

Purpose for doing this research: Final stop sound in Min was dropped, affected by Mandarin

Research Q: Do Min speakers have the ability to discriminate final stop sound?

Prediction: By comparing the f2 of each vowel that preceded the stop sound, it will form a transition size (a gap)

Vowel under investigation: /a/ and /i/

Stop sounds that follow /a/ and /i/: /p/ /t/ /k/

Sound recording and processing - praat

1. By using a software called praat to record the actual sound with /a/ and /i/ vowel preceding /p/, /t/, /k/ stop sound respectively.

(1) Participant: speaker_1: male

speaker_2: female

(2) Stimuli: /a/ with /p/ ending - 6 words

/a/ with /t/ ending - 6 words

/a/ with /k/ ending - 6 words

/i/ with /p/ ending - 6 words

/i/ with /t/ ending - 6 words

/i/ with /k/ ending - 6 words

total: 36 words

2. Mark the duration of each vowel in the sound file

(1) Create a text file with a word tier that lines up with the sound file

(2) Mark the duration of each vowel in each word and mark the gap between two sounds (which we don't need) with "xxx" in the word tier

3. Run a praat script to get the data we need for each vowel

(1) Run a script called nasality automeasure, this script can give us the information we need from each individual vowel

Information required...

- f1

- f2

(2) When running this script, ask the code to cut each vowel into ten intervals, by doing this, we can see the changing of f1 and f2 at each timepoint.

(3) After running the script, it will generate a lot of data, but we only need some of them, so next, R will be used for data processing

Data processing - R

1. Import the data that we need
2. Subset the columns that we need from the data, which includes,
"filename", "word", "vowel", "freq_f1", "freq_f2", "vwl_duration", "timepoint",
"point_time", "point_vwlpct"
3. Sort out "xxx" and word
 - (1) The data is mixed with word and "xxx", but word is what we need, so by using sort function, we sort all the word in order so that all the "xxx" will be moved to the end of the data
 - (2) Next step is to exclude "xxx", to do so, we name an empty column called "formantID" and put number from 1:1460 in it.
1:720 represents word and 720:1460 represents "xxx".
Since we need is the data from 1:720, we subset the data from 1:720 base on formantID
4. What we want to analyze is the final stop sound that follows behind each vowel, so we need a column for vowel and another for final stop sound (coda)
 - (1) use substr to get vowel and coda from the column "vowel"
 - (2) first, extract coda from the column "vowel" by using substr function and assign it to coda
 - (3) create a new column called "coda" and put the data that we extract, which is coda, to the new column "coda"
 - (4) second, extract vowel from the column "vowel" by using substr function and assign it to vowel
 - (5) create a new column called "Vowel" and put the data we extract, which is vowel, to the new column "Vowel"
 - (6) since we've created column "coda" and "Vowel", the column "vowel" is no longer needed, so we excluded it by using subset function
5. We have to analyze f1 (freq_f1) and f2 (freq_f2) seperately, so we create a data frame with f1 and another data frame with f2 then rbind these two data frames together
 - (1) add a new column called "formant" and assign "f1" into that column
 - (2) add a new column called "frequency" and assign the data in "freq_f1" into the column "frequency"
 - (3) by using select function to delete "freq_f1" and "freq_f2" then assign this new data as "formant.data.f1", now we have a data frame with f1 only
 - (4) use the data frame that we haven't deleted "freq_f1" and "freq_f2", which is "formant.data", assign "f2" to the column "formant"

- (5) assign the data in "freq_f2" into the column "frequency"
- (6) by using select function to delete "freq_f1" and "freq_f2" then assign this data as "formant.data.f2"
- (7) rbind "formant.data.f1" and "formant.data.f2", then assign this new data as "formant.new"

Data visualizing - R (library required - tidyverse and Hmisc)

1. Plot a line bar with error bars (see whether vowel is influenced by the beginning of the word "onset")

- (1) ggplot
 - x axis represents timepoint (which is the ten intervals)
 - y axis represents frequency
 - colour represents the coda (we have /p/,/t/,/k/ coda)
 - lty for f1 and f2 with two different types of line
 - assign this ggplot as formant.plot
- (2) use function stat_summary to calculate the mean f1 and f2 of each interval
 - get the point of each interval
 - line all the points up (to have a better look on f1 and f2 changing)
 - add errorbars to see whether the vowel is influenced by onset
- (3) add the legend
 - ylab for "frequency"
 - xlab for "Time"
 - back ground as white
 - plot the graph with different filename and Vowel

2. Plot the line bar with error bars separately (hard to see the error bars when merging together)

- (1) use ggplot "formant.plot"
- (2) use function stat_summary to calculate the mean f1 and f2 of each interval
 - get the point of each interval
 - line all the points up (to have a better look on f1 and f2 changing)
 - add errorbars to see whether the vowel is influenced by onset
- (3) add the legend
 - ylab for "frequency"
 - xlab for "Time"
 - back ground as white
 - plot the graph with different filename and word

(by doing so, we can see the error bars in each vowel with different

coda)

3. Plot a line bar without error bars (too see the changing of f1 and f2)

(1) ggplot

- x axis represents timepoint (which is the ten intervals)
- y axis represents frequency
- colour represents the coda (we have /p/,/t/,/k/ coda)
- lty for f1 and f2 with two different types of line

(2) add geom_smooth to make the line smoother (for better vision)

(3) add the legend

- ylab for "frequency"
- xlab for "Time"
- back ground as white
- plot the graph with different filename and Vowel
- assign it to "formant.plot2"