Praat Scripting

LING497
March 21st & 23rd
Yuka Tatsumi

Overview

1. About Praat Scripting
   a. What is scripting?
   b. Why can you do?
   c. Why you want to learn this?
   d. Let's write a very simple script!
2. Basics to learn before scripting
   a. General format of scripts
   b. 7 coding basics
3. WORKSHOP -- writing your first script!
   a. Thinking about the goal and structure
   b. Title → Form → Main Loop → Read Me → Saving outputs
   c. Extra: Using the output table to make a vowel plot in R
4. Next step: Advancing your Praat skills

Part 1

About Praat Scripting

What is scripting?

Short Answer:

Language for Praat

You write to tell Praat what you want it to do
→ Praat will do it for you, instead of clicking around GUI.

GUI = graphical user interface. Easy & Intuitive platform you can interact with systems.

What can you do?

Short Answer:

Everything you can do with Praat GUI!

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

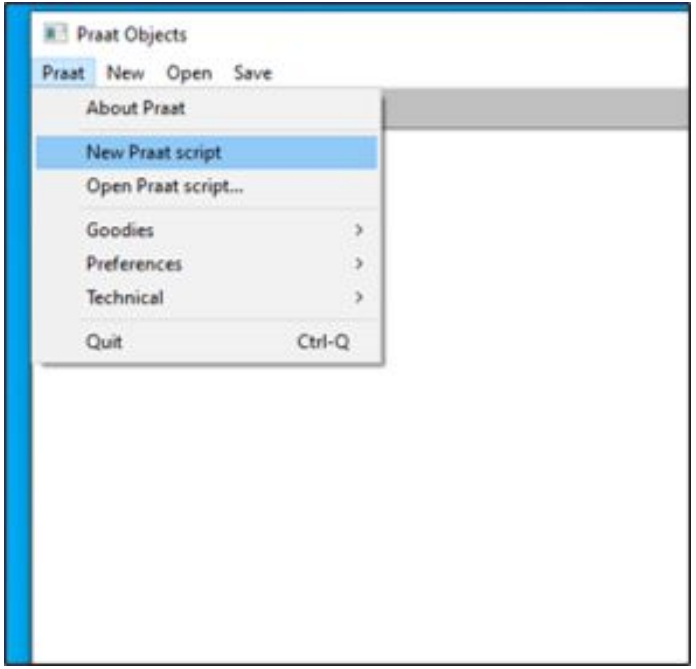
Why you want to learn this?

Short Answer:
To automate Praat processings for a lot of data.

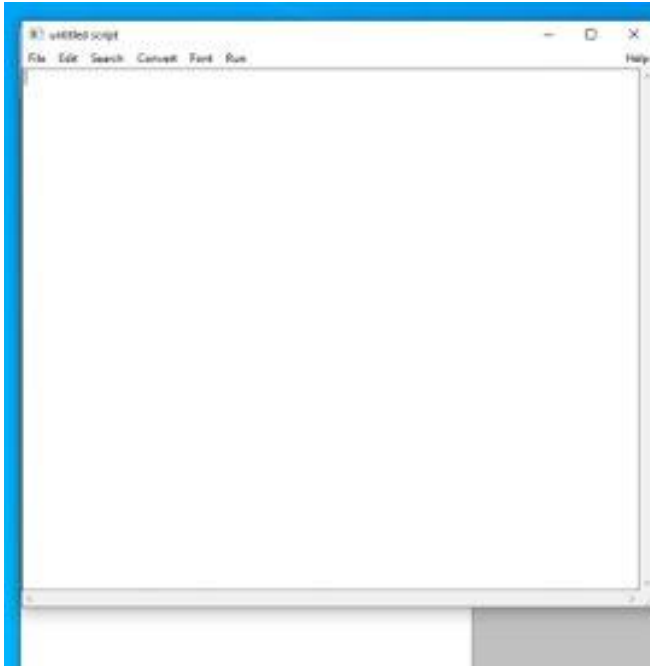
You could use others' script online...but learning how to script by yourself can really help you understand others' scripts, and apply others' scripts to your data!

Let’s write a very simple script!

1. Open a new script

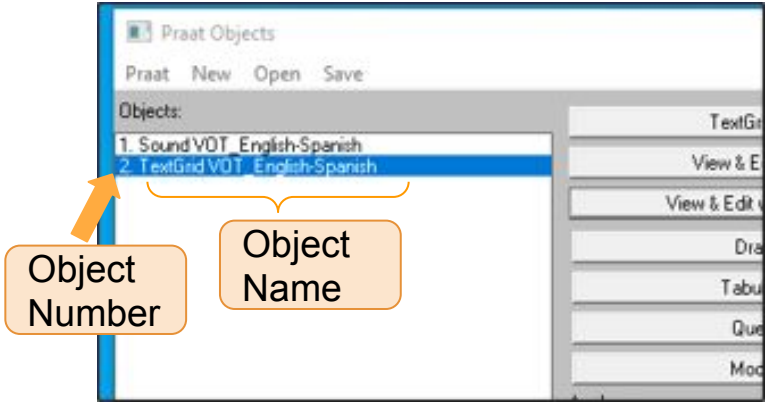


2. this is your canvas!

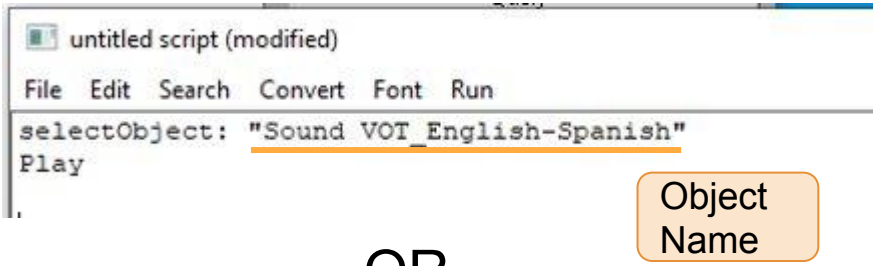


Let’s write a very simple script!

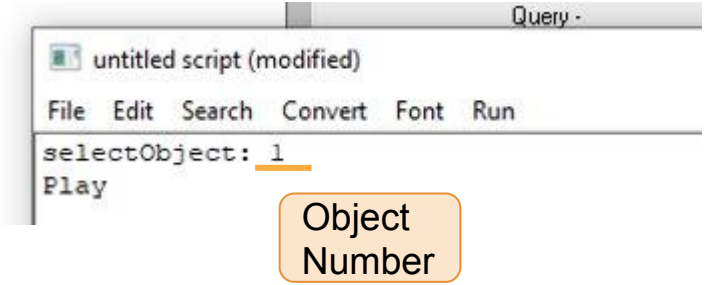
3. Import a sound & a TextGrid to your Object Window



4. Write this and run.



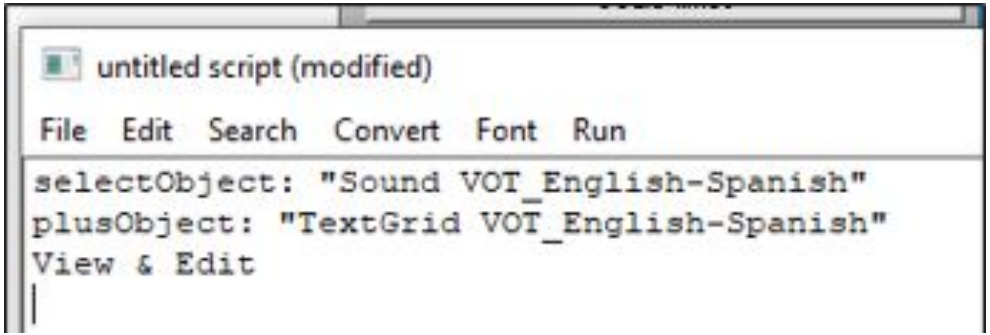
OR



Let’s write a very simple script!

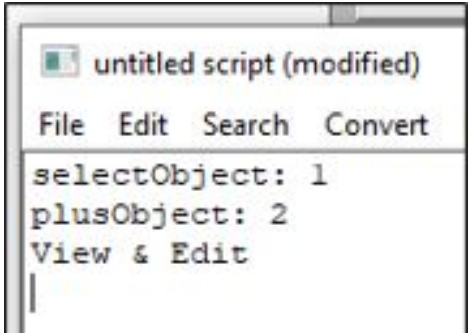
5. Write this and Run (Ctrl + R)

Object Name



OR

Object Number



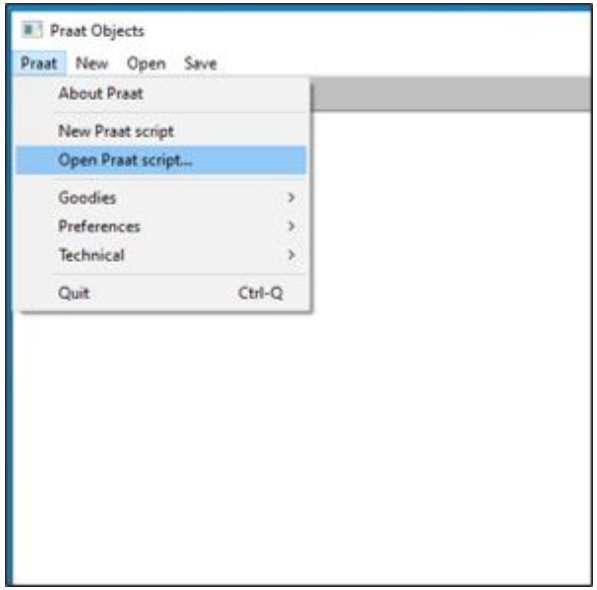
Part 2

Basics to Learn Before Scripting

Format & syntax

Let's look at an example script

1. Open an existing script



Let's look at an example script

praat script to save many 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!

General Script Structure

1

2.1

2.2

2.3&2.4

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

To write a full script like this, there are a few basics we need to learn…

Coming next:

1. 3 things to remember
2. Calculations
3. Texts
4. Variables
5. Commands & Arguments
6. Loops
7. if statement

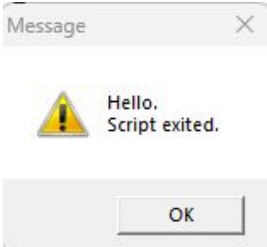
1) 3 things to remember first

In general, Praat reads what you wrote from the top to the bottom, line by line.

1. “#” = Comment out

if type # before any lines, Praat will not recognize that line.
We use this to write a note.

2. exitScript: “Hello.”



Praat ends to read lines when it encounters this.
Then, gives a little Message window with texts/values after colon. In this case, “Hello.”
“Script exited.” is always at the end of the window.
We use this often to test the script partially.

3. writeInfoLine: “Hello.”

Praat creates a text window with texts/values after colon.
Praat does not end to read lines by this.

2) Calculations

1. Write this and Run

untitled script (modified)

File Edit Search Convert Font Run

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

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

2. this script will produce:

Praat Info

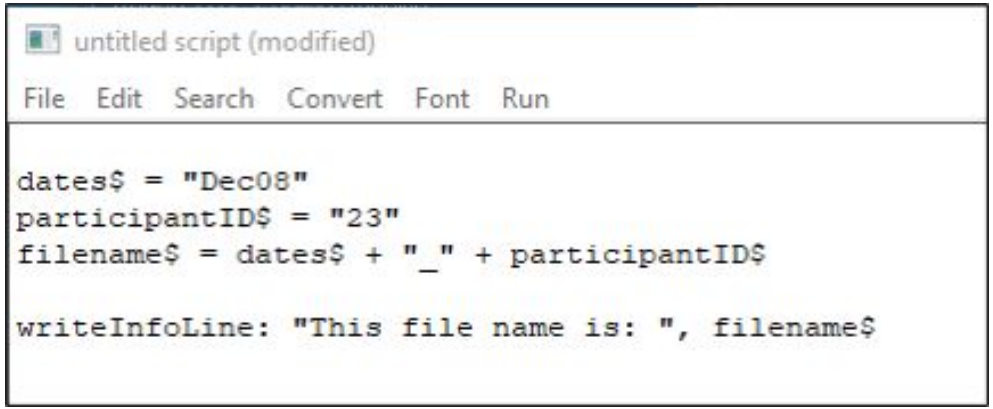
File Edit Search Convert Font Help

```
The value of z is 5.
```

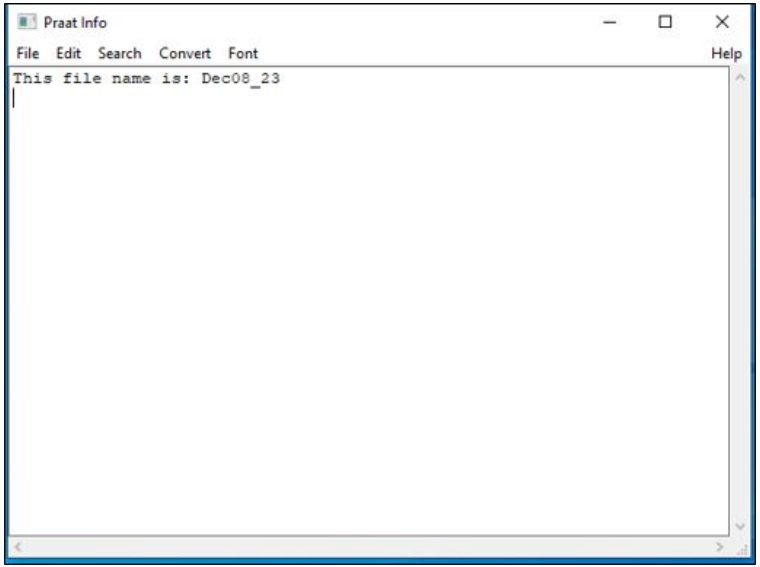
This means:
You assigned the value of **3** to the variable **x**.
You assigned the value of **2** to the variable **y**.
You assigned the value of **“x+y”** to the variable **z**.
You told Praat to create a window that writes: **“The value of z is ”**, the value of **z**, and then “**.**”

3) Texts

1. Write this and Run



2. this script will produce:



This means:
You assigned the text “Dec08” to the variable **dates$**.
You assigned the text “23” to the variable **participantID$**.
You assigned the text connecting **dates$ value, “_”, and participantID$ value**, to the variable **filename$**.
You told Praat to create a window that writes: “**The file name is:** ”, and then the text inside the **filename$** variable.

4) Summary of Variables in Praat

"texts" … **String Variable**

Variable names always 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
- commands can follow after "*variable* = "

Praat is case sensitive!

Praat is NOT whitespace sensitive!
(= ignores spaces and indents)
Indentation is only for readability.

Praat does not autosave
Save the script frequently!

5) Commands & Arguments

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

Command

command to get a mean pitch from “Pitch” object

Arguments

5) Commands & Arguments

Another example:

Telling Praat to make a TextGrid of selected Sound:

```
To TextGrid: "word phonemes comment", "comment"
```

Command **Arguments**

6) Loops

= 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. "i" is chosen just as a convention.

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

7) if statement

= Telling Praat which codes to run depending on conditions

Example Use

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

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

Part 3

Workshop
- Let’s write a full script! -

Disclaimer: This is structured to show a possible flow of thoughts when one writes a script, so the order of explanation is not following the final script’s code order (we go back and forth to write). If you prefers to follow the order of the script to understand, please see the “with explanations” version of the script on my GitHub.

GOAL

Last time:
You made
- Sound with vowels ([bVd] and [hVd])
- TextGrid (w/ a vowel tier)

and then manually made a table of F1&F2 means on Excel.

Today:
We have these Sound&TextGrid sets for all students and want to process all at once.

As a product, we want a table

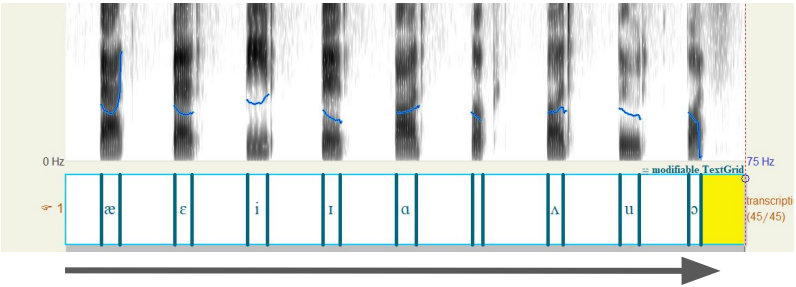
STEP0: Plan the structure

Today:
We have these Sound&TextGrid sets for all students and want to process all at once.

As a product, we want a table

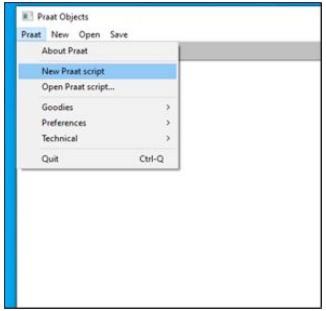
We want to …
Loop (=repeat) below for each prsn's data:
1)Read a Sound and its Textgrid of a person

2)Look at each interval from left to right, and
- only if the interval is not empty,
  - get the IPA of the interval
  - get F1 and F2 of midpoint
  - save these information on the table

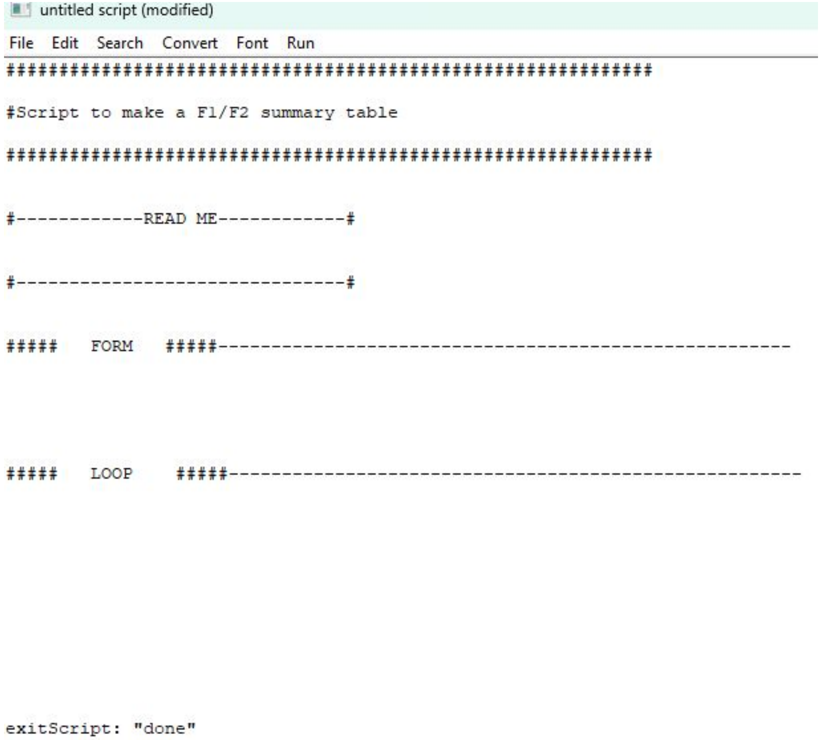


STEP1: Write the broad structure

1. Open a new script



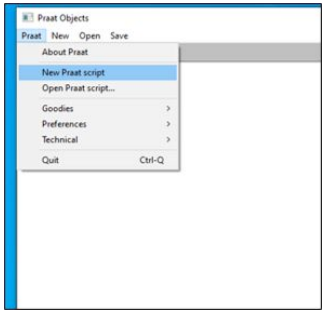
2. Make a title, readme, form, & loop sections



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

#Script to make a F1/F2 summary table

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


#------------READ ME------------#


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


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



#####   LOOP    #####------------------------------------------------------






exitScript: "done"
```

STEP1: Write the broad structure

1. Open a new script



2. Make a title, readme, form, & loop sections

untitled script (modified)

File Edit Search Convert Font Run

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

#Script to make a F1/F2 summary table

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

#------------READ ME------------#

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

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

#####   LOOP   #####-------------------------------------------------------

exitScript: "done"
```

STEP 2 : Title section

Write a title on the top.

untitled script (modified)

File Edit Search Convert Font Run

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

#Script to make a F1/F2 summary table

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

#------------READ ME------------#


#-------------------------------#
```

When you write an original script and if you plan to make it public, you might also want to write your name & dates as well to claim the authorship.

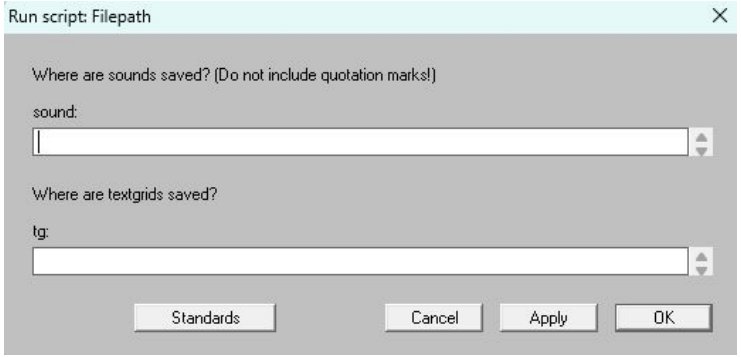
Reminder

Praat does not autosave
Save the script frequently!

STEP 3 : Form section

Making a form
= make a little pop up window to enter information before you run a script

we are making this today:



WHY MAKE THIS?
To give users a chance to specify some variables in the script.

Example (today's case)
You will write a script line to import sounds & TextGrids, but you do not know if the location of them will always be the same in the future.
So you make a form to allow users to fill out the file addresses.

STEP 3 : Form section

Write this:

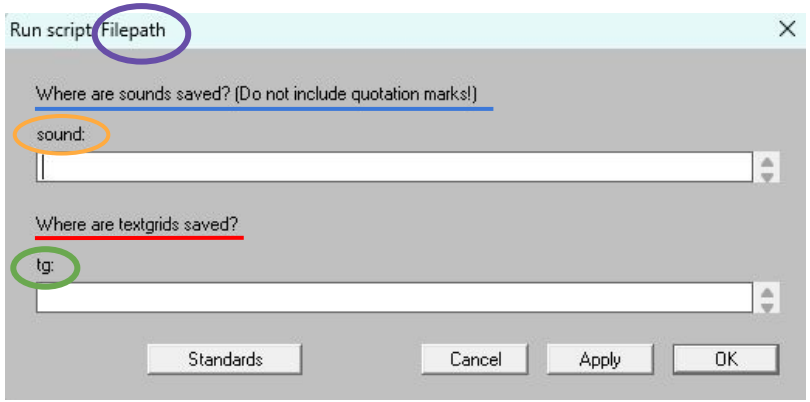
```
#####   FORM   #####-------------------------------------------------------

form Filepath
        comment Where are sounds saved? (Do not include quotation marks!)
        text sound

        comment Where are textgrids saved?
        text tg
endform
```

STEP 3 : Form section

Write this:

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

form Filepath
        comment Where are sounds saved? (Do not include quotation marks!)
        text sound

        comment Where are textgrids saved?
        text tg
endform
```

This will produce:



STEP 4 : Thinking of what we need to add before Loop section

Again, this is what we want:

Loop (=repeat) below for each prsn's data:
1)Read a Sound and its Textgrid of a person

2)Look at each interval from left to right, and
- only if the interval is not empty,
  - get the IPA of the interval
  - get F1 and F2 of midpoint
  - save these information on the table

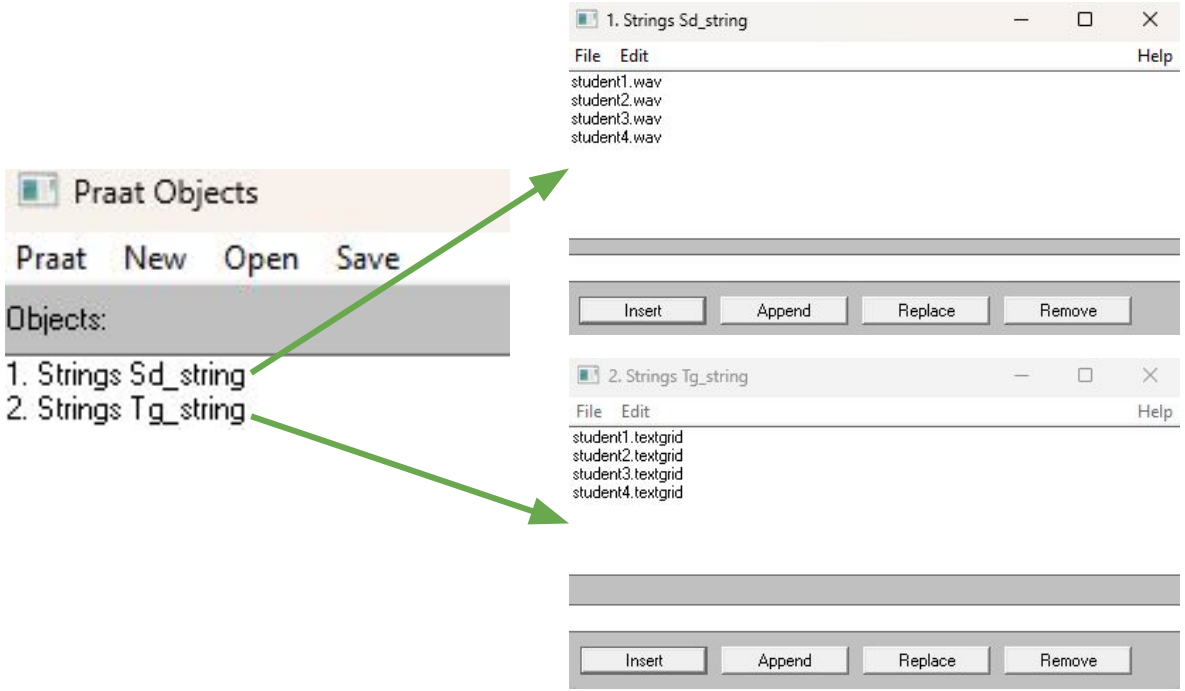
Before loop, we have to prepare:

1)String objects to store lists of sound and TextGrid file names
(so that we can read it one by one, by combining the text strings with folder address)

2)a Table object to store information

STEP 4 : 1) Prep String Objects

We are making these



STEP 4 : 1) Prep String Objects

Write this under the “endform”:

```
#Make string objects---------------
sdID = Create Strings as file list: "Sd string", sound$ + "/*.wav"
tgID = Create Strings as file list: "Tg string", tg$ + "/*.TextGrid"
```

This variable stores the folder address for sounds. We got this from the “form”

This variable stores the folder address for TextGrids. We got this from the “form”

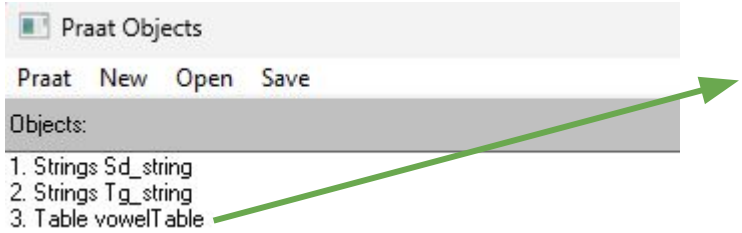
Meaning of these:

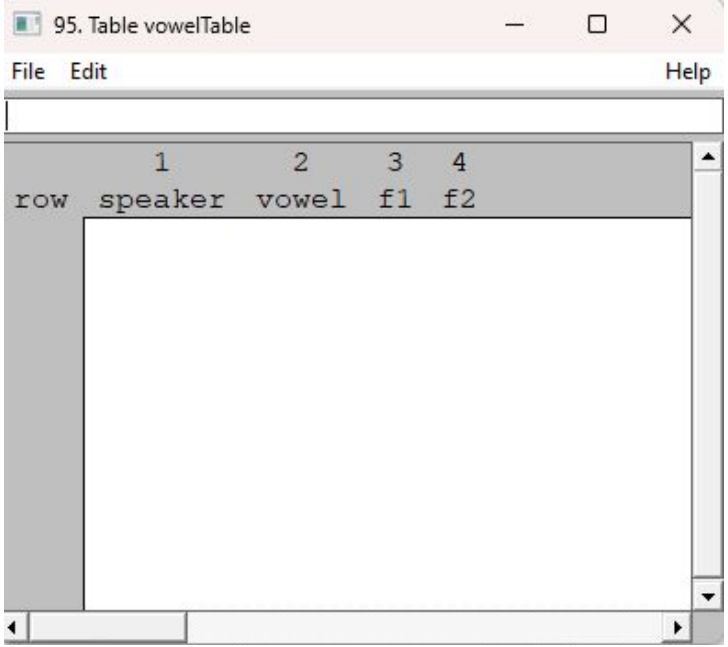
Line 1:
Within the sound file folder (stored in sound$), only look for files that have “.wav.” Then create a list of their file names, in a String object. Name that object as “Sd string.” Remember this object’s object ID in “sdID” variable.

Line2:
Within the TextGrid file folder (stored in tg$), only look for files that have “.TextGrid.” Then create a list of their file names, in a String object. Name that object as “Tg string.” Remember this object’s object ID in “tgID” variable.

STEP 4 : 2) Prep an output Table

We are making this empty table:





STEP 4 : 2) Prep an output Table

Write this under the “tgID” line:

```
#Output Table-----------
tableID = Create Table with column names... vowelTable 0 speaker vowel f1 f2
```

Meaning of these:

Name this table as “vowelTable”.
Do not add any rows now. (“0”)
Make following columns from left to right: “speaker”, “vowel”, “f1”, “f2”
Assign this Table object’s object ID to “tableID” variable.

STEP 5 : Loop section

Again, this is what we want:

Loop (=repeat) below for each prsn's data:
1)Read a Sound and its Textgrid of a person

2)Look at each interval from left to right, and
- only if the interval is not empty,
  - get the IPA of the interval
  - get F1 and F2 of midpoint
  - save these information on the table

Let's start with writing that structure under #### LOOP ####----------

STEP 5 : Loop section

Write:

```
#####   LOOP    #####-----------------------

for i to nTg

        for k to nIntervals

                if kLabel$ <> ""


                endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Write:

```
#####   LOOP    #####-----------------------

for i to nTg
        "Repeat below for every person's data"
        (nTg = number of TextGrids we have)
    for k to nIntervals
            "Within this TG, repeat below for every interval "
            (nInterval = number of Intervals the TG has)
        if kLabel$ <> ""
            "Do below if kLabel variable is not empty"
            (<> = "is not equal to") ( "" = no texts. empty.)
        endif
    endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

We need to define nTg (= to get a number of files we have, so that we can loop for that amount of times)

```
#####   LOOP    #####-----------------
```

Add

```
selectObject: tgID
nTg = Get number of strings
for i to nTg


        for k to nIntervals


                if kLabel$ <> ""



                endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

For each loop, we want to import a sound and textgrid.

```
#####   LOOP    #####-------------------------------------------------------

selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgID = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundID = Read from file: sound$ + "/" + targetSoundName$

        for k to nIntervals


                if kLabel$ <> ""



                endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

For each loop, we want to import a sound and textgrid.

```
for i to nTg
    selectObject: tgID
    targetTgName$ = Get string: i
    targetTgID = Read from file: tg$ + "/" + targetTgName$

    selectObject: sdID
    targetSoundName$= Get string: i
    targetSoundID = Read from file: sound$ + "/" + targetSoundName$

    for k to nIntervals
```

Meaning

Grab the String object for TG

↓

Get *the number i* file name and store it in “targetTgName$” variable

↓

Read the TG file with its adress that made by combining
TG folder adress (tg$) & filename (targetTgName$)
Store its ID in targetTgID

…and the same for sound

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

We need to define nIntervals (= to get the number of intervals this textgrid has, to repeat the same process for all intervals)

```
#####   LOOP    #####-------------------------------------------------------

selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgID = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundID = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgID
        nIntervals = Get number of intervals: 1
        for k to nIntervals


                if kLabel$ <> ""



                endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

We need to define kLabel$ (= to get the label of *the number k* interval, so that we can use it to check if the interval has a vowel in it)

```
#####   LOOP    #####-------------------------------------------------------

selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgID = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundID = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgID
        nIntervals = Get number of intervals: 1
        for k to nIntervals
            selectObject: targetTgID
            kLabel$ = Get label of interval: 1, k
            if kLabel$ <> ""



            endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Now we want to write what Praat does for an interval that has a vowel. We need these information for each interval: the vowel, F1, and F2.

First, Let's save a kLabel$ in vowel$ for clarity (we'll use this info when adding info on the output table)

```
#####   LOOP    #####-------------------------------------------------

selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgID = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundID = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgID
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgID
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                    vowel$ = kLabel$


                endif
        endfor
endfor
```

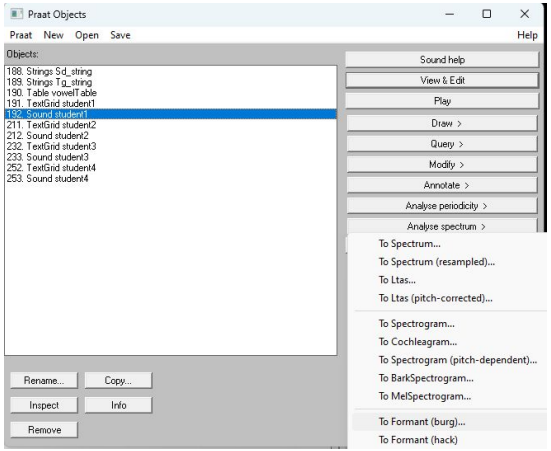
Progress Checker

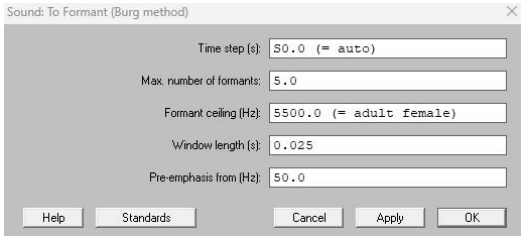
```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Next, Let's get F1 and F2.
We want to take a midpoint of that interval and get F1&2 at that point.

To get F1 and F2 in Praat GUI Object Menu, you would do:
select Sound object > Analyse spectrum > To Formant (burg)...
and you would see this pop up window ↓ and click OK



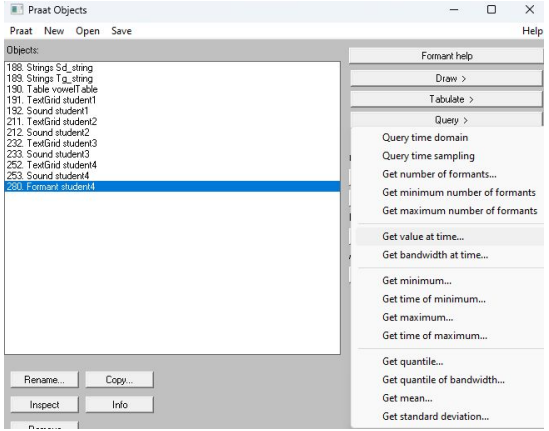


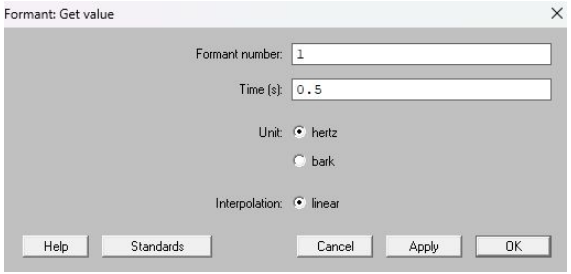
Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Then, select that Formant Object > Query > Get value at time…
and you will see this pop up window↓





Then, type in the formant number of interest and the time point you want to measure.

So we can just write this exact process in the script!

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Write:

```
#####   LOOP    #####-------------------------------------------------------

selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgID = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundID = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgID
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgID
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundID
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                endif
        endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Meaning:

Getting time of the midpoint of interval

1. timeS = start time of the interval
2. timeE = end time of the interval
3. intDuration = duration of the interval (subtracting timeS from timeE)
4. midpoint = adding the half of the duration interval to the timeS

Same as the GUI Processing

1. selecting the sound object
2. Making Formant object (arguments are the defaults)
3. f1… “1” = formant 1. Feeding “midpoint” variable value as a time point. (the rest of the arguments are the defaults)
4. f2… same as above except the formant number.

```
timeS = Get start time of interval: 1, k
timeE = Get end time of interval: 1, k
intDuration = timeE - timeS
midpoint = timeS + (intDuration/2)
selectObject: targetSoundID
formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
f1 = Get value at time: 1, midpoint, "hertz", "linear"
f2 = Get value at time: 2, midpoint, "hertz", "linear"
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
    selectObject: tgID
    targetTgName$ = Get string: i
    targetTgFile = Read from file: tg$ + "/" + targetTgName$

    selectObject: sdID
    targetSoundName$= Get string: i
    targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

    selectObject: targetTgFile
    nIntervals = Get number of intervals: 1
    for k to nIntervals
        selectObject: targetTgFile
        kLabel$ = Get label of interval: 1, k
        if kLabel$ <> ""
            vowel$ = kLabel$
            timeS = Get start time of interval: 1, k
            timeE = Get end time of interval: 1, k
            intDuration = timeE - timeS
            midpoint = timeS + (intDuration/2)
            selectObject: targetSoundFile
            formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
            f1 = Get value at time: 1, midpoint, "hertz", "linear"
            f2 = Get value at time: 2, midpoint, "hertz", "linear"

            selectObject: tableID
            Append row
            tblrow = Get number of rows
            Set numeric value: tblrow, "speaker", i
            Set string value: tblrow, "vowel", vowel$
            Set numeric value: tblrow, "f1", f1
            Set numeric value: tblrow, "f2", f2

            selectObject: formantID
            Remove
        endif
    endfor
endfor
```

STEP 5 : Loop section

Lastly, let's append these info in the output table!

```
        kLabel$ = Get label of interval: 1, k
        if kLabel$ <> ""
                vowel$ = kLabel$
                timeS = Get start time of interval: 1, k
                timeE = Get end time of interval: 1, k
                intDuration = timeE - timeS
                midpoint = timeS + (intDuration/2)
                selectObject: targetSoundID
                formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                f1 = Get value at time: 1, midpoint, "hertz", "linear"
                f2 = Get value at time: 2, midpoint, "hertz", "linear"

                selectObject: tableID
                Append row
                tblrow = Get number of rows
                Set numeric value: tblrow, "speaker", i
                Set string value: tblrow, "vowel", vowel$
                Set numeric value: tblrow, "f1", f1
                Set numeric value: tblrow, "f2", f2
        endif
    endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Meanings of the last 4 lines:

add value “i” to the “speaker” column, on #“tblrow” row.

add value “vowel$” to the “vowel” column, on #“tblrow” row.

add value “f1” to the “f1” column, on #“tblrow” row.

add value “f2” to the “f2” column, on #“tblrow” row.

```
selectObject: tableID
Append row
tblrow = Get number of rows
Set numeric value: tblrow, "speaker", i
Set string value: tblrow, "vowel", vowel$
Set numeric value: tblrow, "f1", f1
Set numeric value: tblrow, "f2", f2
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
	selectObject: tgID
	targetTgName$ = Get string: i
	targetTgFile = Read from file: tg$ + "/" + targetTgName$

	selectObject: sdID
	targetSoundName$= Get string: i
	targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

	selectObject: targetTgFile
	nIntervals = Get number of intervals: 1
	for k to nIntervals
		selectObject: targetTgFile
		kLabel$ = Get label of interval: 1, k
		if kLabel$ <> ""
			vowel$ = kLabel$
			timeS = Get start time of interval: 1, k
			timeE = Get end time of interval: 1, k
			intDuration = timeE - timeS
			midpoint = timeS + (intDuration/2)
			selectObject: targetSoundFile
			formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
			f1 = Get value at time: 1, midpoint, "hertz", "linear"
			f2 = Get value at time: 2, midpoint, "hertz", "linear"

			selectObject: tableID
			Append row
			tblrow = Get number of rows
			Set numeric value: tblrow, "speaker", i
			Set string value: tblrow, "vowel", vowel$
			Set numeric value: tblrow, "f1", f1
			Set numeric value: tblrow, "f2", f2

			selectObject: formantID
			Remove
		endif
	endfor
endfor
```

STEP 5 : Loop section

Lastly, let's append these info in the output table!

```
kLabel$ = Get label of interval: 1, k
if kLabel$ <> ""
        vowel$ = kLabel$
        timeS = Get start time of interval: 1, k
        timeE = Get end time of interval: 1, k
        intDuration = timeE - timeS
        midpoint = timeS + (intDuration/2)
        selectObject: targetSoundID
        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
        f1 = Get value at time: 1, midpoint, "hertz", "linear"
        f2 = Get value at time: 2, midpoint, "hertz", "linear"

        selectObject: tableID
        Append row
        tblrow = Get number of rows
        Set numeric value: tblrow, "speaker", i
        Set string value: tblrow, "vowel", vowel$
        Set numeric value: tblrow, "f1", f1
        Set numeric value: tblrow, "f2", f2
endif
    endfor
endfor
```

Grab the table object

Adding a row

Get the row number

Adding values to each column on that row

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Loop section

Optional: **Cleaning**

Every Time an interval is processed, it's creating a Formant object. To erase it before going to the next interval, you can add these lines right before you close the conditional processing with endif:

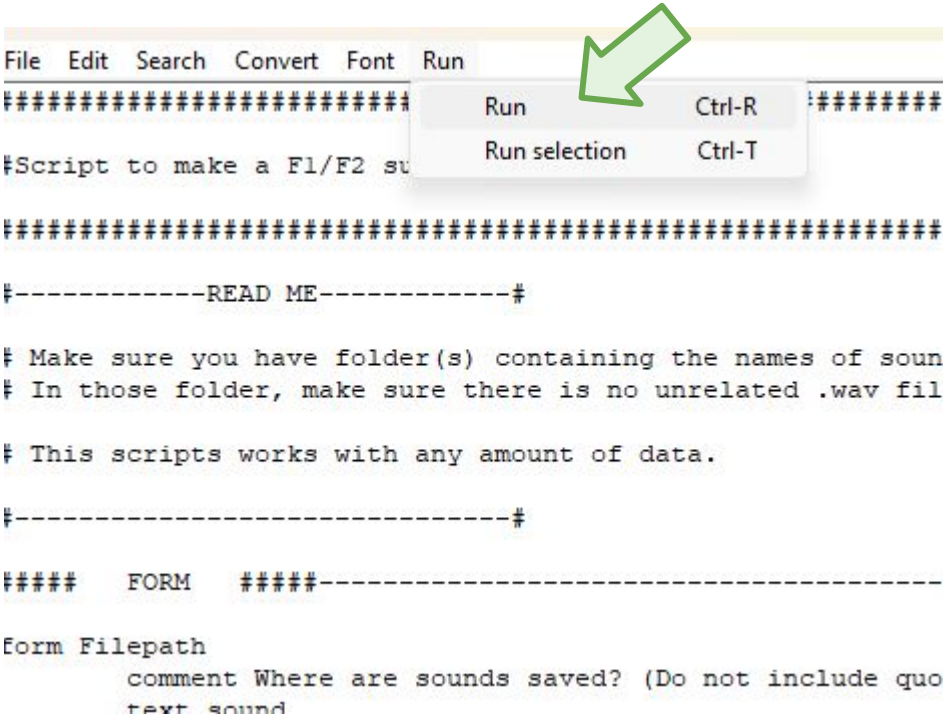
```
if kLabel$ <> ""
        vowel$ = kLabel$
        timeS = Get start time of interval: 1, k
        timeE = Get end time of interval: 1, k
        intDuration = timeE - timeS
        midpoint = timeS + (intDuration/2)
        selectObject: targetSoundID
        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
        f1 = Get value at time: 1, midpoint, "hertz", "linear"
        f2 = Get value at time: 2, midpoint, "hertz", "linear"

        selectObject: tableID
        Append row
        tblrow = Get number of rows
        Set numeric value: tblrow, "speaker", i
        Set string value: tblrow, "vowel", vowel$
        Set numeric value: tblrow, "f1", f1
        Set numeric value: tblrow, "f2", f2

        selectObject: formantID
        Remove
endif
endfor
endfor
```

Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 5 : Done with the Loop!

Let's run and see if it works!



Progress Checker

```
selectObject: tgID
nTg = Get number of strings
for i to nTg
        selectObject: tgID
        targetTgName$ = Get string: i
        targetTgFile = Read from file: tg$ + "/" + targetTgName$

        selectObject: sdID
        targetSoundName$= Get string: i
        targetSoundFile = Read from file: sound$ + "/" + targetSoundName$

        selectObject: targetTgFile
        nIntervals = Get number of intervals: 1
        for k to nIntervals
                selectObject: targetTgFile
                kLabel$ = Get label of interval: 1, k
                if kLabel$ <> ""
                        vowel$ = kLabel$
                        timeS = Get start time of interval: 1, k
                        timeE = Get end time of interval: 1, k
                        intDuration = timeE - timeS
                        midpoint = timeS + (intDuration/2)
                        selectObject: targetSoundFile
                        formantID = To Formant (burg): 0, 5, 5500, 0.025, 50
                        f1 = Get value at time: 1, midpoint, "hertz", "linear"
                        f2 = Get value at time: 2, midpoint, "hertz", "linear"

                        selectObject: tableID
                        Append row
                        tblrow = Get number of rows
                        Set numeric value: tblrow, "speaker", i
                        Set string value: tblrow, "vowel", vowel$
                        Set numeric value: tblrow, "f1", f1
                        Set numeric value: tblrow, "f2", f2

                        selectObject: formantID
                        Remove
                endif
        endfor
endfor
```

STEP 6 : Readme section

Once you write a whole script that works, write all things users need to note upon the script use (including yourself!) in the Readme section.

untitled script (modified)

File Edit Search Convert Font Run

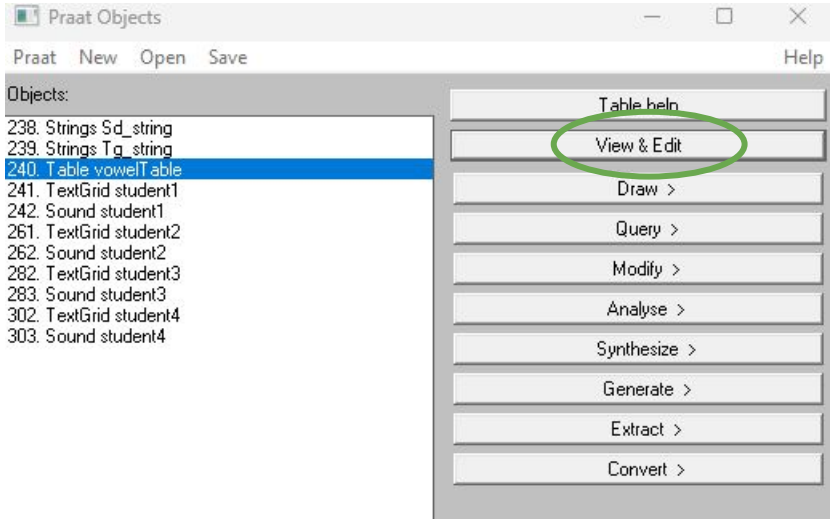
```
#################################################################

#Script to make a F1/F2 summary table

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

#------------READ ME------------#


#------------------------------#
```

It can include…

- what project this script was for
- what kind of data can/cannot be processed
- how the data should be prepared

This is helpful especially when you write an original script and if you plan to make it public; users can look at this section to learn how to use the script without reading the whole script.

STEP 7 : Let’s check the table



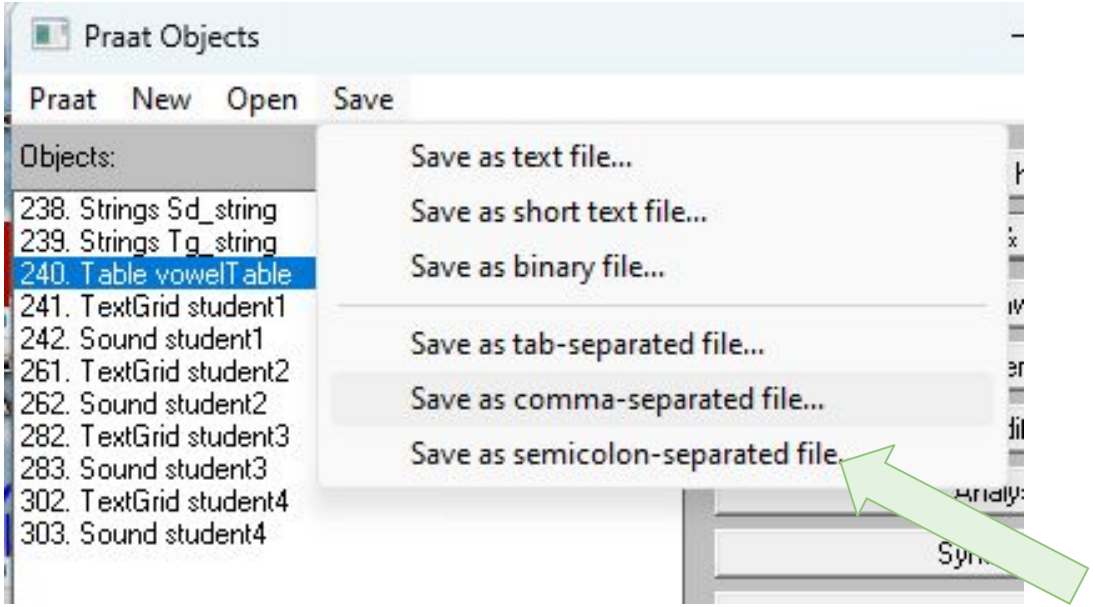
The final product!

132. Table vowelTable

File Edit

row	1 speaker	2 word	3 f1	4 f2
1	1	æ	700.9543385974703	1660.8486079622262
2	1	ɛ	584.441664072188	1936.3834410026514
3	1	i	270.3230718649986	2619.180027854013
4	1	ɪ	488.3684220848647	2091.8613286823083
5	1	ɑ	653.2454706356866	1114.3539487322898
6	1	ʊ	499.2535027409792	1587.9861196004438
7	1	ʌ	552.6011164255677	1592.3228055423895
8	1	u	326.1497048509948	1071.8483203236074
9	1	ɔ	598.3088983068092	969.9511099092432
10	1	æ	790.995012749639	1570.2810328934747
11	1	ɛ	588.6038000587411	1963.72731295921
12	1	i	303.7898193106315	2635.61844939361
13	1	ɪ	438.4737644819856	2080.704392763611
14	1	ɑ	675.1017428935488	1129.006813443664
15	1	ʊ	572.0937633446067	1594.3735266100396
16	1	ʌ	549.1162373895701	1599.848919365283
17	1	u	326.2707955287432	1416.948742796891
18	1	ɔ	795.0899405090258	1295.8836157845599
19	2	æ	744.8488102274576	877.2356485965348
20	2	ɛ	690.8324041145922	2154.7742290820106
21	2	i	371.3545253440421	2774.235792446776
22	2	ɪ	496.2856326379923	2178.980713269071
23	2	ʊ	764.9012144515085	1279.1003067181605
24	2	ʊ	543.8047046361087	1373.5796550035966
25	2	ʌ	694.0549170396871	1482.4638984639548
26	2	u	387.82886975402914	812.043578775809
27	2	ɔ	687.5380143922416	1104.253579498646
28	2	æ	765.7183528433184	1476.358591105421
29	2	ɛ	646.0462985714886	2064.8122875965696
30	2	i	361.5240576485962	2705.856072132221
31	2	ɪ	495.26696244777446	2108.326401615448
32	2	ɑ	788.1793094980189	1122.7642295684502
33	2	ʊ	429.8378535336295	1106.4376227603175
34	2	ʊ	578.6805136641018	1167.5501381103777
35	2	ʌ	709.2497842737794	1548.5160790094094

STEP 8 : Save the table



The Praat script we wrote today is here (clean ver.) and here (with full explanations in the script)

Extra: Let’s import this into R and plot!

Open your R Studio

Open a new script

The whole example script is here:

https://github.com/yukat237/Praat_Scripting_Lecture/blob/main/R%20script%20for%20creating%20a%20vowel%20plot

Or, search this: https://github.com/yukat237
and >> Praat_Scripting_Lecture >> R script for creating a vowel plot

Extra: Let's import this into R and plot!

4 lines in total!

Assign the file address (Replace the green part with your own file address)

```
tableFileAdress <- "C:\Users\Desktop\vowelTable.Table"
```

Import the table

```
vowelTable <- read.delim(tableFileAdress, sep = ",", fileEncoding = "UTF-16")
```

Take out if your .Table file is not encoded as "UTF-16. You can check it at the right bottom corner of the file when you open it

Rearrange the table

```
vowelTable2 <- as.data.frame(summarise(
        group_by(vowelTable, speaker, vowel), meanF1 = mean(f1), meanF2 = mean(f2), , .groups = 'drop'))
```

Plot a chart

```
ggplot(vowelTable2, aes(x=meanF2, y=meanF1, label=vowel, color=vowel))+ geom_text()+
  scale_x_reverse()+
  scale_y_reverse()
```

Inside the “vowelTable” variable:

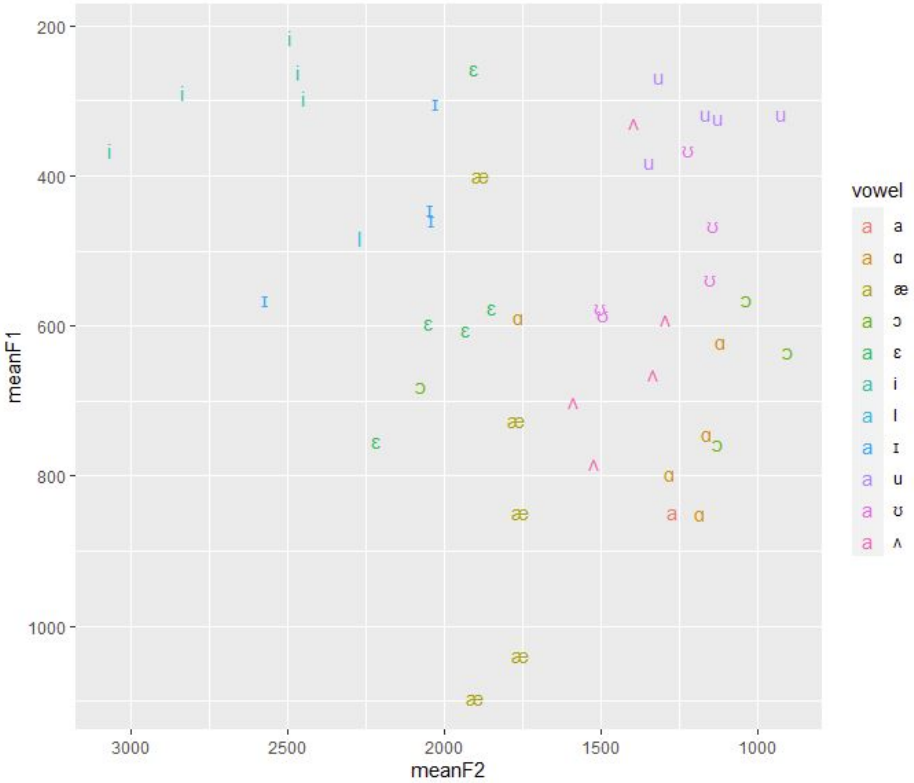
	speaker	word	f1	f2
1	1	æ	700.9543	1660.8486
2	1	ɛ	584.4417	1936.3834
3	1	i	270.3231	2619.1800
4	1	ɪ	488.3684	2091.8613
5	1	ɑ	653.2455	1114.3539
6	1	ʊ	499.2535	1587.9861
7	1	ʌ	552.6011	1592.3228
8	1	u	326.1497	1071.8483
9	1	ɔ	598.3089	969.9511
10	1	æ	790.9950	1570.2810
11	1	ɛ	588.6038	1963.7273
12	1	i	303.7898	2635.6184
13	1	ɪ	438.4738	2080.7044
14	1	ɑ	675.1017	1129.0068
15	1	ʊ	572.0938	1594.3735
16	1	ʌ	549.1162	1599.8489
17	1	u	326.2708	1416.9487
18	1	ɔ	795.0899	1295.8836
19	2	æ	744.8488	877.2356
20	2	ɛ	690.8324	2154.7742
21	2	i	371.3545	2774.2358
22	2	ɪ	496.2856	2178.9807
23	2	ʊ	764.9012	1279.1003
24	2	ʊ	543.8047	1373.5797
25	2	ʌ	694.0549	1482.4639

After grouping

Inside the “vowelTable2” variable:

	speaker	word	meanF1	meanF2
1	1	ɑ	664.1736	1121.6804
2	1	æ	745.9747	1615.5648
3	1	ɔ	696.6994	1132.9174
4	1	ɛ	586.5227	1950.0554
5	1	i	287.0564	2627.3992
6	1	ɪ	463.4211	2086.2829
7	1	u	326.2103	1244.3985
8	1	ʊ	535.6736	1591.1798
9	1	ʌ	550.8587	1596.0859
10	2	ɑ	788.1793	1122.7642
11	2	æ	755.2836	1176.7971
12	2	ɔ	690.7203	1103.6967
13	2	ɛ	668.4394	2109.7933
14	2	i	366.4393	2740.0459
15	2	ɪ	495.7763	2143.6536
16	2	u	374.0160	865.7692
17	2	ʊ	579.3061	1231.6669
18	2	ʌ	701.6524	1515.4900
19	3	ɑ	748.3510	1116.3927
20	3	æ	904.7669	1832.1419
21	3	ɔ	686.7271	1060.7545
22	3	ɛ	821.0294	1811.3599
23	3	i	402.6665	2811.2015
24	3	ɪ	507.2178	2012.5977
25	3	u	388.9245	1218.7737

The final product: a vowel plot based on class data!



Part 4

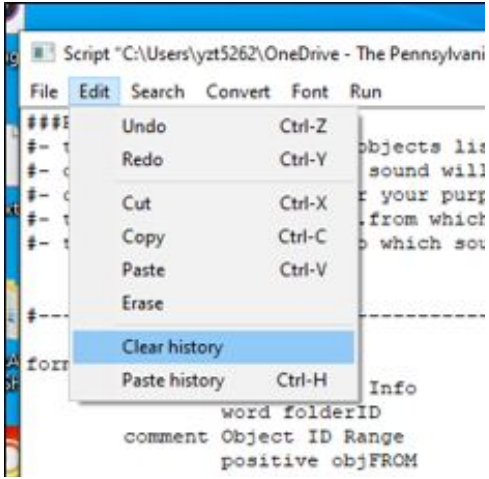
Advancing your Praat skills

How do I know which commands to use?

Method 1 – 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**



Method 2 – Google and Visit [official Praat manual]

Method 3 – Look at others' scripts and try reading it

Learn from others' scripts!

Add silence to the beginning of all sound files in a folder
This script adds a specified amount of silence to the beginning of every sound file in a folder. The resulting sound files are 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 every sound file in a folder. The resulting sound files are saved with their original names to a folder specified by the user. We have used this script to add silence to stimuli we were presenting in Qualtrics, since this software sometimes clips the end of sound files.

Get F1 to F4 at 7 times points for all labeled intervals
This script extracts measurements for F1, F2, F3, and F4 at 7 equidistant times points (25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%) in all labeled intervals. It processes all labeled intervals for all sound fil
text file. Each sound file and its corresponding TextGrid should have the same name.

Get duration and timepoints for all labeled intervals or points on specified tiers for al
This script logs the starting point, end point, and duration of labeled intervals in a specified
timepoint of labeled points in a specified point tier for all files in a folder. You can specify up
point tier. Each sound file and its corresponding TextGrid should have the same name.

Pull out words found in text file from one tier and put on additional tier
This script takes a list of words in a text file as input. This text file should have a header ro
irrelevant), and all the words in a single column. For all files in a folder, it then searches a
for those words, and copies them into a new interval tier at the bottom of the TextGrids at
appear in the tier above. This is useful if you want to pull out specific words (or any other
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 ti
- 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 fo any other corpus annotated by Praat.
- Equalizing amplitude (dB) This praat script adjusts the average amplitude (in dB) of all files in a folde
- 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 create multi-speaker noise.
- Change F0 This praat script raises/lowers the whole pitch contour by the specified factor for all the fi
- Adjust to nearest zero crossing This praat script adjusts the beginning and the end of all files to near 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 separate intervals)***

Measurements

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

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 testing scripts).

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 their **descriptions**).
  - KR's version improves on these mainly by combining them: if a grid exists, it opens it, otherwise it sets up a new
- **label-from-text-file** (Mietta Lennes, back up **here**)
  Replace interval labels in selected TextGrid with labeled text from a file (ML's **description** and a **streamlined vers**
- **open-multiple-textgrids** (John Tondering)
  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 **description**); cf. **word-chomper**
- **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 (ML's **description**).

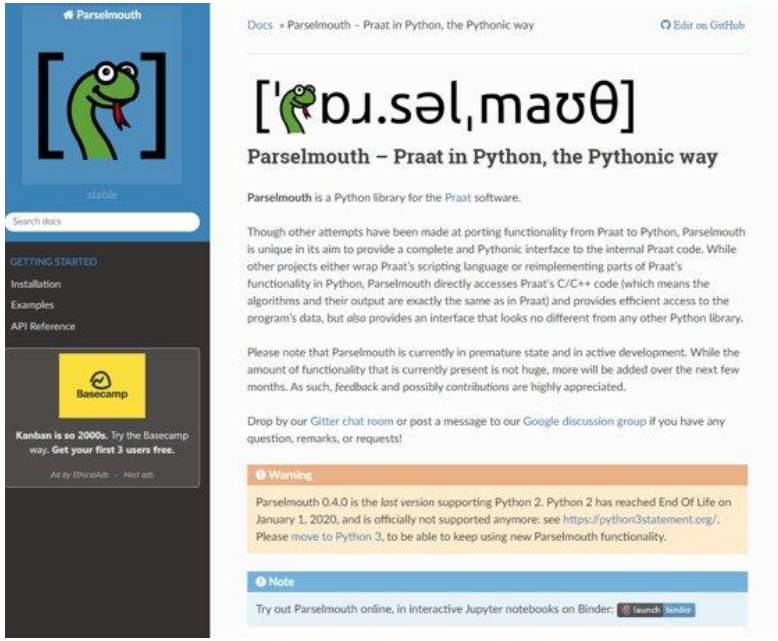
Analysis of sounds using text grids

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

You can use Python to use Praat

Not this:

But this: [Parselmouth]



Plugins to make your GUI more useful

Vocal Toolkit

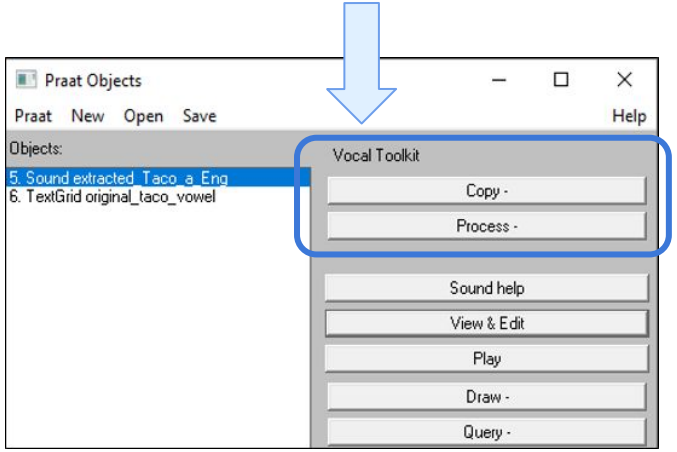
plugin with automated scripts for voice processing

Easy Steps!

1) Download a folder from here

2) Praat > Open Praat Script > "INSTALL.praat" in the folder > Run

3) Restart



[θæŋkju]!!