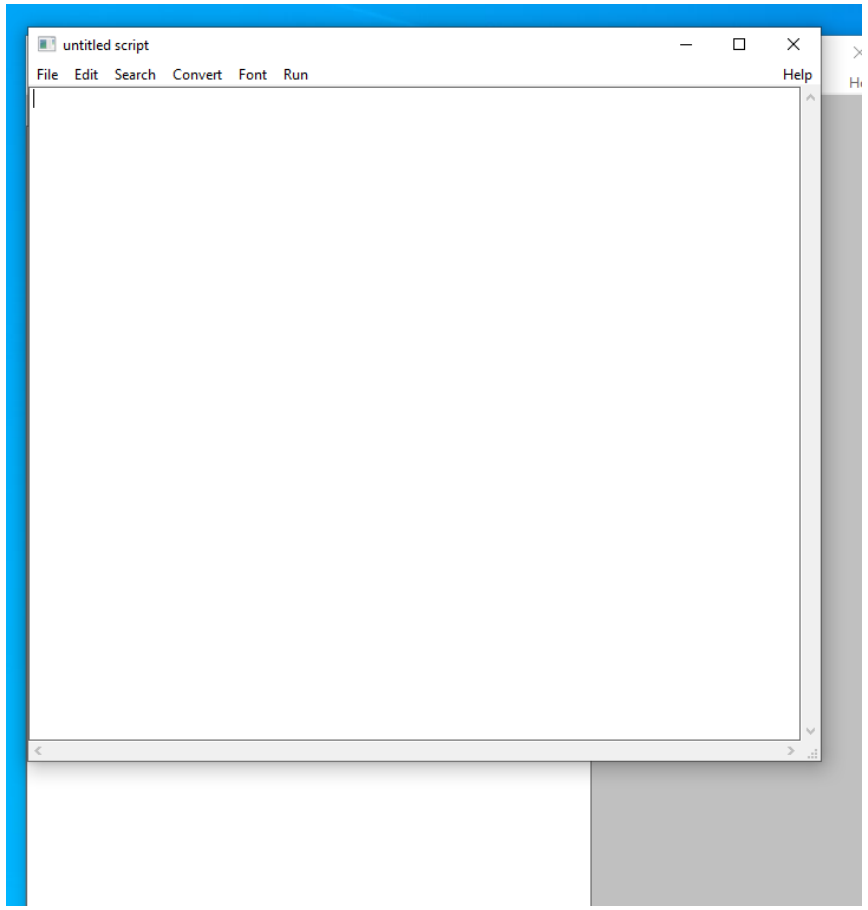
Basic Syntax

Getting started with your new script

1. Open a new script

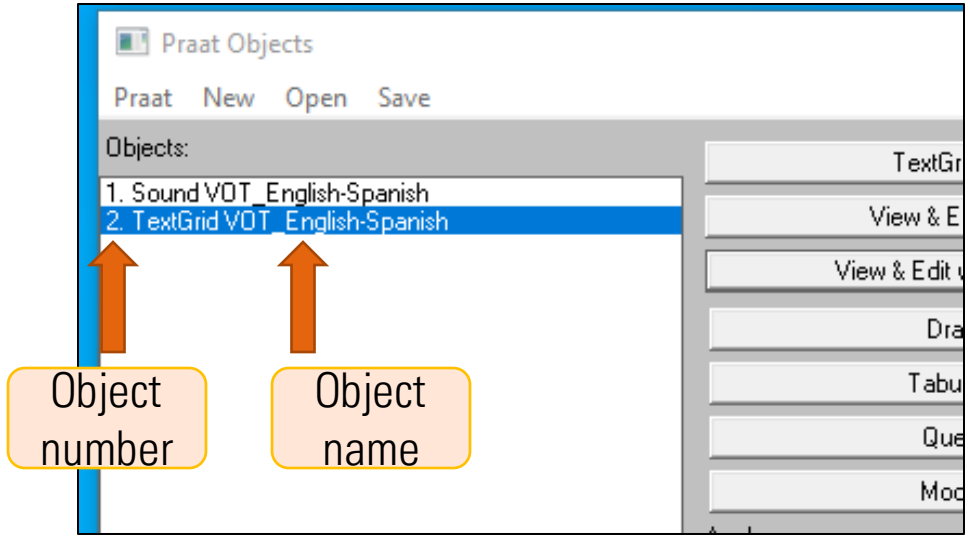


2. this is your canvas!

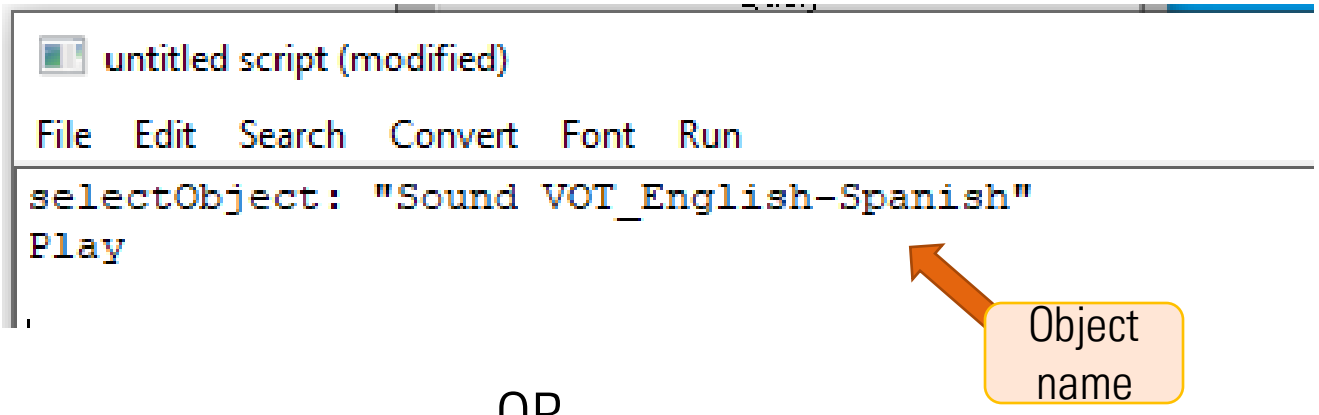


Let's write a simple script (Play Sound)

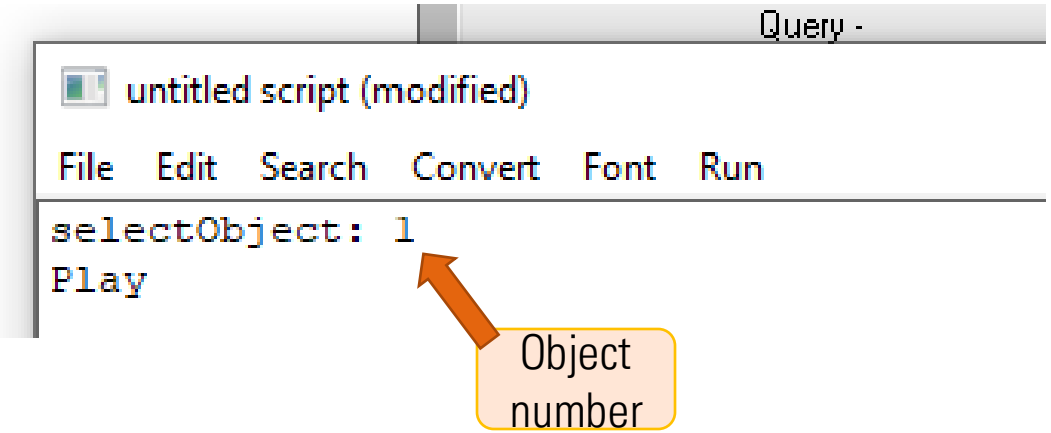
Prep: Import a sound to Object Window
(VOT_EnglishSpanish.wav & .TextGrid)



1. Write this and Run (Ctrl + R)

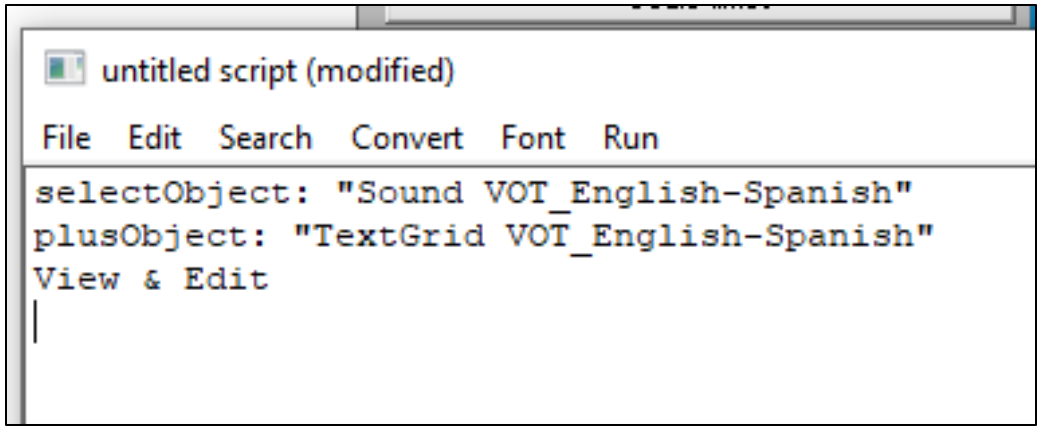


OR

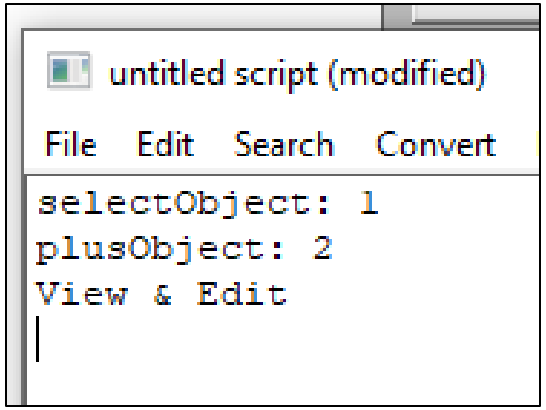


Let's write a simple script (View&Edit, and more)

2. Write this and Run (Ctrl + R)

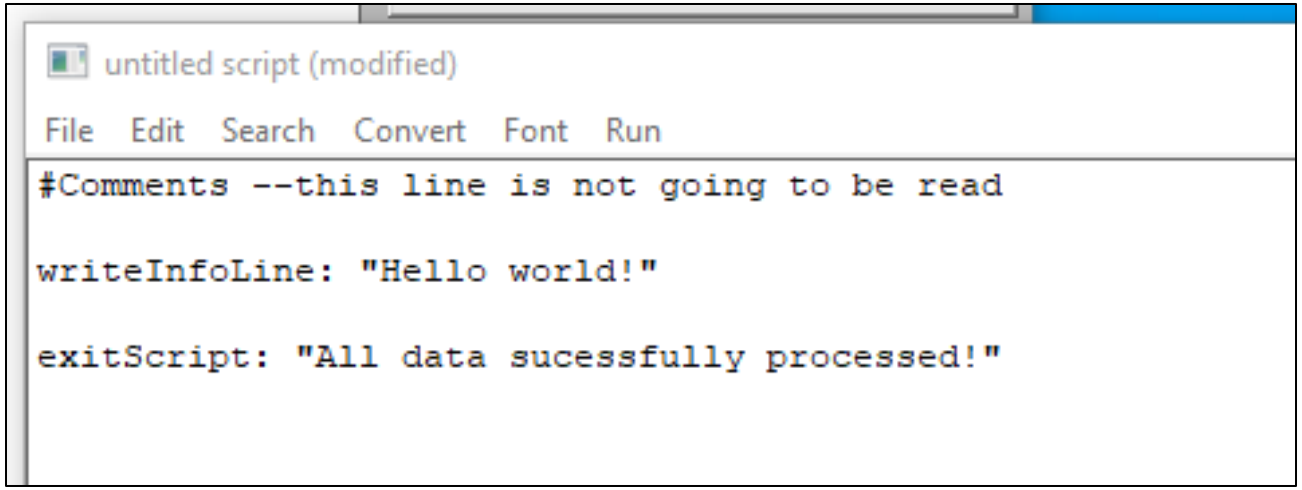


OR

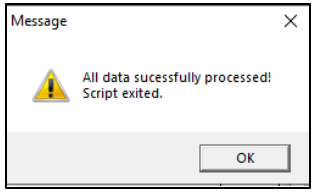


3. Use "#" for comments.

4. "Print" function in Praat is "writeInfoLine: "
(but I use "exitScript: " more often)



WriteInfoLine...opens another script (as a text file)
ExitScript...pop-up message



Let's write a simple script (Calculations)

1. Write this and Run (Ctrl + R)

untitled script (modified)

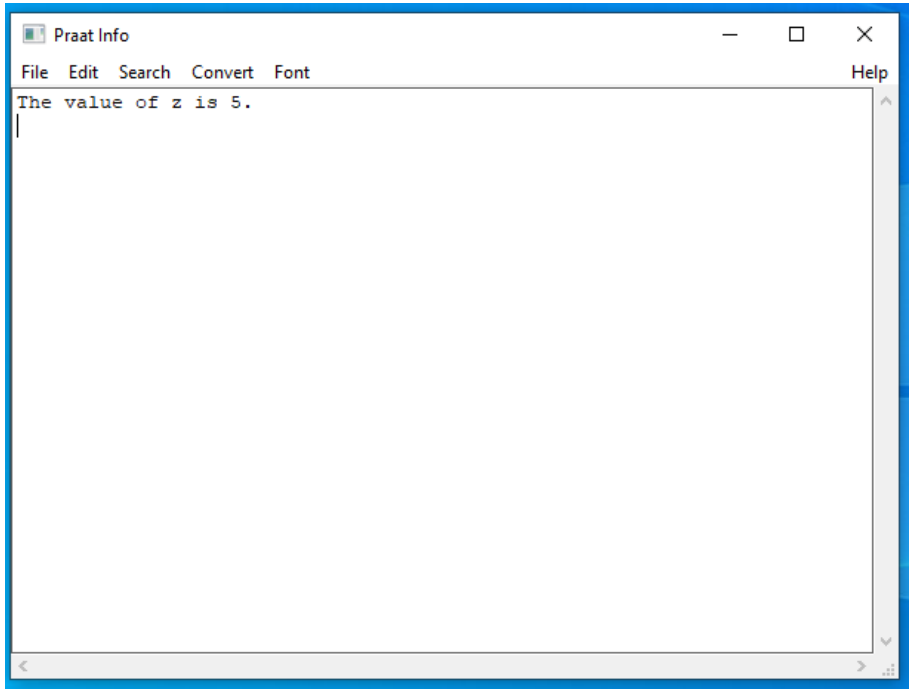
File Edit Search Convert Font Run

```
x = 3
y = 2
z = x+y

writeInfoLine: "The value of z is ", z, "."
```

Terms:

You assigned the value of **3** to the variable **x**.

2. this script will produce:



Let's write a simple script (Texts)

1. Write this and Run (Ctrl + R)

untitled script (modified)

File Edit Search Convert Font Run

```
dates$ = "Dec08"
participantID$ = "23"
filename$ = dates$ + "_" + participantID$

writeInfoLine: "This file name is: ", filename$
```

2. this script will produce:

Praat Info

File Edit Search Convert Font Help

```
This file name is: Dec08_23
```

Summary of Variables in Praat

"letters" … **String** Variable

Variable names end with "**$**".
Assigned values are marked with quotes (" ")

"numbers" … **Numeric** Variable

Variable names end as it is.
Assigned values are the bare numbers.

General Rules:

- All variables must start with lower-case
- Define variables with an equal sign
- functions can follow after "*variable* = "

Praat is case sensitive!

Praat is not whitespace sensitive.
(ignores spaces & indents)
Indentation is used sorely for readability!

Save the script frequently!

Loops

Meaning→ Repeat the code wrapped within "for" and "endfor," for the known number of times.

Basic Idea

```
for i from 1 to numberOfFiles
        BODY CODE TO REPEAT
endfor
```

This code means:
"Repeat 'BODY CODE TO REPEAT' from i = 1 to i = the number value in the variable 'numberOfFiles'

"i"...this is an empty box to save which number you are at when repeating the loop.
this can be any alphabet. It is "i" by convention.

Another option: for i to *variable* (starts from i = 1)

Let's try running a loop

Readme

Extract sounds based on tier 1 (by Word segmentation)

Get unique IDs of 1st and 10th words

Prep an Info window to write results on

FOR LOOP

```
#---CHECK--------------
# Make sure you have Sound VOT_English-Spanish & TextGrid VOT_English-Spanish
# in your objects window.
#----------------------

selectObject: "Sound VOT_English-Spanish"
plusObject: "TextGrid VOT_English-Spanish"
Extract non-empty intervals: 1, "no"

firstWordID = selected("Sound", 1)
lastWordID = selected("Sound", 10)

writeInfoLine: "The mean intensities in all extracted words:"

for i from firstWordID to lastWordID
        selectObject: i
        soundName$ = selected$ ()
        meanIntensity$ = Get intensity (dB)
        appendInfoLine: soundName$ + "   " + meanIntensity$
endfor


exitScript ()
```

https://github.com/yukat237/Praat_Workshop/blob/main/List%20Intensities%20(Let's%20try%20running%20a%20loop%20slide)

Let's try running a loop

Readme

Extract sounds based on tier 1 (by Word segmentation)

Get unique IDs of 1st and 10th words

Prep an Info window to write results on

FOR LOOP

```
#---CHECK--------------
# Make sure you have Sound VOT_English-Spanish & TextGrid VOT_English-Spanish
# in your objects window.
#----------------------

selectObject: "Sound VOT_English-Spanish"
plusObject: "TextGrid VOT_English-Spanish"
Extract non-empty intervals: 1, "no"

firstWordID = selected("Sound", 1)
lastWordID = selected("Sound", 10)

writeInfoLine: "The mean intensities in all extracted words:"

for i from firstWordID to lastWordID
        selectObject: i  Select the extracted sound #i
        soundName$ = selected$ ()  set soundName variable (getting extracted sound name)
        meanIntensity$ = Get intensity (dB) Set meanIntensity variable as string (using Function)
        appendInfoLine: soundName$ + "   " + meanIntensity$    Adding line on the info window
endfor


exitScript ()
```

Other loops

Run the code repeatedly, until the condition is met.

Repeat - until

```
throws = 0
repeat
    eyes = randomInteger (1, 6) + randomInteger (1, 6)
    throws = throws + 1
until eyes = 12
writeInfoLine: "It took me ", throws, " trials to throw 12 with two dice."
```

while - endwhile

```
while x < 0
    x = x + 2 * pi
endwhile
while x >= 2 * pi
    x = x - 2 * pi
endwhile
```

less common than For Loops, but useful.

Jumping depending on conditions "if statement"

Basic Idea

1 if/else section

```
if duration <= 1
        CODE TO LABEL AS 'FAILED DATA'
elsif duration >=20
        CODE TO LABEL AS 'FAILED SEGMENTATION'
else
        CODE TO PROCESS DATA
endif
```

Condition to run the following code

Another condition to run the following code

Otherwise, run the following code

This code means:
1) if duration is less than 1, label that data as 'failed data.'
2) if duration is more than 20, label that data as 'failed segmentation'
3) Otherwise, process data as a good quality data

Especially useful when you want to force the code to skip some data with errors!

Various Commands

```
Get mean: 0, 0, "Hertz"
```

Command
to get a mean pitch from "Pitch" object

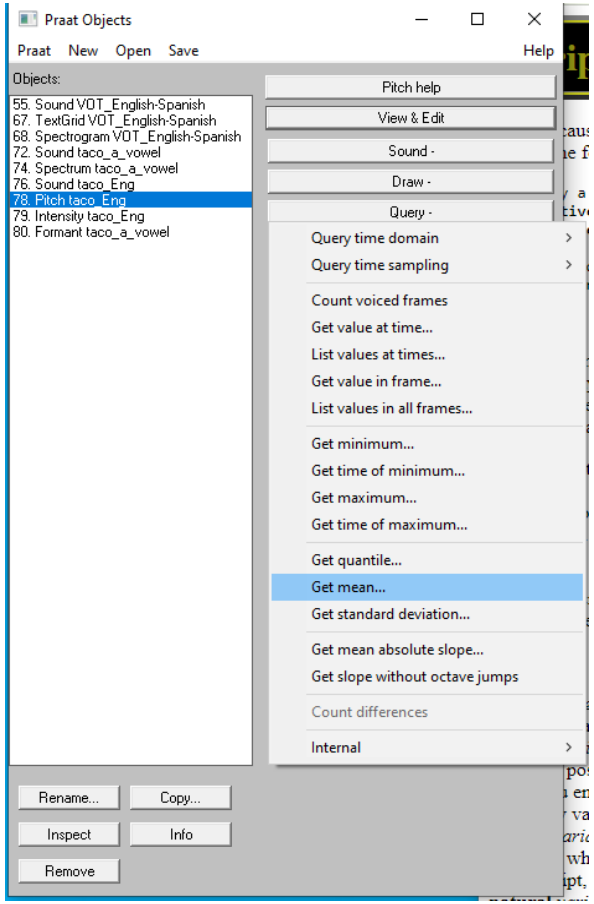
Arguments

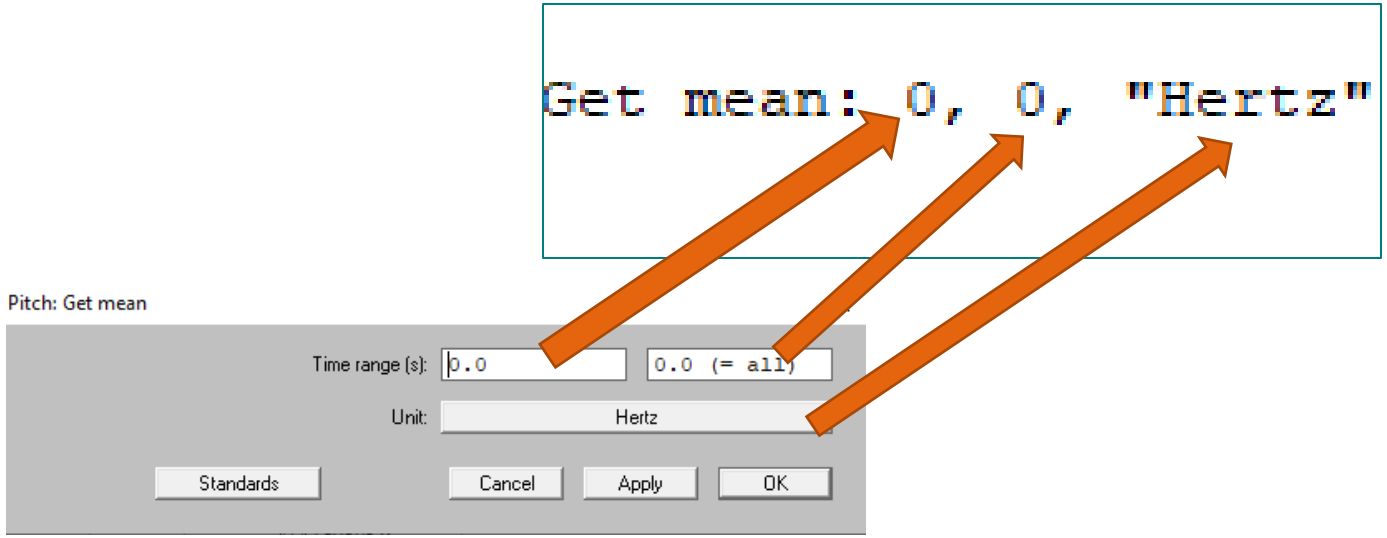
- Refresher -
Object types

- Sound _____
- TextGrid _____
- Spectrogram _____
- Spectrum _____
- Pitch _____
- Formant_____
- Intensity_____

Question is...
How do I know what these arguments mean?

Method1: if you know how to do it in GUI, go to GUI and do it. These arguments are corresponding to the form (pop-up window) contents.





Method2 – Google and Visit [official Praat manual]

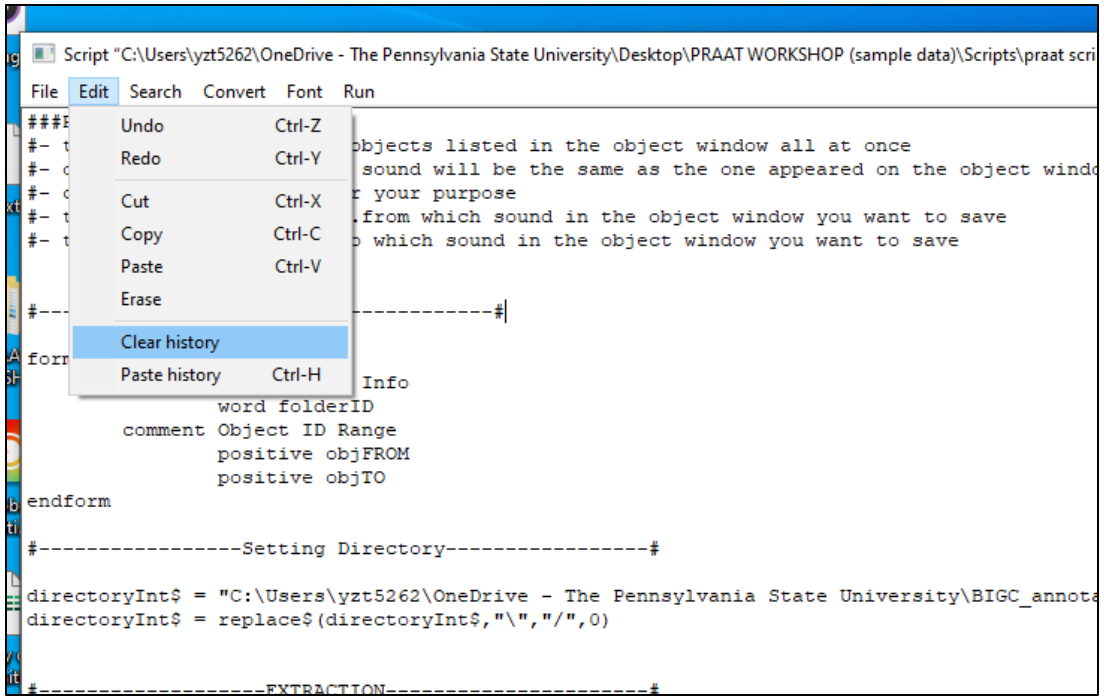
Method3 – Look at others' scripts and try reading it

How to know which **command** to use, first of all?

Method1 – if you know how to do it in GUI, use **Paste History** function

Easy 3 steps:

1. On your script window, **Edit** > **Clear History**
2. Go to GUI and do what you want your script to do
3. Come back to the script, **Edit** > **Paste History**



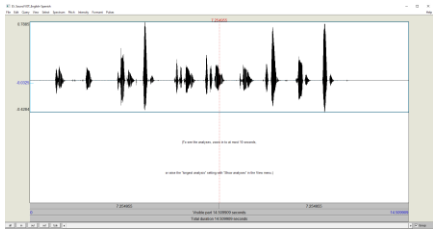
Method2 – Google and Visit [official Praat manual]

Method3 – Look at others' scripts and try reading it

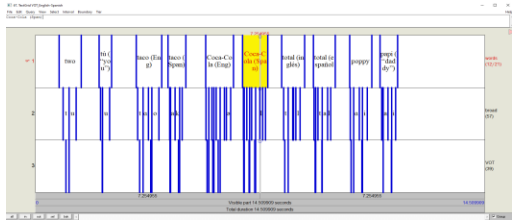
- Refresher -
Object types

1. Sound _____
2. TextGrid _____
3. Spectrogram _____
4. Spectrum _____
5. Pitch _____
6. Formant_____
7. Intensity_____

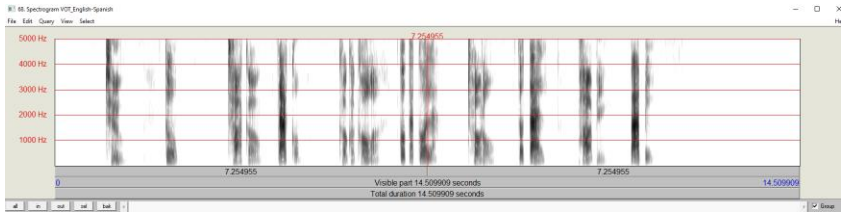
[1]



[2]

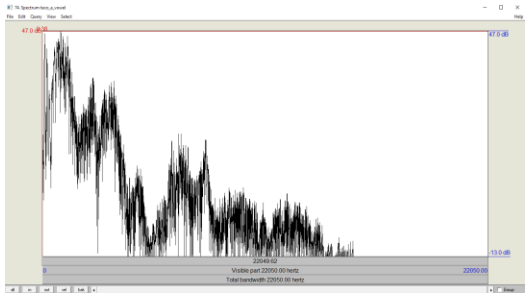


[3] Select Sound > Analsyze spectrum > Sound: To Spectrogram...

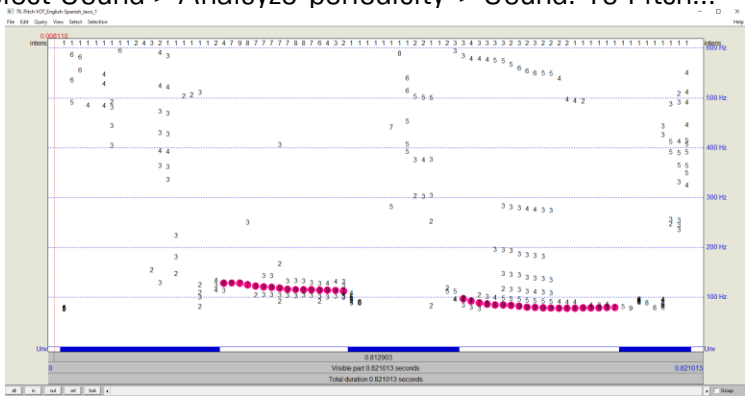


*Draw this onto Picture Window; select this object > Spectrogram: Paint...

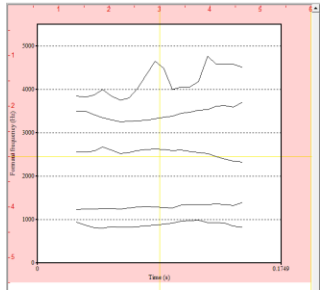
[4] Select Sound > Analsyze spectrum > Sound: To Spectrum...



[5] Select Sound > Analsyze periodicity > Sound: To Pitch...

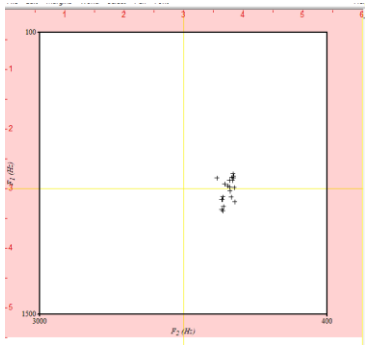


[6] Select Sound > Analyze spectrum... > Sound: To Formant (burg)...

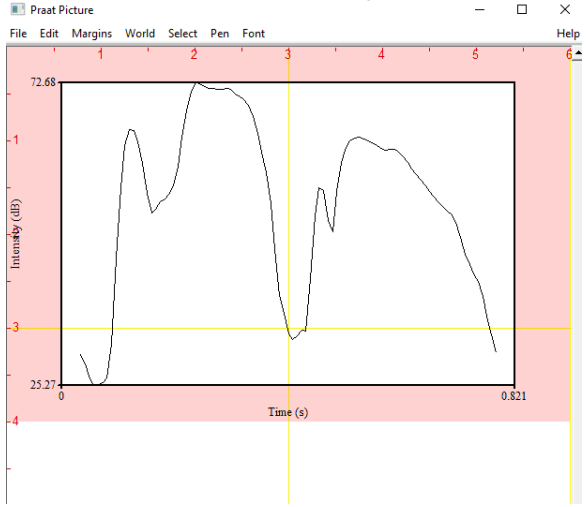


Draw > Draw Tracks

Draw > Scatter Plot



[7] Select Sound > To Intensity...



- Refresher -
Object types

https://www.fon.hum.uva.nl/praat/manual/Types_of_objects.html

...and MORE!!!

Praat contains the following types of objects and Editors. For an introduction and tutorials, see Intro.

General purpose:

- Matrix: a sampled real-valued function of two variables
- Polygon
- PointProcess: a point process (PointEditor)
- Sound: a sampled continuous process (SoundEditor, SoundRecorder, Sound files)
- LongSound: a file-based version of a sound (LongSoundEditor)
- Strings
- Distributions, PairDistribution
- Table, TableOfReal
- Permutation
- ParamCurve

Periodicity analysis:

- Tutorials:
- Intro 4. Pitch analysis
- Intro 6. Intensity analysis
- Voice (jitter, shimmer, noise)
- Pitch: articulatory fundamental frequency, acoustic periodicity, or perceptual pitch (PitchEditor)
- Harmonicity: degree of periodicity
- Intensity, IntensityTier: intensity contour
- Electroglottogram

Spectral analysis:

- Tutorials:
- Intro 3. Spectral analysis
- Intro 5. Formant analysis
- Spectrum: complex-valued equally spaced frequency spectrum (SpectrumEditor)
- Ltas: long-term average spectrum
- Spectro-temporal: Spectrogram, BarkSpectrogram, MelSpectrogram
- Formant: acoustic formant contours
- LPC: coefficients of Linear Predictive Coding, as a function of time
- Cepstrum, CC, LFCC, MFCC (cepstral coefficients)
- Excitation: excitation pattern of basilar membrane
- Excitations: an ensemble of **Excitation** objects
- Cochleagram: excitation pattern as a function of time

Labelling and segmentation (see Intro 7. Annotation):

- TextGrid (TextGridEditor)

Listening experiments:

- ExperimentMFC

Manipulation of sound:

- Tutorials:
- Intro 8.1. Manipulation of pitch
- Intro 8.2. Manipulation of duration
- Intro 8.3. Manipulation of intensity
- Filtering
- Source-filter synthesis
- PitchTier (PitchTierEditor)
- Manipulation (ManipulationEditor): overlap-add
- DurationTier
- FormantGrid

Articulatory synthesis (see the Articulatory synthesis tutorial):

- Speaker: speaker characteristics of a woman, a man, or a child
- **Articulation**: snapshot of articulatory specifications (muscle activities)
- Artword: articulatory target specifications as functions of time
- (VocalTract: area function)

Neural net package:

- FFNet: feed-forward neural net
- PatternList
- Categories: for classification (**CategoriesEditor**)

Numerical and statistical analysis:

- Eigen: eigenvectors and eigenvalues
- Polynomial, Roots, ChebyshevSeries, LegendreSeries, ISpline, MSpline
- Covariance: covariance matrix
- Confusion: confusion matrix
- Discriminant analysis: Discriminant
- Principal component analysis: PCA
- Correlation, ClassificationTable, SSCP
- DTW: dynamic time warping

Multidimensional scaling:

- Configuration (Salience)
- Kruskal analysis: Dissimilarity (Weight), Similarity
- INDSCAL analysis: Distance, ScalarProduct
- Correspondence analysis: ContingencyTable

Optimality-theoretic learning (see the OT learning tutorial)

- OTGrammar (OTGrammarEditor)

Bureaucracy

- WordList, SpellingChecker

Regular Expression

Regex list:
https://www.fon.hum.uva.nl/praat/manual/Regular_expressions_1__Special_characters.html

Part 3

Hands-on practice -- let's write a full script!

Part 3

Hands-on practice -- let's write a full script!

The process of writing the script explained here is reflecting my real flow of thoughts-- it is not the top-to-bottom writing, which might not be your preference, but I thought presenting an authentic writing process would be helpful to who has no background in coding

Plan the structure

Think about:

1) what kind of **data** you have (input)

2) what **products** you want to make (output)

Then, write the overall structure with #(comments function) on a new script

Practice Input & Output

Example:

1. I have a lot of sentence stimuli and I want to normalize its intensity
--> input = , output =

2. I have spontaneous speech data and want to automatically segment all the speech by words and phonemes
--> input = , output =

3. I have word production data and want to get acoustic measures of various kinds
--> input = , output =

4. I have a set of TextGrids with manually segmented by words and excel sheets of their responses, and I want to label them with the words by referring to the excel sheets.
--> input = ; Output =

5. I want to extract sounds from the data based on the segments on the Textgrid and save all the new sounds
--> input = , output =

Practice Input & Output

Example:
1. I have a lot of sentence stimuli and I want to normalize its intensity
--> input = sounds, output = sounds

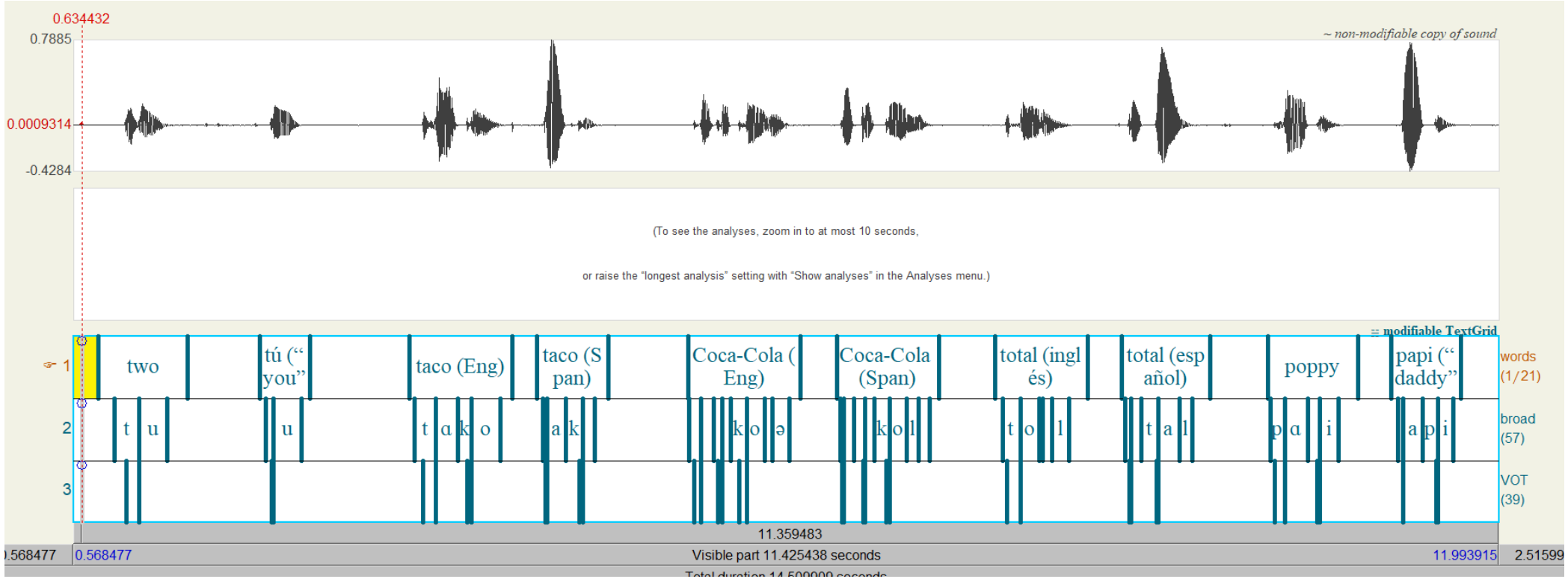
2. I have spontaneous speech data and want to automatically segment all the speech by words and phonemes
--> input = sounds, output = TextGrid

3. I have word production data and want to get acoustic measures of various kinds
--> input = sound, output = a table or text file

4. I have a set of TextGrids with manually segmented by words and excel sheets of their responses, and I want to label them with the words by referring to the excel sheets.
--> input = text file made out of excel sheets, sounds, and Textgrids; Output = updated textgrids

5. I want to extract sounds from the data based on the segments on the Textgrid and save all the new sounds
--> input = sound and textgrid, output = sound files (.wav)

A script we try together today



You have this sound file annotated by words, (broadly transcribed) phonemes, and VOT.
You want to extract each sound, normalize the intensity, and save them as separate sound files.
Also, you want to get data of VOT duration.

--reminder--
3 things you can do with Praat
1. Analyze sound
2. Manipulate sound
3. Data Process of sound

Step1: think about inputs and outputs

You have this sound file annotated by words, (broadly transcribed) phonemes, and VOT.
You want to extract each sound, normalize the intensity, and save them as separate sound files.
Also, you want to get data of VOT duration

Input

You have:
- 1 sound file
- 1 TextGrid (with 3 tiers)

Output

You want:
- 10 .wav files for each word (with Intensity normalized) in your local folder
- 1 table with VOT duration data on it

2. Structure - think of what you would do with GUI

Input

You have:
- 1 sound file
- 1 TextGrid (with 3 tiers)

Output

You want:
- 10 .wav files for each word (with Intensity normalized) in your local folder
- 1 table with VOT duration data

Structure

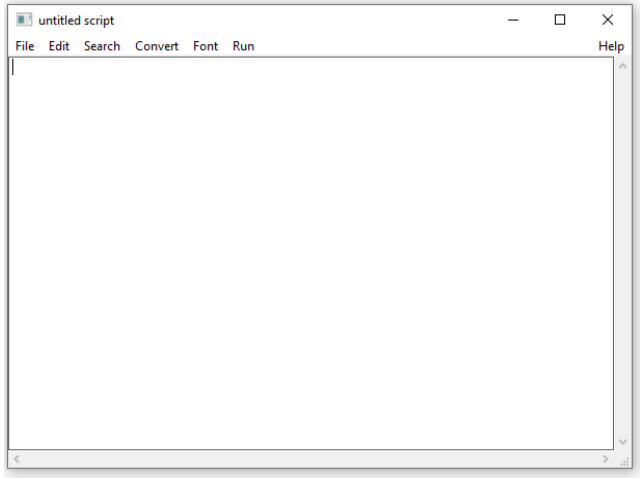
- Starting point: Original sound file and TextGrid is in the object window
- Body:
  - Extract the sounds word by word based on the word tier
  - For each extracted sound...
    - normalize intensity
    - save as .wav file
  - For each TextGrid interval...
    - Get VOT duration
    - Get word & phoneme information for this VOT
    - Record these in a table

Goal is something like:

word	phoneme	VOT
Two (En)	t	0.1
Tu (Sp)	t	0.02
...	...	...

Step3: write the structure down in a new script

1. Open a new script
(Praat > New Praat Script)



(the full script is HERE)

2. Write down your plan with "#"

```
#=====Project Title=====================================#

#  Extract word sound files, get VOT and Formants
#             LAST EDIT:2022/12/8

#=========================================================#

#-----------------ReadMe----------------------#

#---------------------------------------------#


#####   FORM   #####-------------------------------------------------------


#####   TABLE   #####------------------------------------------------------


#### Extract sounds for each word, normalize dB, and save####-----------------------------------

#extracting as Sound objects

#Loop through extracted sounds

        #Normalize intensity

        #Saving as .wav files

#### LOOP through VOT intervals to get VOT durations ####-------------------------------------

        #get VOT

        #get the word for this VOT

        #get the phoneme for this VOT

        #save the VOT info on the table

exitScript: "success!"
```

```
#=====Project Title=====================================#

#  Extract word sound files, get VOT and Formants
#          LAST EDIT :  2022/12/8

#=========================================================#

#----------------ReadMe-------------------#

#-------------------------------------#


#####   FORM   #####-----------------------------------------------


#####   TABLE   #####----------------------------------------------


#### Extract sounds for each word, normalize dB, and save####-----------------------------------------

#extracting as Sound objects

#Loop through extracted sounds

   #Normalize intensity

   #Saving as .wav files

#### LOOP through VOT intervals to get VOT durations ####-------------------------------------------------

   #get VOT

   #get the word for this VOT

   #get the phoneme for this VOT

   #save the VOT info on the table

exitScript: "success!"
```

Project Title, Dates, Your Name

Read Me (Notes for others & future self)

Form

Prepping Table

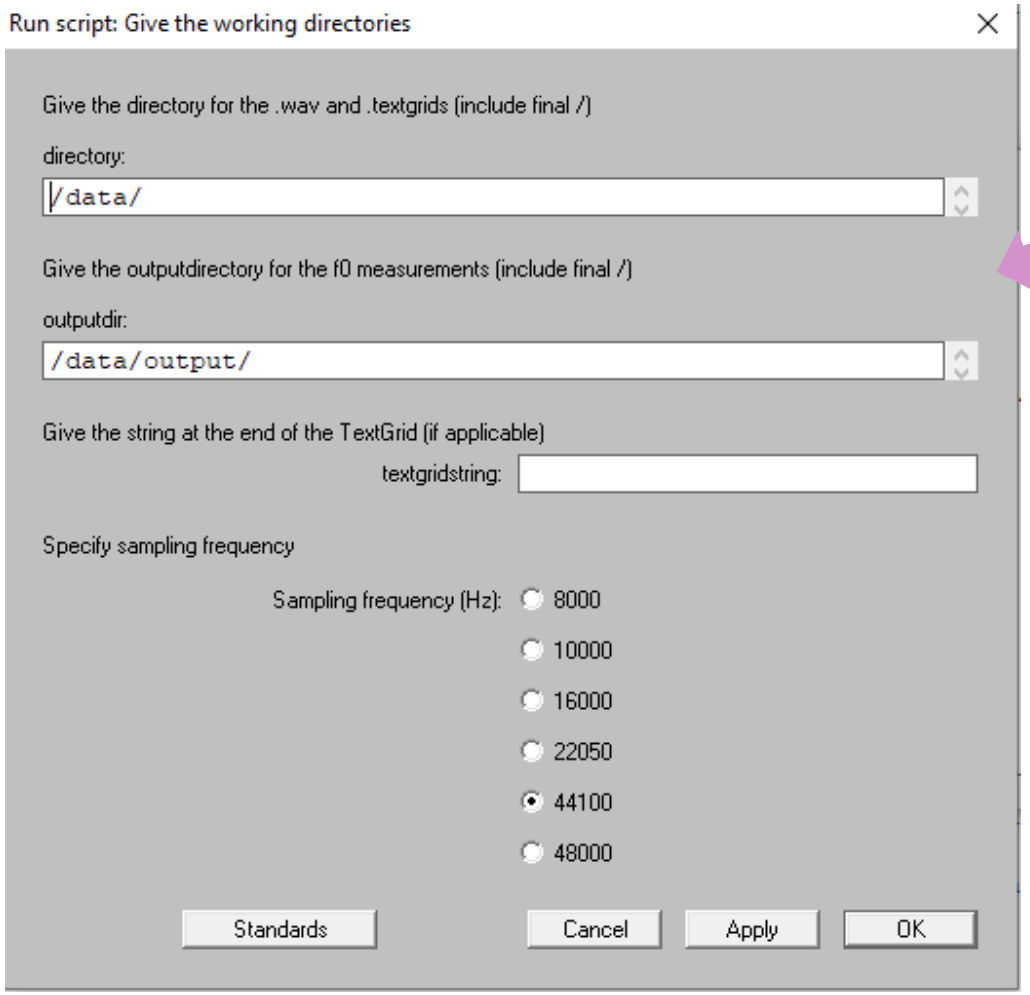
Extract sounds, Loop through them to normalize dB, Save.

Loop through Tier3 intervals (VOT tier) to get VOT duration data

BODY

Step4: Write a form

Form in Praat is something looks like this:



untitled script (modified)

File Edit Search Convert Font Run

```
form Give the working directories
        comment Give the directory for the .wav and .textgrids (include final /)
        text directory /data/

        comment Give the outputdirectory for the f0 measurements (include final /)
        text outputdir /data/output/

        comment Give the string at the end of the TextGrid (if applicable)
        positive textgridstring

        comment Specify sampling frequency
        choice Sampling_frequency_(Hz) 5
                button 8000
                button 10000
                button 16000
                button 22050
                button 44100
                button 48000
endform
```

You get the form on the left from this kind of coding

Form is helpful to generalize the script's use, and manually change how you want code to behave without going back to the code.

Step4: Write a form

For our purpose, let's write this:

Surrounding by "form" -- "endform" makes a form.
Hit Run to see how this looks like!

```
form    Title of this form
        sentence   sound_object_name    VOT_EnglishSpanish
        positive   target_intensity   70
        comment   Where the sound should be saved?
        text   folderDir   Folder path to save your sounds
endform
```

*** Gray & Italicized words...you decide what to type!*

Step4: Write a form

For our purpose, let's write this:

Surrounding by "form" -- "endform" makes a form.
Hit Run to see how this looks like!

```
form   Title of this form
        sentence   sound_object_name   VOT_EnglishSpanish
        positive   target_intensity   70
        comment   Where the sound should be saved?
        text   folderDir   Folder path to save your sounds
endform
```

variable

Default in textbox

variable

variable

Default in textbox

*** Gray & Italicized words...you decide what to type!*

FYI: field types for the form

- numeric
  - **real** `variable initialValue` real numbers
  - **positive** `variable initialValue` positive real numbers
  - **integer** `variable initialValue` whole numbers
  - **natural** `variable initialValue` positive whole numbers

- string
  - **word** `variable initialValue` a string without spaces
  - **sentence** `variable initialValue` any short string
  - **text** `variable initialValue` any possibly long string

- selection
  - **boolean** `variable initialValue` a check box
  - **choice** `variable initialValue` radio buttons

- **comment** `text` a line with any text. Does not become a variable!

https://www.linguisticsociety.org/sites/default/files/e-learning/Day7.pdf

Step5: Write a code to extract each sound

If it's GUI, you would:
1) select sound object and the TextGrid
2) click on "Extract" > "Extract the non-empty intervals"
3) choose the tier you want to refer to.
4) all sounds listed as Sound objects in the object window.

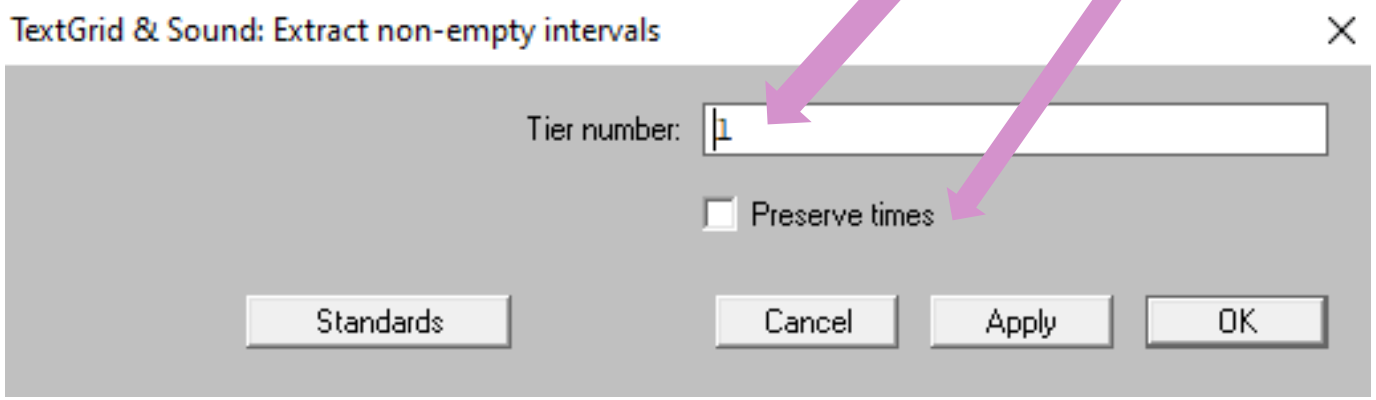
```
#extracting as Sound objects
selectObject: "Sound " + sound_object_name$
plusObject: "TextGrid " + sound_object_name$
Extract non-empty intervals: 1, "no"
```

Tier Number variable

Step5: Write a code to extract each sound

```
#extracting as Sound objects
selectObject: "Sound " + sound_object_name$
plusObject: "TextGrid " + sound_object_name$
Extract non-empty intervals: 1, "no"
```

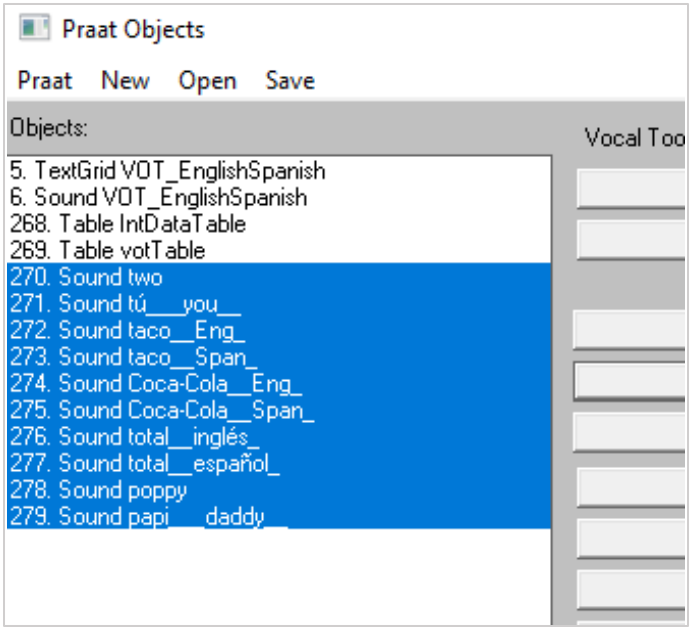
Corresponding with what you will see in GUI!



Step6: Loop through extracted sounds to normalize dB

If you run the 3 lines, you will get extracted sounds as Sound object, all selected.

When you loop through these sounds, you want to know the ID of these sounds. But that depends on how many objects you have already made. So, let's write a script to get the ID of these.




```
Extract non-empty intervals: 1, "no"

firstSdID = selected(1)
lastSdID = selected (-1)
totalNumofWords = lastSdID - firstSdID + 1
```

Step6: Loop through extracted sounds to normalize dB

```
for i from firstSdID to lastSdID
    #Normalize intensity
    selectObject: i
    nameOfSdObj$ = selected$ ("Sound")
    oldInt = Get intensity (dB)
    Scale intensity... 'target_intensity'

    #Saving as .wav files
    Save as WAV file: folderDir$ + "\" + nameOfSdObj$ + ".wav"
    diffOfInt =  70 - oldInt

endfor
```

Step6: Loop through extracted sounds to normalize dB

```
for i from firstSdID to lastSdID
    #Normalize intensity
    selectObject: i
    nameOfSdObj$ = selected$ ("Sound")
    oldInt = Get intensity (dB)
    Scale intensity... 'target_intensity'

    #Saving as .wav files
    Save as WAV file: folderDir$ + "\" + nameOfSdObj$ + ".wav"
    diffOfInt =  70 - oldInt

endfor
```

Select sound object

Get the name of sound object (for later use)

get original intensity and save it into the variable "oldInt"

Normalizing Intensity to 70dB (as entered in the form)

Save sound as .wav in the folder you entered in the form

Get the Intensity difference with original dB and save it into the variable "diffOfInt"

Step6: Loop through extracted sounds to normalize dB

```
for i from firstSdID to lastSdID
    #Normalize intensity
    selectObject: i
    nameOfSdObj$ = selected$ ("Sound")
    oldInt = Get intensity (dB)
    Scale intensity... 'target_intensity'

    #Saving as .wav files
    Save as WAV file: folderDir$ + "\" + nameOfSdObj$ + ".wav"
    diffOfInt =  70 - oldInt

    selectObject: tableID1
    Append row
    tblrow = Get number of rows
        Set string value: tblrow, "word", nameOfSdObj$
        Set numeric value: tblrow, "OldIntensity", oldInt
        Set numeric value: tblrow, "Difference", diffOfInt

endfor
```

Optionally, let's make a table to save the **word**, **original intensity**, & **the difference of the intensity**

We have not made the table object yet, so let's make one right below the form! (next slide)

*Gray...parts you already typed with the previous slide

Step6: Loop through extracted sounds to normalize dB

```
for i from firstSdID to lastSdID
    #Normalize intensity
    selectObject: i
    nameOfSdObj$ = selected$ ("Sound")
    oldInt = Get intensity (dB)
    Scale intensity... 'target_intensity'

    #Saving as .wav files
    Save as WAV file: folderDir$ + "\" + nameOfSdObj$ + ".wav"
    diffOfInt =  70 - oldInt

    selectObject: tableID1
    Append row
    tblrow = Get number of rows
        Set string value: tblrow, "word", nameOfSdObj$
        Set numeric value: tblrow, "OldIntensity", oldInt
        Set numeric value: tblrow, "Difference", diffOfInt

endfor
```

Optionally, let's make a table to save the **word**, **original intensity**, & **the difference of the intensity**

We have not made the table object yet, so let's make one right below the form! (next slide)

"I am adding a row!"

How many row exists?

On "word" column, add this nameOfSdObj$ string

On "OldIntensity" column, add this oldInt number value

On "Difference" column, add this diffOfInt number value

*Gray...parts you already did typed with the previous slide

Step6: Loop through extracted sounds to normalize dB

Let's make the table below the Form section, to keep Intensity information!

Also, we know we want one table for VOT, so let's make that too.

```
    text folderDir C:\Users\yz65262\OneDrive - The Pennsylvania State University\PRAAT

endform

#####   TABLE   #####-------------------------------------------------------

tableID1 = Create Table with column names: "IntDataTable", 0, "word OldIntensity Difference
tableID2 = Create Table with column names: "votTable", 0, "word phoneme VOT"

#### Extract sounds for each word, normalize dB, and save####-----------------------------

#extracting as Sound objects
selectObject: "Sound " + sound_object_name$
plusObject: "TextGrid " + sound_object_name$
```

Step6: Loop through extracted sounds to normalize dB

Let's make the table below the Form section, to keep Intensity information!

Also, we know we want one table for VOT, so let's make that too.

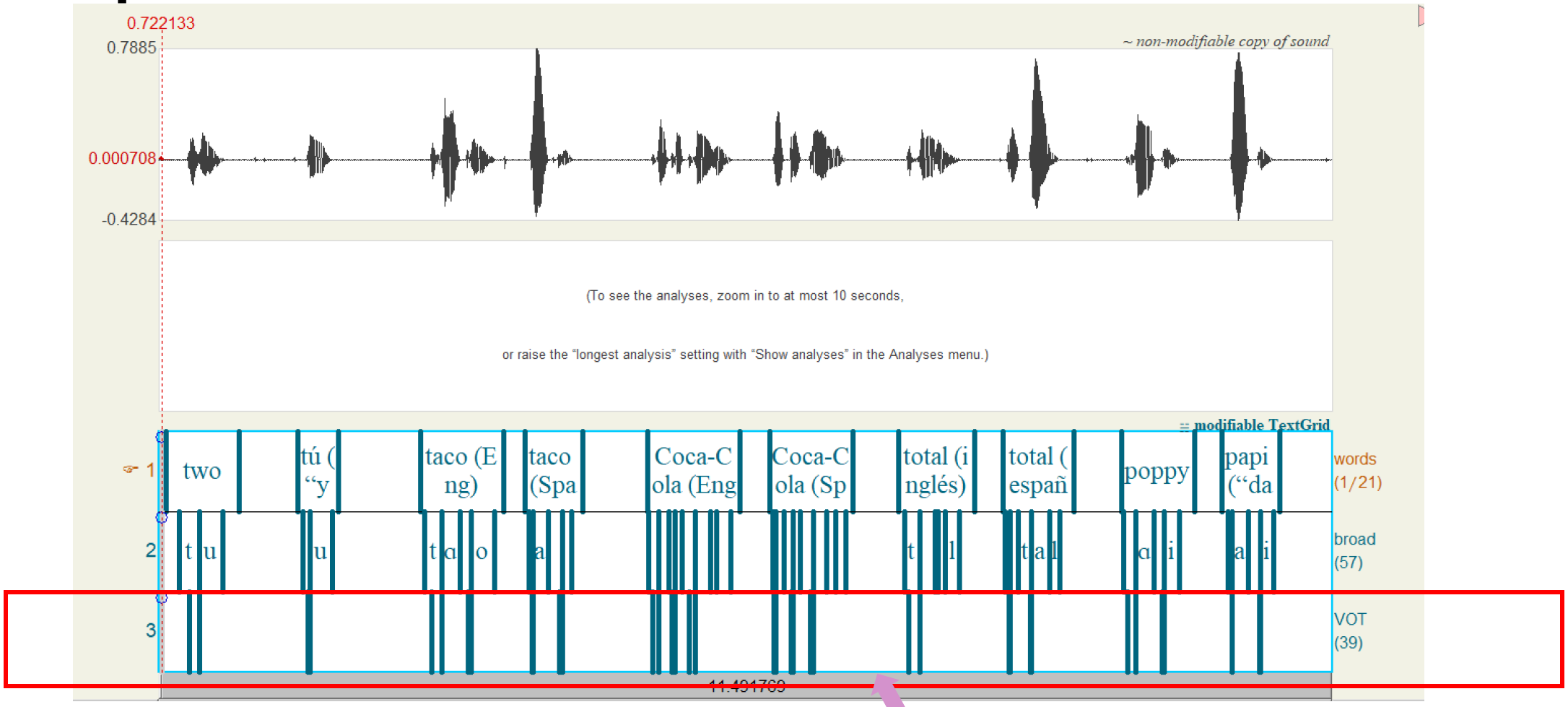
```
text folderDir C:\Users\yz65262\OneDrive - The Pennsylvania State University\PRAAT

endform

#####   TABLE   #####-------------------------------------------------------
```

Naming Table Object # of row List of column names (space = end of name)

```
tableID1 = Create Table with column names: "IntDataTable", 0, "word OldIntensity Difference
tableID2 = Create Table with column names: "votTable", 0, "word phoneme VOT"

#### Extract sounds for each word, normalize dB, and save####-----------------------------

#extracting as Sound objects
selectObject: "Sound " + sound_object_name$
plusObject: "TextGrid " + sound_object_name$
```

We are halfway through :)

Step7: Get VOT and save the info in the table



The plan: loop should check each interval on Tier3, from left to right, and get duration of the interval only if the interval has the label "VOT" in it.

Step7: Get VOT and save the info in the table

1. make a for loop.

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3
```

```
for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

1. make a for loop.

2. Make "nVOT" variable. (we want this for-loop to continue for the total # of tier3 intervals. So, get that by Get number of intervals)

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

1. make a for loop.

2. Make "nVOT" variable. (we want this for-loop to continue for the total # of tier3 intervals. So, get that by Get number of intervals)

3. select TextGrid

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

4. get the VOT label (1st argument: tier 3 2nd argument: interval number counting from left) And save it in the variable "labelT3$"

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

4. get the VOT label (1st argument: tier 3 2nd argument: interval number counting from left) And save it in the variable "labelT3$"

5. write if statement. (we do not need to get duration if it's empty, so checking if the label is "VOT"

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$

    labelT3$ = Get label of interval: 3, i

    if labelT3$ = "VOT"
    startTime = Get start time of interval: 3, i
    endTime = Get end time of interval: 3, i
    votDur = endTime - startTime

    #save the VOT info

    endif
endfor
```

6. get duration and assign the value to votDur variable (getting duration often involves 3 steps: get Start time, get End time, and do "End - Start")

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT

    selectObject: "TextGrid " + sound_object_name$

  labelT3$ = Get label of interval: 3, i

  if labelT3$ = "VOT"
  startTime = Get start time of interval: 3, i
  endTime = Get end time of interval: 3, i
  votDur = endTime - startTime


  #save the VOT info
  selectObject: tableID2
  Append row
  tblrow = Get number of rows
        Set string value: tblrow, "word", wordlabel$
        Set string value: tblrow, "phoneme", phonlabel$
        Set numeric value: tblrow, "VOT", votDur
  endif
endfor
```

7. Now we want to write the votDur value into the table! (same as the appending to table1, except the tableID2 and column names and the corresponding variables

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT

    selectObject: "TextGrid " + sound_object_name$

  labelT3$ = Get label of interval: 3, i

  if labelT3$ = "VOT"
  startTime = Get start time of interval: 3, i
  endTime = Get end time of interval: 3, i
  votDur = endTime - startTime

  #save the VOT info
  selectObject: tableID2
  Append row
  tblrow = Get number of rows
        Set string value: tblrow, "word", wordlabel$
        Set string value: tblrow, "phoneme", phonlabel$
        Set numeric value: tblrow, "VOT", votDur
  endif
endfor
```

8. we need these 2 variables. That is, we want to know what was the word and the phoneme for the VOT we just measured (next slide!)

Step7: Get VOT and save the info in the table

```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$
  labelT3$ = Get label of interval: 3, i
  if labelT3$ = "VOT"
  startTime = Get start time of interval: 3, i
  endTime = Get end time of interval: 3, i
  votDur = endTime - startTime
      #get the word for this VOT
      intervalNumOfWord = Get interval at time: 1, startTime
      wordlabel$ = Get label of interval: 1, intervalNumOfWord

      #get the phoneme for this VOT
      intervalNumOfPhon =Get interval at time: 2, startTime
      phonlabel$ = Get label of interval: 2, intervalNumOfPhon
  #save the VOT info
  selectObject: tableID2
  Append row
  tblrow = Get number of rows
      Set string value: tblrow, "word", wordlabel$
      Set string value: tblrow, "phoneme", phonlabel$
      Set numeric value: tblrow, "VOT", votDur
  endif
endfor
```

8. make wordlabel$ and phonlabel$.
Each is looking at: "what is the interval number of Tier 1or2, at the start time of this VOT interval?" & "What is the label of that interval?

Step7: Get VOT and save the info in the table

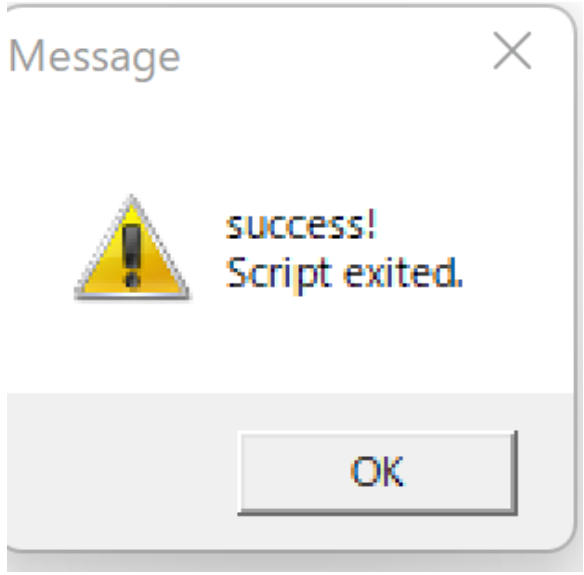
```
selectObject: "TextGrid " + sound_object_name$
nVOT = Get number of intervals: 3

for i to nVOT
    selectObject: "TextGrid " + sound_object_name$
  labelT3$ = Get label of interval: 3, i
  if labelT3$ = "VOT"
  startTime = Get start time of interval: 3, i
  endTime = Get end time of interval: 3, i
  votDur = endTime - startTime
      #get the word for this VOT
      intervalNumOfWord = Get interval at time: 1, startTime
      wordlabel$ = Get label of interval: 1, intervalNumOfWord

      #get the phoneme for this VOT
      intervalNumOfPhon =Get interval at time: 2, startTime
      phonlabel$ = Get label of interval: 2, intervalNumOfPhon
  #save the VOT info
  selectObject: tableID2
  Append row
  tblrow = Get number of rows
      Set string value: tblrow, "word", wordlabel$
      Set string value: tblrow, "phoneme", phonlabel$
      Set numeric value: tblrow, "VOT", votDur
  endif
endfor
```

Done!!! :)

Option 1: exitScript

The clear sign of successful processing. Instead of the "success" , you can also write variables here to see their values.

```
                    intervalNumOfPhon
                    phonlabel$ = Get l

                    #save the VOT info
                    selectObject: tabl
                    Append row
                    tblrow = Get numbe
                            Set string
                            Set string
                            Set numeri

          endif
endfor


exitScript: "success!"
```



Option 2: write something in Read Me

For future self or any readers of your script, write down
- things users of the script have to do before running scripts
- what kind of data format you need to use it
- any notes about the form

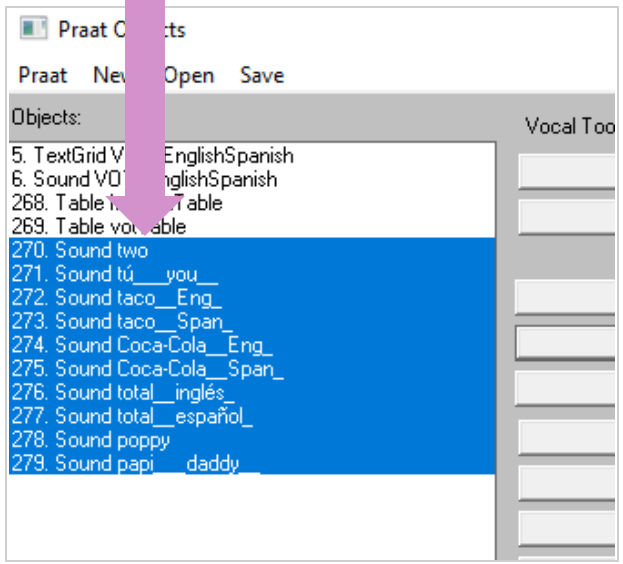
```
#==================================================

#-----------------ReadMe---------------------#

# Make sure you have below in your object window
#       1)Sound Object
#       2)TextGrid, with word/phoneme/VOT annotated

#-----------------------------------------#


#####   FORM   #####-------------------------------
```

In our case, the user has to have Sound obj and Tg obj in their object window. And TextGrid has to have 3 Tiers. So I am writing these down.

Option 3: Cleaning

Without cleaning, too many objects created through the porcess may be left in the object window.

In our case, for example,
these selected objects are not needed after the loop of saving them as wav files.



So, consider adding these 2 lines before closing for loop for extracted sounds:

```
        selectObject: tableID1
        Append row
        tblrow = Get number of ro
                Set string value:
                Set numeric value
                Set numeric value

        #Cleaning
        selectObject: i
        Remove
endfor
```

NOTE

Similarly to any other programming languages,

There are many ways to write scripts for 1 same goal.
Although there are "more efficient" scripts than others,
No script is "wrong" as long as it achieves the goal.

Part 4

Other Implementations you can use with scripting knowledge

Other kind of scripts available everywhere

My script:

[Getting an acoustic measures table from sound files]

Please see the reference page at the end of this powerpoint for more resources

Add silence to the beginning of all sound files in a folder
This script adds a specified amount of silence to the beginning
saved with their original names to a folder specified by the user

Add silence to the end of all sound files in a folder
This script adds a specified amount of silence to the end of eve
with their original names to a folder specified by the user. We
presenting in Qualtrics, since this software sometimes clips the

Get F1 to F4 at 7 times points for all labeled intervals
This script extracts measurements for F1, F2, F3, and F4 at 7 e
87.5%, and 100%) in all labeled intervals. It processes all labe
text file. Each sound file and its corresponding TextGrid should

Get duration and timepoints for all labeled intervals or poi
This script logs the starting point, end point, and duration of la
timepoint of labeled points in a specified point tier for all files in
point tier. Each sound file and its corresponding TextGrid shoul

Pull out words found in text file from one tier and put on a
This script takes a list of words in a text file as input. This text
irrelevant), and all the words in a single column. For all files in
for those words, and copies them into a new interval tier at the
appear in the tier above. This is useful if you want to pull out s
run analyses on in a separate tier.

- Annotation checker This praat will help you to check all the annotation files in the specified folder.
- Extract intervals This script extracts all the sound intervals with an interval name on the annotation tier.
- Word count This script counts the number of labels (e.g. frequencies of particular words) in all annotation files in a folder. Originally written to analyze Corpus of Spoken Japanese, but can be used for any other corpus annotated by Praat.
- Equalizing amplitude (dB) This praat script adjusts the average amplitude (in dB) of all files in a folder.
- Scale peak This praat script scales peak of all files in a folder.
- Equalize duration This praat script adjusts the duration of all files to a specified value.
- Combine all sounds This praat script combines (not concatenates) all sound files in a directory. Use it to create multi-speaker noise.
- Change F0 This praat script raises/lowers the whole pitch contour by the specified factor for all the files.
- Adjust to nearest zero crossing This praat script adjusts the beginning and the end of all files to nearest zero-crossings.
- Mono converter This praat script converts all stereo sounds into mono sounds.
- get duration This praat script takes all the textgrid files in a folder and gets duration of all labelled intervals. (This is based on the script that Mietta Lennes originally wrote.)
- get F0 min max This praat script takes all the files in a folder, and for all intervals, it takes the F0 maximum, and F0 minimum preceding the maximum and following the maximum. (This is based on the script that Matsuura Toshio originally wrote.)
- get intensity min max This praat script takes all the files in a folder, and for all intervals, it takes the average intensity, minimal intensity, its time, maximal intensity and its time.
- get F1, F2, F3 (averages) This praat script takes all the files in a folder, and for all intervals, it calculates the average F1, F2 and F3.
- get F1, F2, F3 (midpoints) This praat script takes all the files in a folder, and for all intervals, it calculates the F1, F2 and F3 at their midpoints.
- get F0, F1 and duration This script is intended to help an acoustic analysis of a voicing contrast. Specifically, for each interval (for all the files in the folder), it calculates F0 and F1 at both edges and its duration.
- suffixation This script combines one suffix sound file (say your context or burst) at the end of all other files in the folder (say your continuum or closure).
- Remove noise This script removes noise. Please read Praat's help for specific details.

At each "sp", label separate sentence interval ***(e.g., from FAVE align output)***

Get sentence from labeled word intervals ***(assumes you have (unlabeled) sentences in s***
intervals)

Measurements

Get intensity, duration, & mean f0 over 12 points of the vowel ***(Adapted from Christian D***

Get duration measurements for each annotated word

Get duration of vowel and coda for each word

Combining sounds

Concatenate 2 sounds (1 sound == frame)

Concatenate all sounds in a directory

Resample all files & combine with other sound

Extract & save all sounds

... from a force aligned file (finding "sp"s) and numbering sentences

Split individual sounds from a word and save to directory

Writing to .txt file

Sound file management

- **get-files** (**Kevin Ryan**)
  Open multiple files from the specified directory at once.
- **get-files-from-list** (**Bert Remijsen**, back up **here**)
  Open multiple files enumerated in a list in the specified text file (BR's **description**).
- **remove-all** (**Kevin Ryan**)
  Remove all objects from object list.
- **change-sample-rate-or-format** (**Mietta Lennes**, back up **here**)
  Resample and/or change the format of a set of sound files (ML's **description**).
- **concatenate-sounds** (**Chad Vicenik**)
  Concatenate (daisy-chain) two or more selected Sound objects into one Sound object.
- **duplicate-sound** (**Chad Vicenik**)
  Concatenate (daisy-chain) a Sound object with itself the specified number of times.
- **combine-sounds** (**Chris Darwin**, back up **here**)
  Combine (merge) two Sounds with specified gains.
- **script-installation-script** (**Niels Petersen**)
  An example of a script used to install several scripts to the Praat menus.
- **wave-maker** (**Kevin Ryan**)
  Create multiple varied sine waves at once in the object list and/or a directory (useful for testi

Text grid management

- **grid-maker** (**Kevin Ryan**)
  Make or edit text grids for a set of sound files.
  - See also **K. Crosswhite**'s amply commented grid **maker** and **reviewer** scripts (and
  - KR's version improves on these mainly by combining them: if a grid exists, it opens it,
- **label-from-text-file** (**Mietta Lennes**, back up **here**)
  Replace interval labels in selected TextGrid with labeled text from a file (ML's **description**
- **open-multiple-textgrids** (**John Tøndering**)
  Open multiple text grids from a directory at once.
- **mark-pauses** (**Mietta Lennes**, back up **here**)
  Mark pauses in a LongSound (can then run segmenter to get separate files) (ML's **descripti**
- **total-duration-of-labeled-segments** (**Mietta Lennes**, back up **here**)
  Total the duration of labeled segments of a TextGrid (ML's **description**).
- **align-textgrid-markers** (**Mietta Lennes**, back up **here**)
  Align TextGrid interval markers in tier one to those in tier two if they are sufficiently close (M

Analysis of sounds using text grids

- **calculate-segment-durations** (**Mietta Lennes**, back up **here**)

Plugin

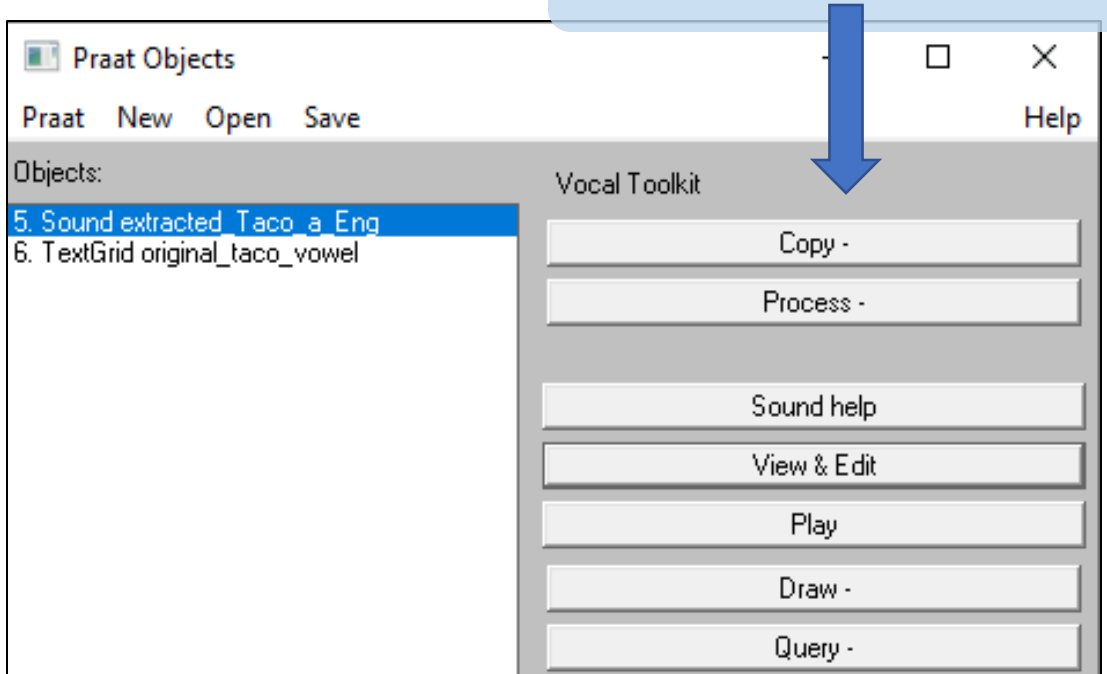
Vocal Toolkit

plugin with automated scripts for voice processing

Easy Steps!

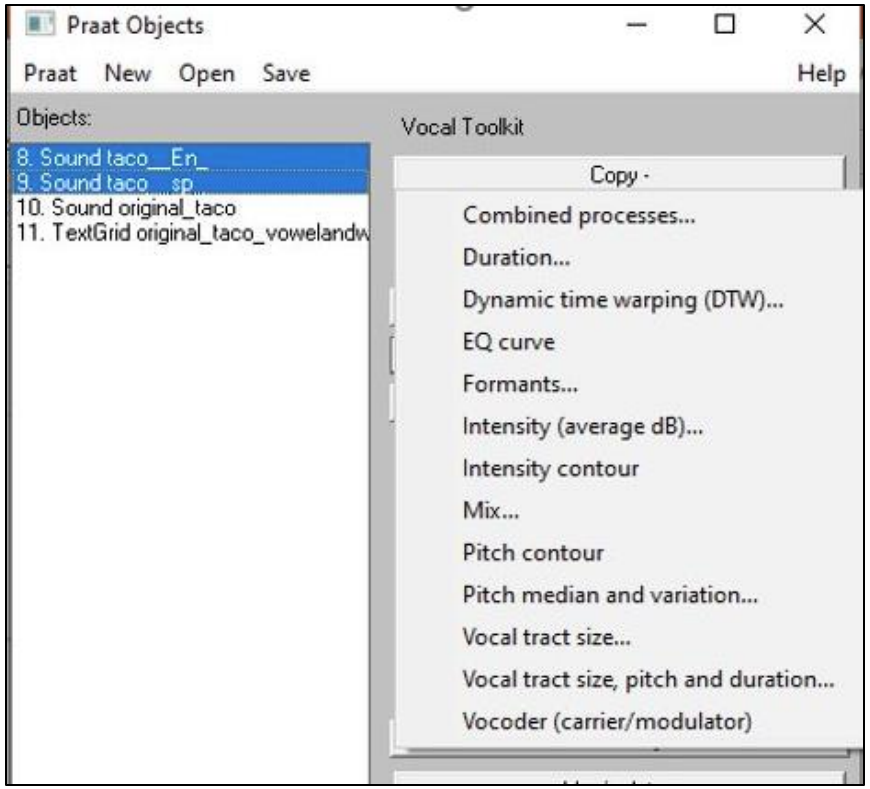
1) Download a folder from here
2) Praat > Open Praat Script > "INSTALL.praat" in the folder > Run
3) Restart

Now you see this in the menu

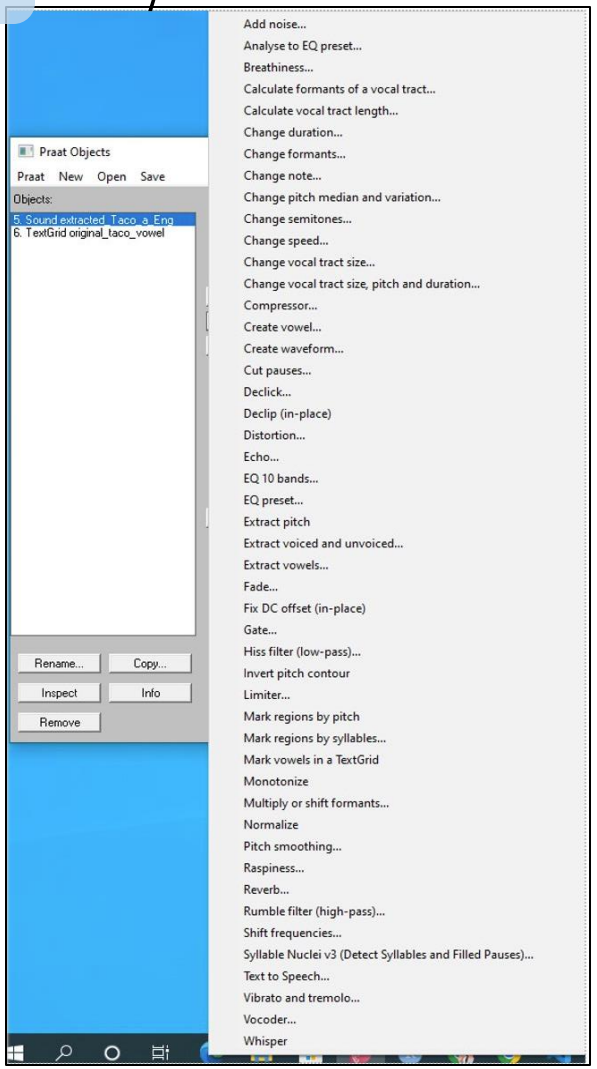


Vocal Toolkit

"Copy" ...for 2 sounds selection.



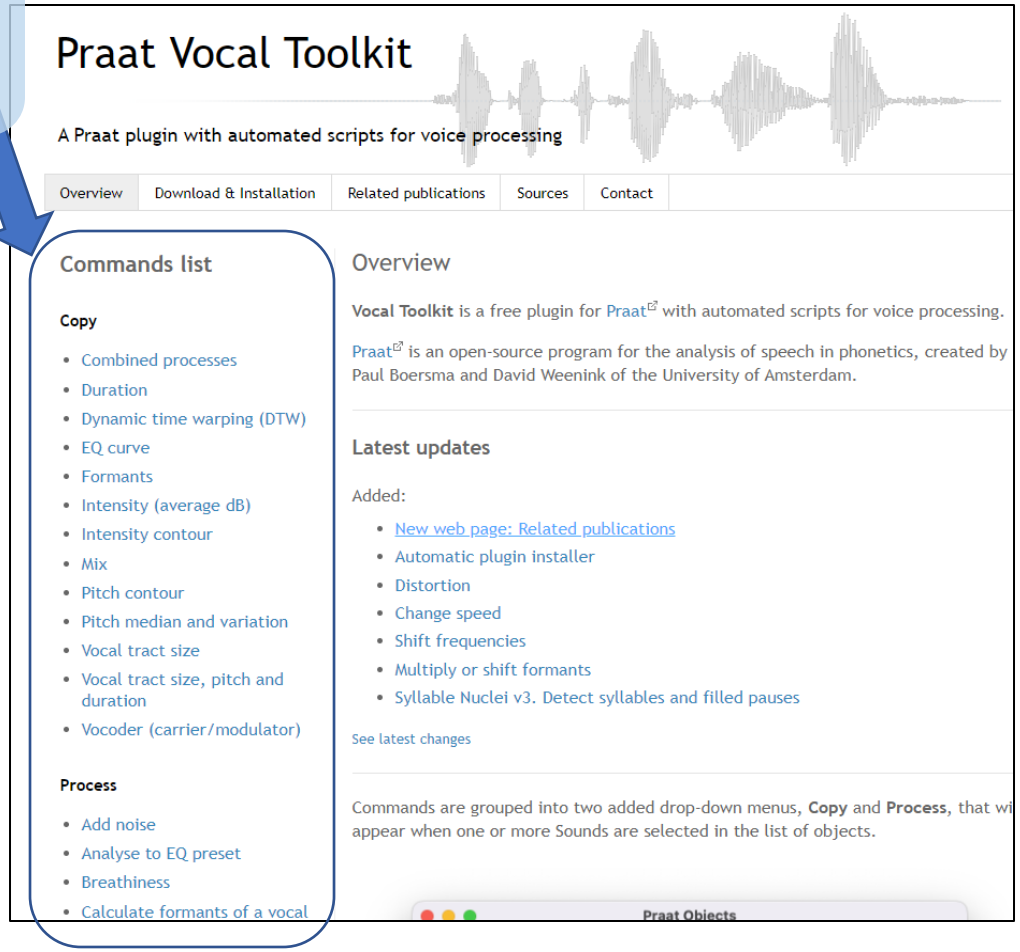
"Process" ...any sound selections / Create new



Vocal Toolkit

Go to the website to check what algorithms are used

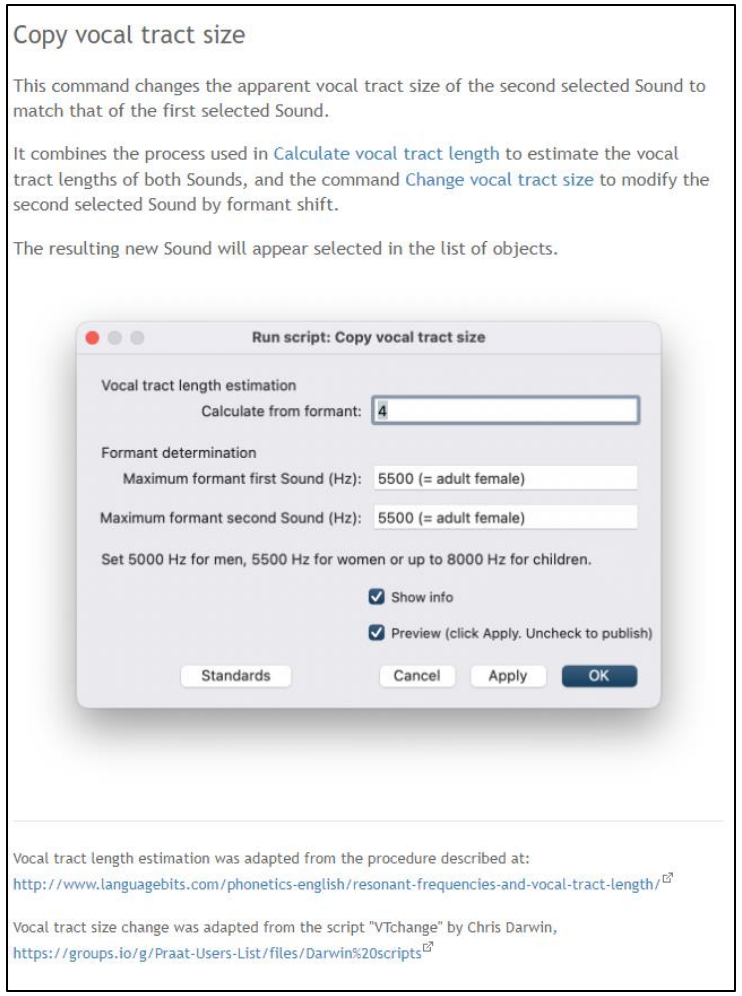
Find the command of your interest



Example: Copy > Vocal Tract Size

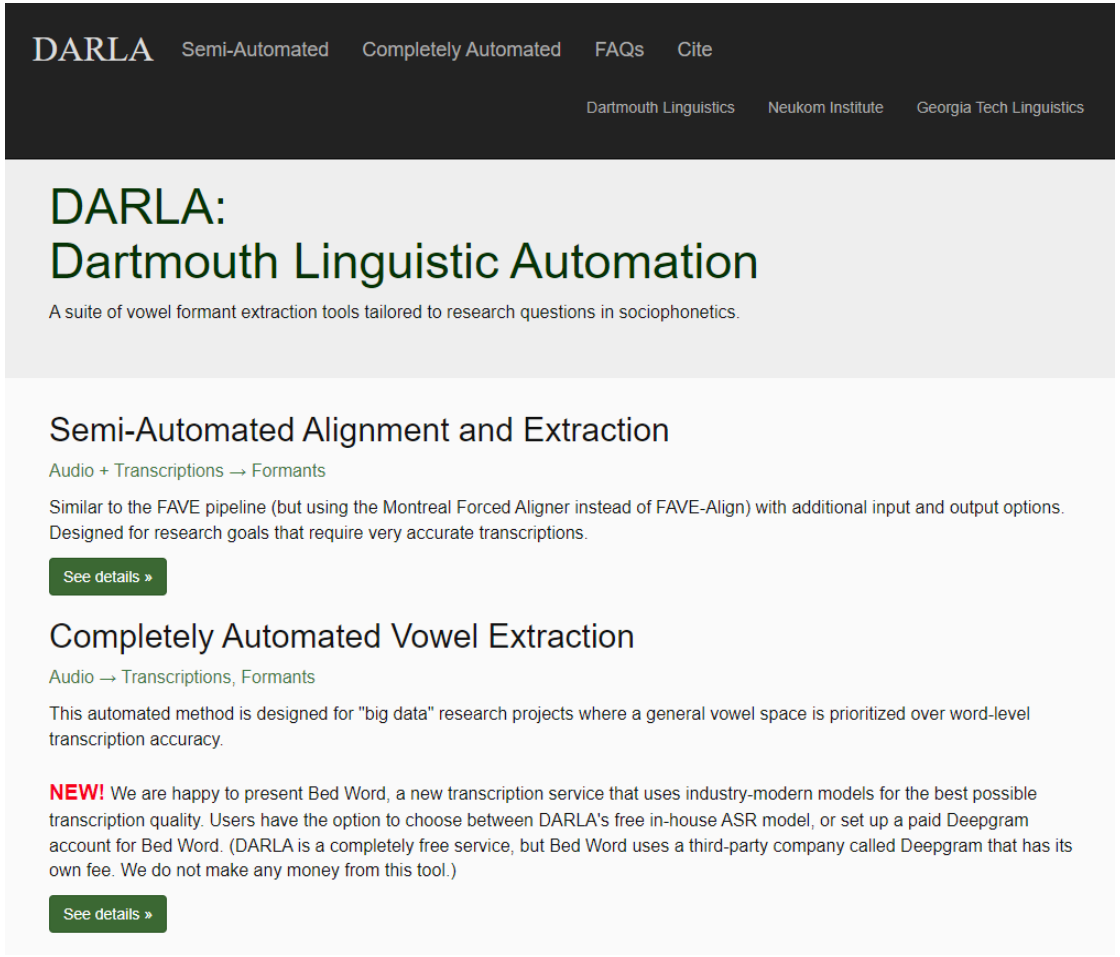
Explanations

References for algorithm



Forced Alignment

DARLA: automated vowel extraction



1. Semi-Automated Alignment & Extraction
   1. Audio with transcriptions showing sentence or breath group boundaries (as TextGrids)
   2. Audio with transcriptions in a plaintext file
   3. Audio with aligned TextGrids, for formant extraction only
2. Completely Automated Vowel Extraction
   1. Bed Word: Automated Interview Transcription via Deepgram
   2. Audio with transcriptions provided by our in-house speech recognition
   3. ASR evaluation

Forced Alignment

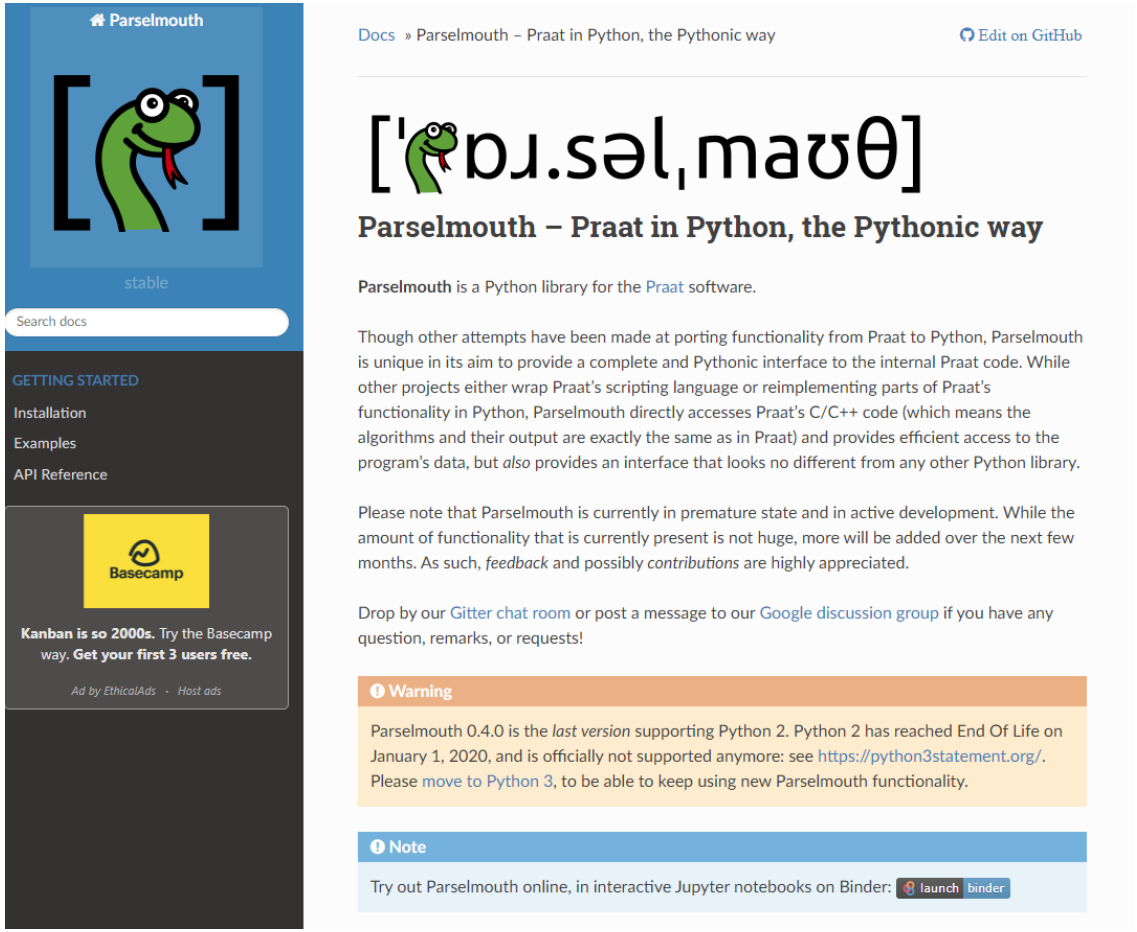
DARLA: automated vowel extraction

Title	Input	Output
1.1 force alignment of vowels and get formants info, from sound & TextGrid (recommended)	1) sound file 2) manual transcription TextGrid (with sentence & breath group annotated)	Forced vowel alignment & Formants
1.2 get formants from sound & txt file	1) sound file 2) transcript in plaintext file (.txt) (can be manual or autocreated)	formants
1.3 get formants from sound & textgrid	1)sound 2)manually aligned TextGrid (word/phoneme)	Formants
2.1 Newer/more precise system (costs money $0.75/hr for a 3rd party system)	1) deepgram account API key 2) Sound file	transcripts
2.2 their legacy recog system; using Montreal Forced Aligner & FAVE-Extract	sound file	transcripts
2.3 evaluate ASR transcription error rates at word/phoneme level	1) Manual transcription as plain text file 2) ASR OR another manual transcription	Error rates for words, phonemes and stressed vowels

Parselmouth

Praat in Python!

(Link to the website)



To install:

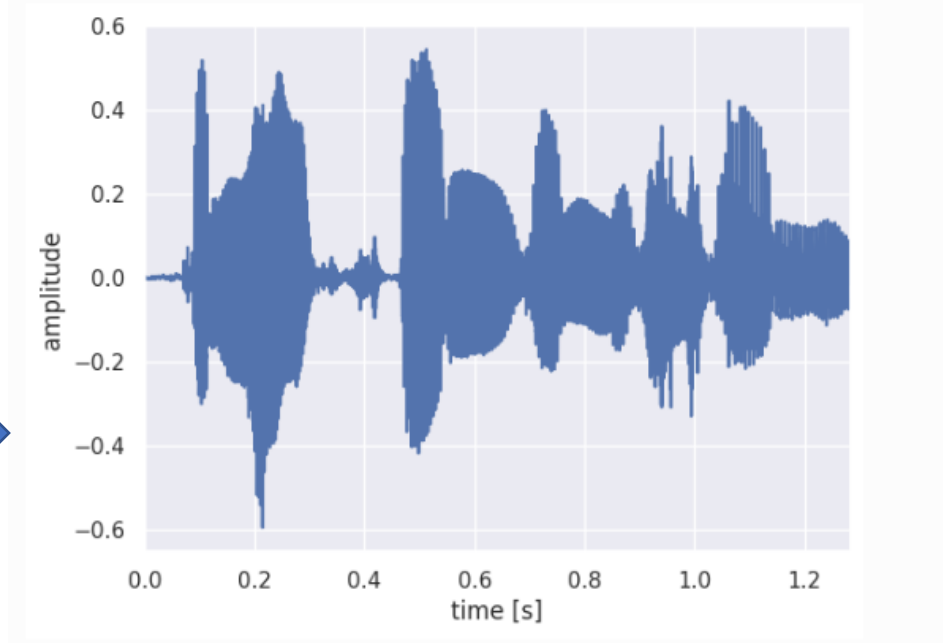
```
PS C:\Users\yzt5262> pip install praat-parselmouth
```

Not this:

Parselmouth

Example from the website

```
[1]: import parselmouth

     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns

[2]: sns.set() # Use seaborn's default style to make attractive graphs
     plt.rcParams['figure.dpi'] = 100 # Show nicely large images in this notebook

[3]: snd = parselmouth.Sound("audio/the_north_wind_and_the_sun.wav")

[4]: plt.figure()
     plt.plot(snd.xs(), snd.values.T)
     plt.xlim([snd.xmin, snd.xmax])
     plt.xlabel("time [s]")
     plt.ylabel("amplitude")
     plt.show() # or plt.savefig("sound.png"), or plt.savefig("sound.pdf")
```



[θæŋkju]!!

Reach out to me if you got any questions about Praat:)

yzt5262@psu.edu

References

I borrowed the idea of "Input" "Output" from the slides made by Dr. Eleanor Chodroff (https://www.eleanorchodroff.com/tutorial/PraatScripting.pdf)

Also, Listen Lab Praat scripts were very helpful to doublecheck my intensity manipulation script (https://www.youtube.com/watch?v=j5GDJs0st_Q) (by Dr.Matt Winn: http://www.mattwinn.com/)

Other references for both of my Intermediate and Basic level workshop are listed here (with my personal summary notes for each): Praat List of Resources.docx

https://pennstateoffice365-my.sharepoint.com/:w:/r/personal/fkb1_psu_edu/Documents/Box%20Migration%20Data/CLS_LabManagers/Software,%20Hardware,%20and%20Tech/Software%20Information/Praat/Praat_Workshop/Praat%20Workshop%20(Yuka)/Praat%20List%20of%20Resources.docx?d=w7c2e8428eb8e4640ac90bd8dc3137412&csf=1&web=1&e=ZFlPR4