Creating a Personalized Vowel Chart with PRAAT and R

Martin Schweinberger

June 28, 2016

Introduction

This post[1] exemplifies how to create a personalized vowel chart with PRAAT and R.

When learning or studying a language – the case in point here being English – it is likely that you are confronted with different classes of sounds, e.g. consonants and vowels. Consonants differ from vowels in that they are formed with an obstruction of the air stream coming from the lungs and they cannot form the nucleus of a syllable. In fact, consonants are classified according to the manner and place of the obstruction of the air stream. As vowels are produced without obstruction of the air stream, other criteria for differentiating between vowel sounds are needed. The criteria for differentiating between different vowel sounds are (i) the number of tongue positions during vowel production (to differentiate between mono-, diph-, and triphthongs), (ii) the height of the tongue, (iii) the position of the tongue, (iv) the roundedness of the lips. The latter two features are used in the production of vowel charts which show where in the mouth the tongue is located during the production of monophthongal vowel phones. A vowel chart for the monophthongal vowel phones in Received Pronunciation (RP) is shown in Figure 1.

[1]Please cite as:
Schweinberger, Martin. 2016. *Creating a personalized Vowel Chart with PRAAT and R.* `http://www.martinschweinberger.de/blog/persvowelchart/`, date.

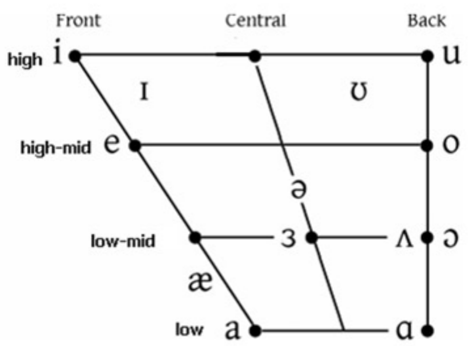

Figure 1: Vowel Chart of monophthongal vowel sounds in Received Pronunciation (RP)

Interestingly, a very similar figure can be created by plotting the Hertz frequency of the first formant of monophthongal vowel sound against the Hertz frequency of second formant minus the Hertz frequency of the first formant of a monophthongal vowel sound. Formants are frequencies of air waves that – if collapsed – form a complex vowel sound. In other words, vowels are periodic, i.e. rythmic, compressions and decompressions of air and to create a vowel sound, i.e. a complex periodic wave, one needs to produce several simple periodic waves simultaneously. During acoustic analysis, the complex wave is deconstructed into its component parts, i.e. the simple periodic waves that make up that sound. This means that we do not necessarily have to plot the position of the tongue of a speaker when he or she produces vowels to create a vowel chart but that analyses of audio recordings of words in which vowels occur, can be utilized to plot a personalized vowel chart of a speaker.

To produce a personalized vowel chart, the following steps are necessary:

1. Install Praat
2. Record words in which all monophthongal vowel sounds of a given variety occur;
3. Measure and extract the first and second formant of each vowel;
4. Visualize the vowel sounds.

The subsequent sections elaborate the above steps. However, before continuing a word of warning is in order. The example focuses on extracting and plotting vowel formants in an easy but also very uncontrolled way. In case vowel formant extraction is part of a proper research project, some additional steps are warranted. For instance, in a "serious" research project, it were necessary to control and reduce environmental noise and to optimize the recording situation, one would have to randomize the test items (words with the required phonetic environment and the respective vowel sounds) and use filler items (words that are not relevant for the analysis proper) in order to avoid participants guessing which items are relevant for the analysis, one would also use text grids in PRAAT to guarantee replicability instead of the simple measurements we use in the example here, etc. However, in case you are only interested in an approximation of your own vowel production and how native-like it is, the example fulfills its purpose and provides the reader with a step-by-step guide on how to plot your personalized vowel chart.

Downloading and installing PRAAT

The first step is thus to download PRAAT form `www.praat.org` and to install it on your machine by following the instructions provided on the website and by the PRAAT installation script. PRAAT is an open–source software for acoustic analysis that was developed by Paul Boersma at the University of Amsterdam.

After having installed PRAAT we need to record the words in which the monophthongal vowel phones occur. In this example, we will simply record the words shown in Table .

vowel	environment	word
/æ/	[*h*__*d*]	had
/ɑ/	[*h*__*d*]	hard
/e/	[*h*__*d*]	head
/iː/	[*h*__*d*]	heed
/ɜ/	[*h*__*d*]	herd
/ɪ/	[*h*__*d*]	hid
/ɔ/	[*h*__*d*]	hoard
/ɒ/	[*h*__*d*]	hod
/ʊ/	[*h*__*d*]	hood
/ʌ/	[*h*__*d*]	hud
/uː/	[*h*__*d*]	who'd

Table 1: Word selection with controlled environments in which monophthongal vowels occur.

The following section describes how to record data in PRAAT.

Recording words in PRAAT

To record these words, start PRAAT with a double click on the PRAAT-symbol which – after intstallation – appears on your Desktop. Two windows will appear: the main "object" window to the left and the picture window to the right (cf. Figure 2). Close the picture window on the right and choose `New` from the menu at the top of the main object window and select `Record mono sound` from the menu which pops up. For the recording it is, of course, necessary that a microphone is hook up to your machine – the better the microphone, the better the recording and thus the more accurate the graphical display we are going to produce.

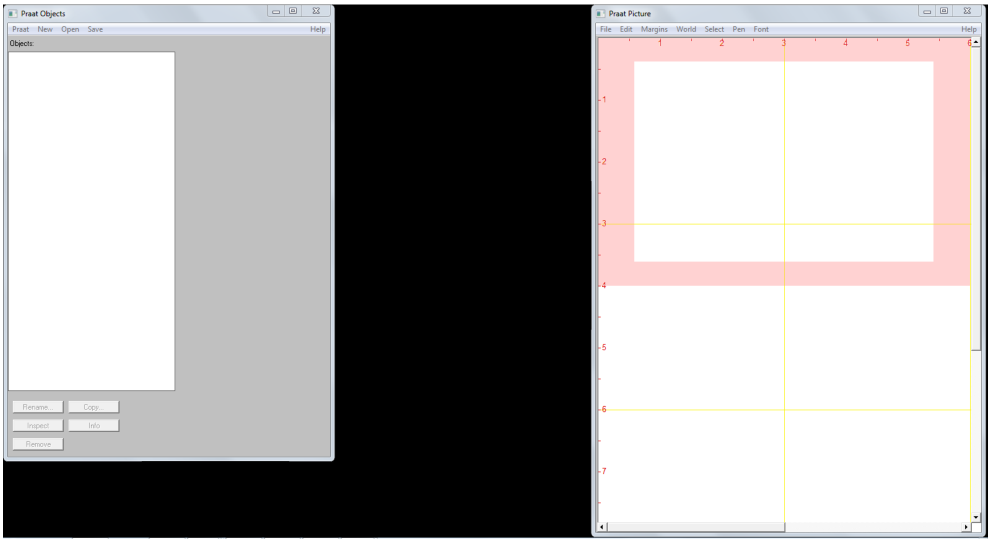

Figure 2: PRAAT's main object window (left) and PRAAT's picture window (right).

Selecting `Record mono sound` opens PRAAT's "SoundRecorder" window (cf. Figure 4). Select `Record`, label the recording by entering a title, e.g. `vowels`, in the `Name` field and read the words form the list shown in Table .

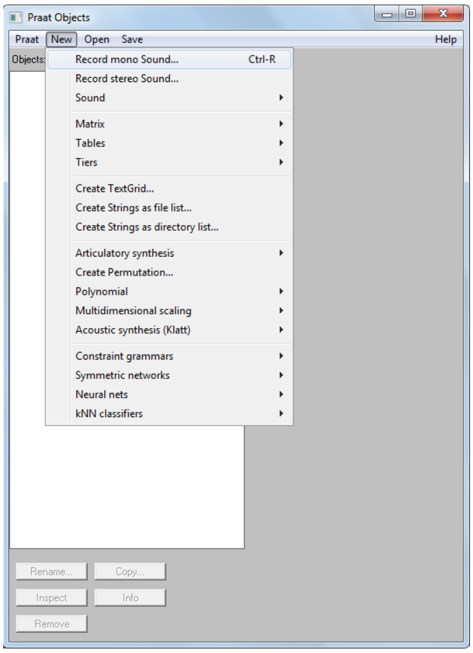

Figure 3: PRAAT's main object window.

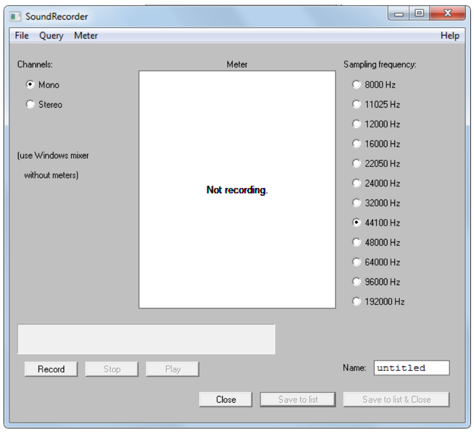

Figure 4: PRAAT's recording window.

Each word should be repeated at least three times with a short break between the individual items so that what you record is actually "had, had, had <pause> hard, hard, hard, etc.". Try to sound natural, i.e. avoid speaking too fast or too slow, and try not to sound artificial or too careful. While recording, there should be some green bouncing up and down in the vertical white "Meter" stripe (no bouncing indicates that your machine is not recording properly from the microphone).

Once you are finished with your recording, select `Stop` and next select `Save to list & close` (cf. Figure 8).

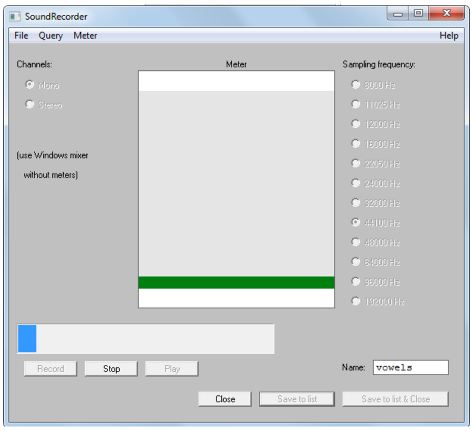

Figure 5: PRAAT's recording window during recording.

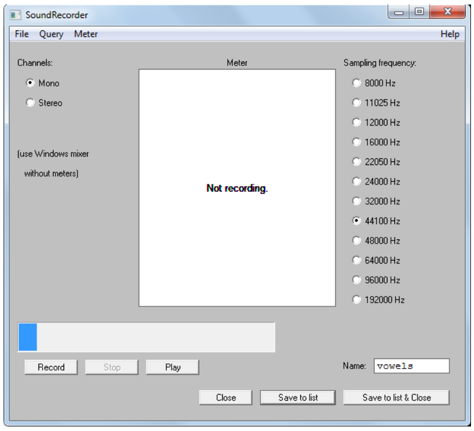

Figure 6: PRAAT's recording window after recording.

Saving has created an object in PRAAT's main object window – in case you have named your recoding `vowels`, the new object will be called `1. Sound vowels` (cf. Figure 7). Before editing the data, it is advisable to save them on your machine. To save the data select the `Save` option from the upper menu, then select `Save as WAV file...` and navigate to the directory in which you want to save the recorded data.

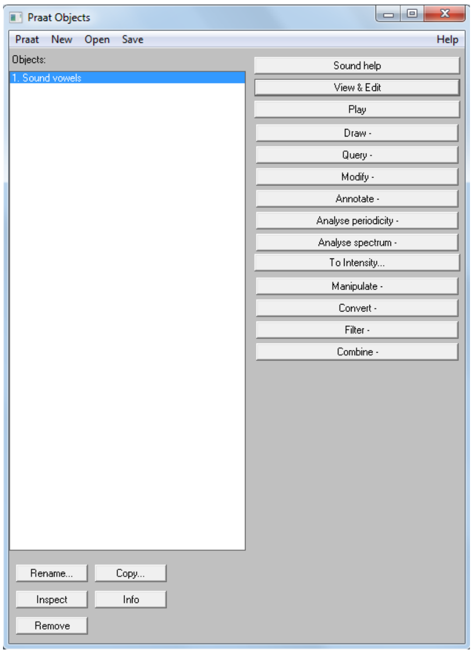

Figure 7: PRAAT's main object window with saved object.

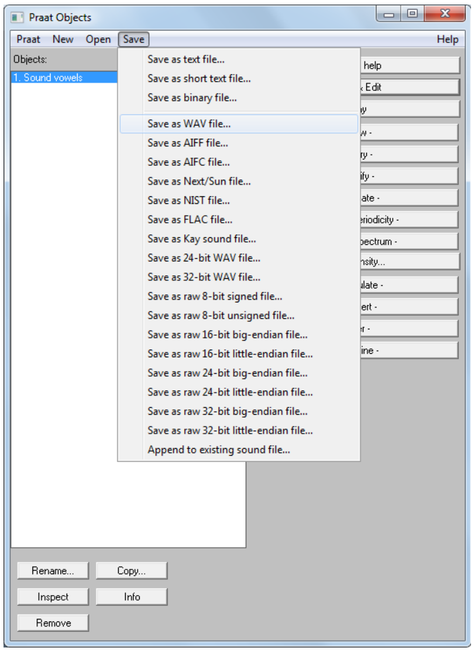

Figure 8: Save the recording as a .wav file.

Next, select `View & Edit` in PRAAT's main menu in the main object window. This will open PRAAT's edit window (cf. Figure 9) – the object represents a recording of the word "heed" repeated three times for sake of simplicity.

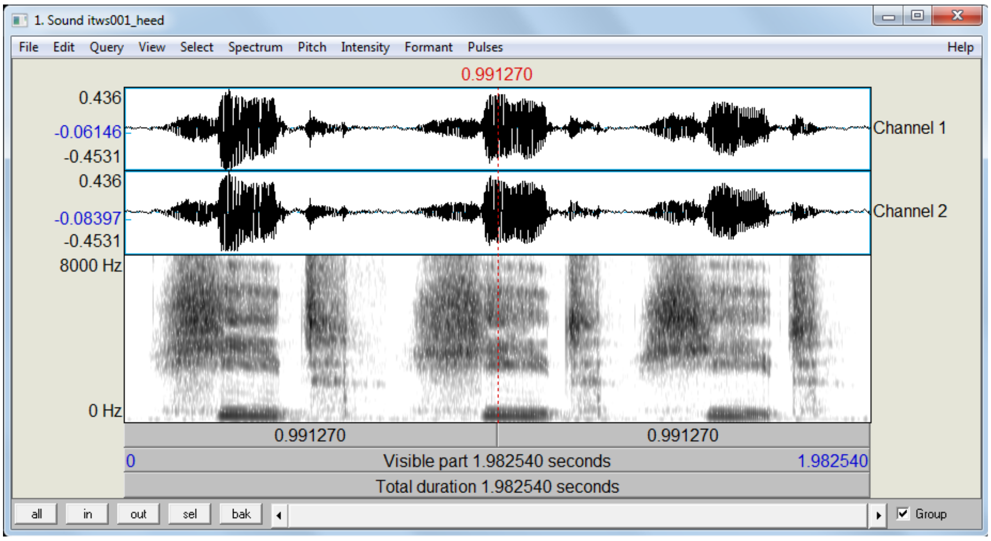

Figure 9: PRAAT's edit window with the word "heed" repeated three times.

After recording and saving the data necessary for the task at hand, we continue by extracting the vowel formants.

Measure and extract vowel formants

Before extracting of the vowel formants, some parameters need adjusting. In a first step, go to `Formant` from the menu at the top of the edit window and select `Formant settings...`. Next, select the option `Show formant` and then, depending on whether the recording represents a male, a female or a child, adjust the `Maximum formant (Hz)` to 5000 Hz (male), 5500 Hz (female) or up to 8000 Hz (for a child) (cf. `http://www.haskins.yale.edu/staff/gafos_downloads/AcouToyPraat(1).pdf`). It may also be necessary to adjust the number of formants that PRAAT aims to find: the default is 5, but it may be set to any number between 3 and 7 depending on the data. To elaborate, if the formants do not exhibit a regular horizontal pattern but they are somewhat unsteady or the dots are all over the place, try to find the number of formants that provide the best results (i.e. steady horizontal lines).

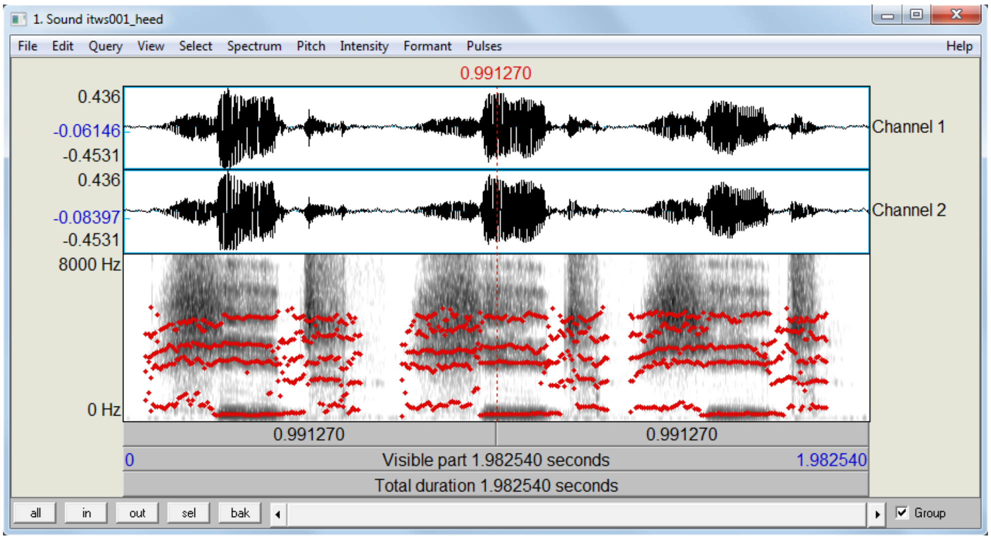

Figure 10: Praat's edit window with the word "heed" repeated three times and formants shown.

After having set the parameters, listen to the recording and highlight the section which represents the vowel sound you want to extract the formants from. Highlightling is done by selecting the start and end point of the vowel sound – the beginning and end of the steady line during which the vowel is produced – within the edit window as done for the first of the three instances of "heed" in Figure 11.

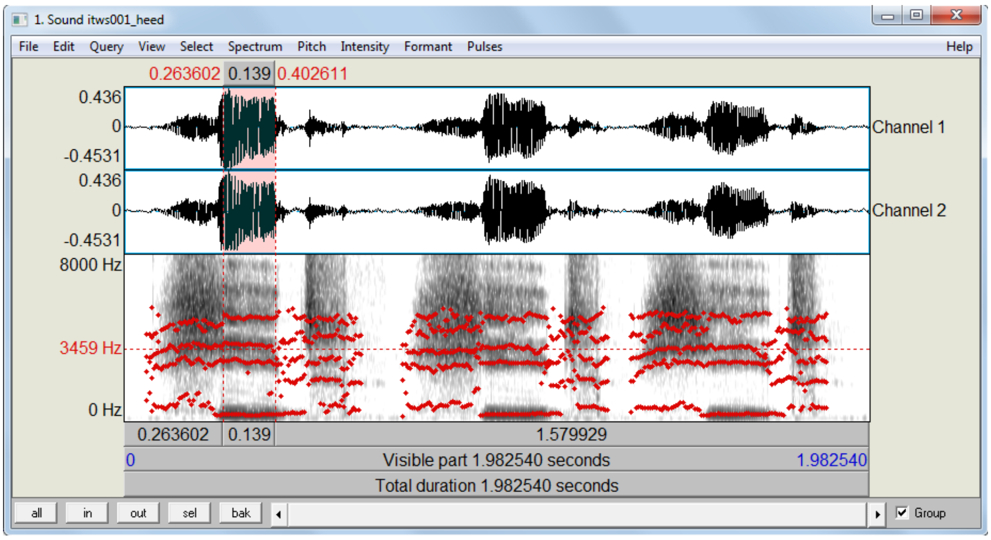

Figure 11: Praat's edit window with the word "heed" repeated three times and formants shown.

The vowel formants can be extracted by going to `Formant` in the edit window and selecting `Get first formant`. Having done so, a window with the mean Hertz frequency of the first formant during the steady state is shown (cf. Figure 12). Please note that you should additionally extract the start and end time of the highlighted section from the display in the edit window.

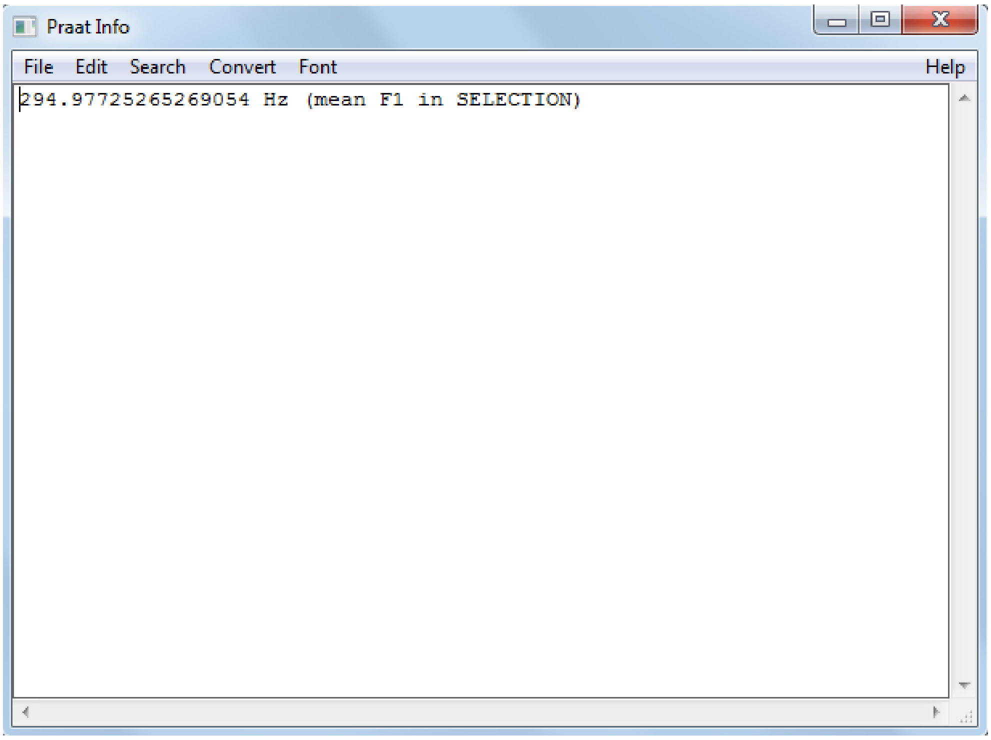

Figure 12: The mean Hertz frequency of first formant of the word “heed” during the steady state.

To extract the second (and in case you want to use your data in other analysis also the third formant) simply choose `Get third formant` (and `Get second formant`), note down the Hertz frequencies in a table, and also note down the start and end time of the steady state.

The final table should look like Table below (some columns are removed for sake of simplicity).

subject	file	item	trial	F1	F2
ms	vowels	had	1	717.33607	1868.175427
ms	vowels	had	2	743.483462	1903.715225
ms	vowels	had	3	720.973991	1938.69279
ms	vowels	hard	1	734.527473	1493.328918
ms	vowels	hard	2	832.922804	1407.824706
ms	vowels	hard	3	797.284207	1498.206365
ms	vowels	head	1	610.894333	2062.882002
ms	vowels	head	2	722.251868	2130.632179
ms	vowels	head	3	625.111709	2009.65073
ms	vowels	heed	1	263.382964	2833.001674
ms	vowels	heed	2	301.417588	2745.84707
ms	vowels	heed	3	286.96557	2822.598805
ms	vowels	herd	1	532.792463	1704.995395
ms	vowels	herd	2	537.796188	1819.891642
ms	vowels	herd	3	524.713714	1704.232122
ms	vowels	hid	1	451.876599	2390.799614
ms	vowels	hid	2	417.033045	2483.389953
ms	vowels	hid	3	410.681731	2360.038236
ms	vowels	hoard	1	540.330631	951.144344
ms	vowels	hoard	2	549.92048	927.095558
ms	vowels	hoard	3	648.048222	1093.346613
ms	vowels	hod	1	698.406882	1144.46685
ms	vowels	hod	2	615.162079	1086.447865
ms	vowels	hod	3	751.018999	1452.466334
ms	vowels	hood	1	431.299339	1478.192998
ms	vowels	hood	2	404.18844	1453.1036
ms	vowels	hood	3	470.146946	1216.302702
ms	vowels	hud	1	646.051365	1700.003028
ms	vowels	hud	2	622.53022	1510.451378
ms	vowels	hud	3	749.353953	1581.757811
ms	vowels	whod	1	346.88118	1013.000682
ms	vowels	whod	2	353.826456	1285.834128
ms	vowels	whod	3	366.813688	1016.979961

Table 2: Vowel formants extracted from PRAAT

The next section describes how to plot the data and compare the vowels to equivalent vowels produced by native-RP speakers.

Visualizing the vowel sounds

After extracting the first and second formants, they will plotted and equivalent vowels produced by native RP speakers are added.

In a first step, the script is initialized by a short description of what our script does and defining the path names.

```
# title: creating a personalized vowel chart with R
# author: martin schweinberger
# date: 2016-06-27
# description: plotting a customized vowel chart with R
# the input data must be a data frame of the format:
# subject (Speaker), file, item, F1_Hz, and F2_Hz.
# the column label of x must at least contain:
# the reference data must be a data frame of the format:
# subject, file, item, F1_Hz, F2_Hz.
##########################################################
# remove all lists from the current workspace
rm(list=ls(all=T))
# set path to data
vowelpath <- "http://www.martinschweinberger.de/docs/data/
    vowels.txt"
# set path to RP data
refpath <- "http://www.martinschweinberger.de/docs/data/
    rpvowels.txt"
```

In a next step we define a function that creates transparent symbols or forms which represent the vowel space in our plot. The code is not self-written but reproduced from `http://stackoverflow.com/questions/12995683/any-way-to-make-plot-points-in-scatterplot-more-transparent-in-r`. This function adds transparency to a color. Define transparency with an integer between 0 and 255 0 being fully transparent and 255 being fully visible works with either `color` and `trans` a vector of equal length, or one of the two of length 1.

```
# function for transparent symbols
addTrans <- function(color,trans)
{
if (length(color)!=length(trans)&!any(c(length(color),length(
    trans))==1)) stop("Vector lengths not correct")
if (length(color)==1 & length(trans)>1) color <- rep(color,
    length(trans))
if (length(trans)==1 & length(color)>1) trans <- rep(trans,
    length(color))
num2hex <- function(x)
{
```

```
hex <- unlist(strsplit("0123456789ABCDEF",split=""))
return(paste(hex[(x-x%%16)/16+1],hex[x%%16+1],sep=""))
}
rgb <- rbind(col2rgb(color),trans)
res <- paste("#",apply(apply(rgb,2,num2hex),2,paste,collapse=""),sep="")
return(res)
}
```

As the necessary preparations are now done, we can now start to set up the plot by loading the data.

```
# load data
v <- read.table(vowelpath, header = T, sep = "\t")
```

Once the data is available in R, it can be processed and the required information is extracted.

```
# convert into data frame
v <- as.data.frame(v)
# order data frame
v <- v[order(v$subject, v$item, v$trial), ]
# calculate mean for each vowel
F1 <- tapply(v$F1_Hz, v$item, mean)
F2 <- tapply(v$F2_Hz, v$item, mean)
# calculate sd for each vowel
F1sd <- tapply(v$F1_Hz, v$item, sd)
F2sd <- tapply(v$F2_Hz, v$item, sd)
# create a data frame from the values
v1 <- data.frame(rep("ms", length(F1)), names(F1), rep("wordlist", length(F1)), F1, F2, F1sd, F2sd)
# adapt column names
colnames(v1) <- c("subject", "item", "context", "F1", "F2", "F1sd", "F2sd")
```

Having prepared the data, we now start to plot the vowel data.

```
# define the axis values/labels for the plot
z1 = seq(-00, 3000, 500)
z2 = seq(250, 1050, 100)
# transform values
x = v1$F2 - v1$F1
y = v1$F1
# set up plot
```

```
symbols(x, y, circles = v1$F1sd, inches = 1/3, bg = addTrans("lightgrey", 100),
fg = NULL, xlim = rev(range(z1)), ylim = rev(range(z2)),
 xlab = "F1 (Hz)", ylab = "F2 - F1 (Hz)", add = F)
# add ipa symbols
ipa <- c("\u00E6", # had
"\u0251", # hard
"e", # head
"i", # heed
"\u025C", # herd
"\u026A ", # hid
"\u0254", # hoard
"\u0252", # hod
"\u028A", # hood
"\u028C", # hud
"u" # whod
)
box() # add box
grid() # add grid
# add symbols
text(x, y, ipa, cex = .8)
text(x, y, v1$item, pos = 1, cex = .8)
```

In a next step, the reference data is processed in order to add it to the plot.

```
# load reference data
ref <- read.table(refpath, header = T, sep = "\t")
# convert into data frame
ref <- as.data.frame(ref)
# order data frame
ref <- ref[order(ref$subject, ref$item, ref$trial), ]
# calculate mean for each refowel
F1 <- tapply(ref$F1, ref$item, mean)
F2 <- tapply(ref$F2, ref$item, mean)
# calculate sd for each refowel
F1sd <- tapply(ref$F1, ref$item, sd)
F2sd <- tapply(ref$F2, ref$item, sd)
# create a data frame from the ref values
ref1 <- data.frame(rep("rpspk", length(F1)), names(F1), rep("wordlist", length(F1)), F1, F2)
# change col names
colnames(ref1) <- c("subject", "item", "context", "F1", "F2")
# add sd to data frame
ref2 <- data.frame(ref1, F1sd, F2sd)
```

The reference data frame will now look as displayed in Table below.

subject	item	context	F1	F2	F1sd	F2sd
subject	item	context	F1	F2	F1sd	F2sd
rpspk	had	wordlist	916.35	1473.15	124.29815	119.43696
rpspk	hard	wordlist	604.15	1040.15	70.91973	40.06478
rpspk	head	wordlist	599.95	1925.70	102.22858	143.60476
rpspk	heed	wordlist	276.15	2337.60	25.48328	223.42440
rpspk	herd	wordlist	493.55	1372.40	47.40917	95.94648
rpspk	hid	wordlist	392.85	2174.35	40.83893	166.85868
rpspk	hoard	wordlist	391.65	629.60	39.70718	81.19074
rpspk	hod	wordlist	483.10	864.90	35.48002	48.49948
rpspk	hood	wordlist	412.85	1286.65	32.98209	193.69870
rpspk	hud	wordlist	658.20	1208.05	116.14945	72.51677
rpspk	whod	wordlist	288.70	1616.30	30.18905	225.73858

Table 3: Vowel formants of native RP speakers

In a next step, the reference data is added to the plot. The formants for the RP vowels is taken from Hawkins and Midgley (2005: 196) (this article can be accessed here: `http://journals.cambridge.org/download.php?file=%2F4190_EE11D5A504D77D9E1521391B92C6038D_journals__IPA_IPA35_02_S0025100305002124a.pdf&cover=Y&code=251b5f479e7fd20814e1b68b258da7cd`) and represents the first and second formant for the words *heed*, *hid*, *head*, *had*, *hard*, *hod*, *hoard*, *hood*, *who'd*, *hud*, and *herd* produced by 5 20 to 25 year old speakers.

```
# transform refalues
x = ref2$F2 - ref2$F1
y = ref2$F1
# set up plot
symbols(x, y, circles = ref2$F1sd, inches = 1/3, bg =
   addTrans("lightblue", 100),
fg = NULL, xlim = reref(range(z1)), ylim = reref(range(z2)),
   xlab = "",
ylab = "", add = T, main = "")
# add ipa symbols
ipa <- c("\u00E6", # had
"\u0251", # hard
"e", # head
"i", # heed
"\u025C", # herd
"\u026A ", # hid
"\u0254", # hoard
"\u0252", # hod
```

```
"\u028A", # hood
"\u028C", # hud
"u" # whod
)
# add symbols
text(x, y, ipa, cex = .8, col = "red")
text(x, y, ref2$refowel, pos = 1, cex = .8)
# add legend
legend("bottomleft", inset=.05, c("Participant", "Native (modern Received Pronunciation)"),
 fill=c("lightgrey", "lightblue"), horiz=F)
```

The resulting visualization is displayed below.

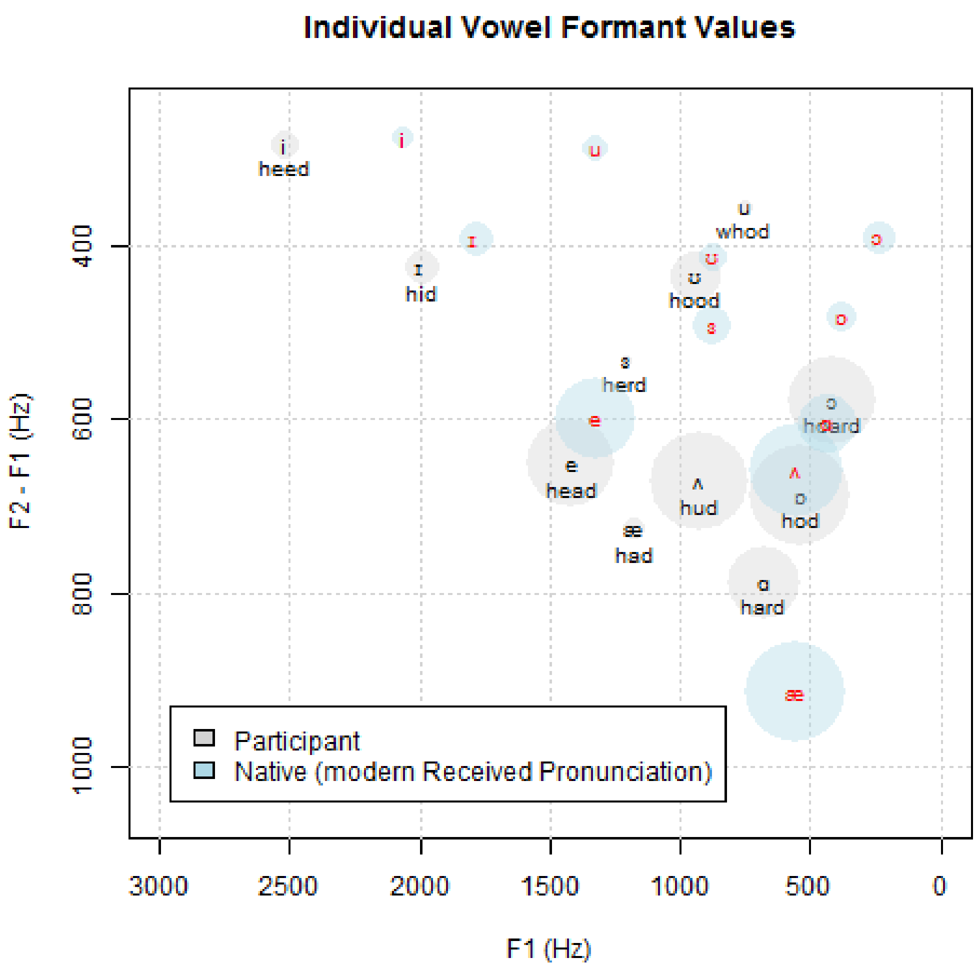

Figure 13: Personalized Vowel Chart with reference RP Vowels

Figure 13 depicts a personalized vowel chart of English vowels produced by a female L1–German speaker with vowel spaces of native RP speakers (blue circles with red symbols) for reference.

The vowel chart shows that the i-sounds by the L1-German speaker are more fronted and that the o-sounds are substantially higher by the non-native speaker compared to the RP reference vowel spaces. The short u-sound, however, is very similar, indicating that this L1-German speaker produces the short u-sound in English very native-like while the long u-sound is higher and more fronted in the speech of the L1-German speaker. Interestingly, the vowel space of the ash differs quite dramatically between the native speakers and the L1 German speaker which could be caused by the fact that German does not have an ash vowel.

I hope this short tutorial helps you in creating your own personalized vowel charts with PRAAT and R.

References

Hawkins, S. and J. Midgley (2005). Formant frequencies of rp monophthongs in four age groups of speakers. *Journal of the International Phonetic Association 35*(02), 183–199.