Building digital resources for research on under-resourced languages

Part II: Survey data and cross-validation

presented by Matt Wagers & Kie Zuraw, AIMM3 @ UMass

2 October, 2015

Contents

0.1 Preface

The source file for this handout is an R Markdown document, which is available here: http://people.ucsc.edu/~mwagers/aimm/word_frequency.Rmd (and there's a compiled PDF version). Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. R Markdown allows you to interweave chunks of code in the text of the document, using the `knitr` package. For more details on using R Markdown see http://rmarkdown.rstudio.com. In R Studio, when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

1 Introduction

In this section of tutorial, we will discuss:

- how to collect survey data that provides information about word and word-form frequency;
- how to do a basic descriptive analysis of that data;
- cross-validation of those data with the web-scraped corpus we compiled in part 1 of the tutorial;
- comparison to behavioral data.

To work through this tutorial, there are several data files you should download.

- word_ratings.csv
- corpus_counts.csv
- manu_mas_yamu.csv
- listening_times.csv

Or download them all, and the source for this document, as a compressed file. Place the (individual files) in a working directory.

```
setwd("~/Desktop/cha_frequency/") # put the path to your working directory here
```

NB: While this handout can be used somewhat passively if you are not familiar with R or not interested in working along, we will not pore over it or necessarily work through it sequentially; instead it will be consulted as resource.

1.1 Word frequency ratings

Data about word and word-form frequency are indispensable for doing experiments on real-time morphosyntactic processing. Typically, these values are estimated from very large, well-curated corpora which are balanced in terms of genre, register, modality, etc. However, for a small language, like Chamorro, it can be difficult to make these estimates. There are limited text resources written by native speakers and there is tremendous orthographic variability.

We will try to make these estimates, first off, by using survey data. In principle, this is a fine idea. The point of compiling word frequency data, and fancier measures, is to provide a baseline model of the contents of the lexical system in someone's head. Of course, it does not follow that those contents will necessarily be directly accessible in any particular kind of task. Fortunately, subjective frequency ratings have been shown to correlate with lexical-decision and naming latencies (Gordon 1985, Connine et al. 1990; sometimes, but not always, more strongly than frequencies obtained from a text corpus). But, unsurprisingly, there are limitations to this method and resource-sensitivities that can be exacerbated by characteristics of the language and speaker population. Here are some practical questions to ask:

- How many words can one participant rate?
- How are the task instructions formulated to elicit a frequency judgment?
- How does you know whether speakers are giving information about the surface form of a word, versus its root?
- What do you do when the surface form, out of context, has multiple syntactic categories?

In this part of the tutorial, we will describe how subjective frequency ratings can be collected via a survey, and how to do a (basic) analysis. Then, we will try to use those ratings in several ways: 1) to validate our web-scraped corpus measures; 2) to compare to other behavioral measures.

1.2 Language background

Chamorro is an Austronesian language spoken indigeneously in the Mariana Islands. The number of speakers is on the order of 10s of thousands (45,000 in the Marianas, and numerous speakers in the continental U.S.), but it is on the cusp of language endangerment. The Mariana Islands, a chain of islands in the Western Pacific, have been under foreign domination since the late seventeenth century. They are now divided into two political entities: the U.S. Commonwealth of the Northern Mariana Islands (the CNMI) and the unincorporated U.S. territory of Guam. Although Guam has a larger number of Chamorro speakers, the language is better maintained in the CNMI, where almost all Chamorros aged 55 or over are fluent speakers and there are many speakers in the 30-55 age range.

Chamorro is a head-initial language. The clause consists of a predicate, which can be any major category type, followed by arguments and adjuncts. Although the relative order of arguments and adjuncts is flexible, the neutral word order of clauses containing verbs is *Verb-Subject-Object*. Verbs are inflected to indicate subject agreement in person and number and to show tense, aspect and mood. As well, they participate in a productive system of voice and valence changing morphology (these include 2 passives, an antipassive, causative, applicative, and recriprocal). The richness and complexity of Chamorro morphology make it an excellent target for investigating morphophonological and morphosyntactic interactions, as well as understanding the role of morphology in real-time sentence processing.

There are no extensive corpora of Chamorro. Ann Cooreman created an archive of spoken stories in the early 1980s, which were transcribed by a native speaker and on which she based her 1987 study of discourse continunity. Scarlett Clothier-Goldschmidt compiled a small translation corpus of the New Testament (translated recently by contemporary speakers), on which she based her 2015 M.A. thesis on person-animacy effects. Currently underway is a large, community-directed revision to the 1975 Chamorro dictionary (Topping, Ogo & Dunca, 1975; revision edited by Dr. Liz Rechebei, Manuel F. Borja & Tita Hocog). Each entry contains at least 3 example sentences, which themselves form a substantive database for research. There are a handful of books in print, occasional newspaper editorials, and some actively maintained personal blogs. There are a number of translations in print and on the web which were not always made by native speakers (including some early religious texts and some US government forms).

A major challenge to investigating Chamorro via written text is orthographic variability. There are two official orthographies, one adopted in Guam in 1983 and the other adopted in the CNMI in 2010. To get a sense of the difference between the two official orthographies, consider the excerpts from the book *Estreyas Mariånas*:

- **CNMI Orthography**
  - Si Kanåriu mama'chechemchum gi ramas trongkun nunu…Si Kanåriu hinasson-ña na manlili'i' gui' birak…Tumekkun si Kanåriu ya ha oppi i kuestion Chungi'. Ilek-ña taiguini, "Kao un li'i' atyu guatu na tinekcha'?"
- **Guam Orthography**
  - Si Kanårio mama'chechemchom gi ramas tronkon nunu…Si Kanårio hinasso-ña na manlili'e' gue' birak…Tumekkon si Kanårio ya ha oppe i kuestion Chunge'. Ilek-ña taiguini, "Kao un li'e' ayo guatu na tinekcha'?"

From the CNMI orthography it is easier to recover the phonology, since each grapheme has a consistent pronunciation. From the Guam orthography, it is easier to recover the root word, since the spelling of all words derived from a particular root preserve one spelling, regardless of regular phonological changes.

However, both the official orthographies coexist with older spellings introduced by Spanish missionaries as well as with idiosyncratic individual styles. For example, the word meaning "good," may be spelled `måolik` (in both official orthographies; `å` is the spelling for the low back vowel), `maolik` (without the typographically more complicated a-ring or a-lonnat), `måolek`, `maolek` or `mauleg` (a very old spelling). Here's what we found in the corpus we created:

- `måolik`: 0 forms
- `maulek`: 8 forms
- `måolek`: 15 forms
- `maolek`: 182 forms
- `mauleg`: 1447 forms

Orthographic variability isn't just a stumbling block for the researcher, but it's also a challenge for literate speakers of Chamorro. For these reasons, working directly with speakers without the print intermediary is very attractive.

2 Chamorro word frequency survey

The survey reported in this section was carried out in Summer 2013 by Wagers, Sandra Chung (UC Santa Cruz) and Manuel F. Borja (Inetnun Åmut yan Kutturan Natibu, Saipan, CNMI). In a previous experiment using the self-paced listening technique, speakers listened to sentences phrase-by-phrase. Because it is not entirely clear how listening times in a particular phrase align with incremental processing, it would be useful to have some marker of when a particular lexeme was being processed - such as a dependency on frequency. To collect frequency estimates for the words in their experiment, they asked participants to consider a list of words and, for each word, answer the following question:

```
Kuåntu   biåhi  un u'usa    pat  un huhunguk  esti na palåbra siha?
how.many times  2P use.PROG or   2P hear.prog DEM  L  word    PL
```

There were five (ordered) response categories, anchored to the following Chamorro phrases:

1. **todu i tiempu**: all the time
2. **sessu**: often
3. **guaha na biåhi**: sometimes (*lit.* "there are times")
4. **håssan**: rarely
5. **ni ngai'an**: never

Instructions, examples and all test items were presented auditorily in person (usually in group settings). Participants completed 4 practice trials to make sure they understood the instructions. Each survey contained 19 words in addition to the practice trials, and two versions of the survey were produced so that ratings were collected for 38 words overall.

An example survey can be viewed here: http://people.ucsc.edu/mwagers/aimm/chamorro.word_survey.pdf.

The words tested are as follows:

2.1 Word list

Word	*Translation*	**Word**	*Translation*
aflitu	fry	kihåyi	tattle; tell on
aguaguat	naughty	konni'	catch; bring/take
akusa	accuse	kula	sift; examine
apåsi	pay	kumbida	invite
ayuda	help	lalåtdi	scold
batnis	varnish	lampåsu	mop
bisita	visit	lasgui	sharpen

Word	*Translation*	**Word**	*Translation*
chiku	kiss	lehgua’	stir
chuda’	pour	mapagåhis	cloud
dingding	ring (a bell)	na’bubu	anger (s.o.)
dispidi	bid farewell	na’chålik	make laugh
diriti	dissolve	na’homlu’	heal
fa’bola	pretend is a ball	patgun	child
gimin	drink	se’si’	cut with a knife
guå’ding	trip; cause to stumble	saibuk	boil starchy food (in coconut milk)
huchum	close (e.g., a door)	tanum	plant (something)
kanta	sing	titik	tear; rip
kassi	tease	toktuk	hug
katdu	make into a soup (tr.)		

*NB: The apostrophe represents the glottal stop; `å` is the low back vowel; and `y` is the voiced affricate.

Notice that most words belong to the syntactic category V, though not exclusively. No words were inflected, although not all words were monomorphemic (as in *fa’bola* = *fa’* “make/pretend” + *bola* “ball”“; or any word containing the causative prefix *na’*).

2.2 Survey data: first-pass analysis

First, we load the data. The `csv` file should be in the path of the script.

```r
ratings.df <- read.csv("word_ratings.csv")

# Alternatively, read these files directly from the web with RCurl library
# Uncomment the following 3 lines:

# library(RCurl)
# webfile <- getURI("http://people.ucsc.edu/~mwagers/aimm/word_ratings.csv")
# ratings.df <- read.csv(text = webfile)
```

This data file contains 8 fields: `participant`, a unique ID for each survey-taker; `list`, the specific survey they received (not every person rated every word); `word`, the word rated; `rating`, on a scale from 1 to 5; `sex` and `age`, basic demographic data for each participant. Finally `subj.cluster` classifies participants according to how uniformly they used the rating scale (see below). Below is a sample of 10 records from the file. (Note the two values for the field `sex` are *palaoan* “female” and *lahi* “male”).

```r
n.records <- nrow(ratings.df)
ratings.df[sample(n.records, 10), ]
```

```
##       X participant list    word rating     sex age subj.cluster
## 902 927        S033    2 dispidi      3 palaoan  62            2
## 586 611        S015    2  aflitu      1 palaoan  47            1
## 831 856        S025    2 guading      3    lahi  62            1
## 708 733        S020    2   chiku      3 palaoan  61            2
## 844 869        S026    2    sesi      1    lahi  29            2
## 225 235        S010    1   titik      2 palaoan  81            2
## 345 360        S027    1  saibuk      3 palaoan  40            2
## 388 405        S029    1   katdu      3 palaoan  56            2
## 293 306        S013    1   tanum      1 palaoan  64             1
## 147 153        S007    1   chuda      2 palaoan  42             2
```

Already we've done something non-trivial: encoded the ratings as equally-spaced integers, assigning *todu i tiempu* the value 1 and *ni ngai'an* the value 5. As long as we remember these numbers correspond to ranked categories and that the participants didn't *actually* give us numbers, not much harm will be done (in the appendix, a more satisfying treatment will be given of this issue.)

Let's summarize the ratings by counting, per word, how many times each rating category was used:

```
# Create a table of _word_ x _rating_
rating_counts.tab <- with(ratings.df, table(word, rating))

# And then let's convert these counts to frequencies
# First, find the total number of ratings per word
# ... this should be about the same for each word, but may not be
sum_ratings <- apply(rating_counts.tab, 1, sum)
# ... then divide the counts by this sum
rating_freq.tab <- (rating_counts.tab / sum_ratings)

# There's a simpler way to do this:
# prop.table(ratings_count.tab, 1)
```

Let's double check the data structures we've created by picking a few words at random to inspect.

```
n.words <- nrow(rating_counts.tab)
selected_words <- sample(1:n.words, 5)  # pick 5 words

print(rating_counts.tab[selected_words,]) # display the raw counts
```

```
##           rating
## word        1 2  3  4 5
##   fabola    2 2  6 14 1
##   konni    12 3  2  0 0
##   aguaguat 26 8  7  1 0
##   sesi      6 7 12 16 1
##   batnis    2 3  8 21 8
```

```
print(rating_freq.tab[selected_words,], digits=2) # and frequencies
```

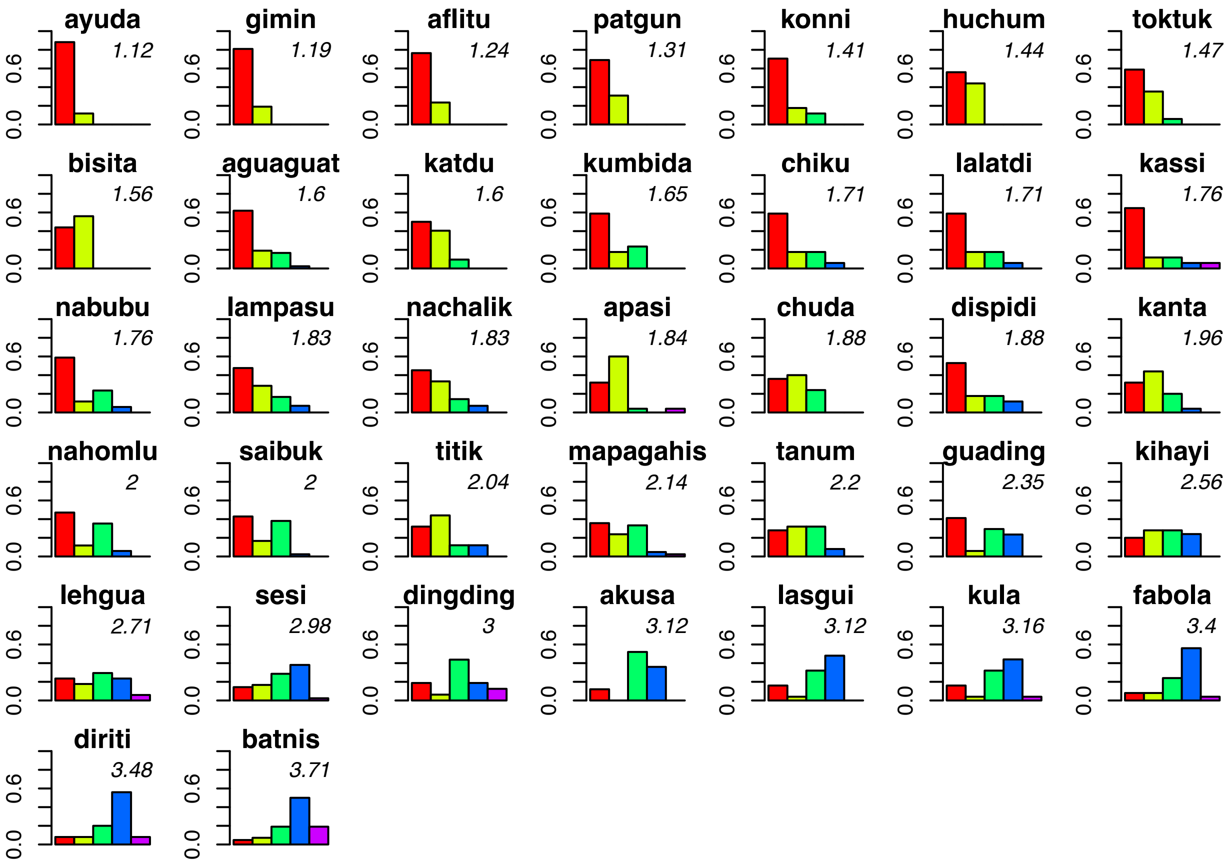
```
##           rating
## word           1     2    3     4     5
##   fabola   0.080 0.080 0.24 0.560 0.040
##   konni    0.706 0.176 0.12 0.000 0.000
##   aguaguat 0.619 0.190 0.17 0.024 0.000
##   sesi     0.143 0.167 0.29 0.381 0.024
##   batnis   0.048 0.071 0.19 0.500 0.190
```

Then, we'll create a visualization. The first thing to do is to calculate each word's average rating.

```
# Create the averages
rating_avg <- with(ratings.df,
                       tapply(rating, word, mean, na.rm=TRUE))

# ... then create a list of integers ordered by those averages
order(rating_avg) -> rating_bymean.ix

# .... put the list of words in that order by extracting them with
# the levels() command. We will use this string vector to order any plots we make.
words_ordered.ix <- levels(ratings.df$word)[rating_bymean.ix]
```

Let's now create a histogram that shows the frequency with which each rating category was used.

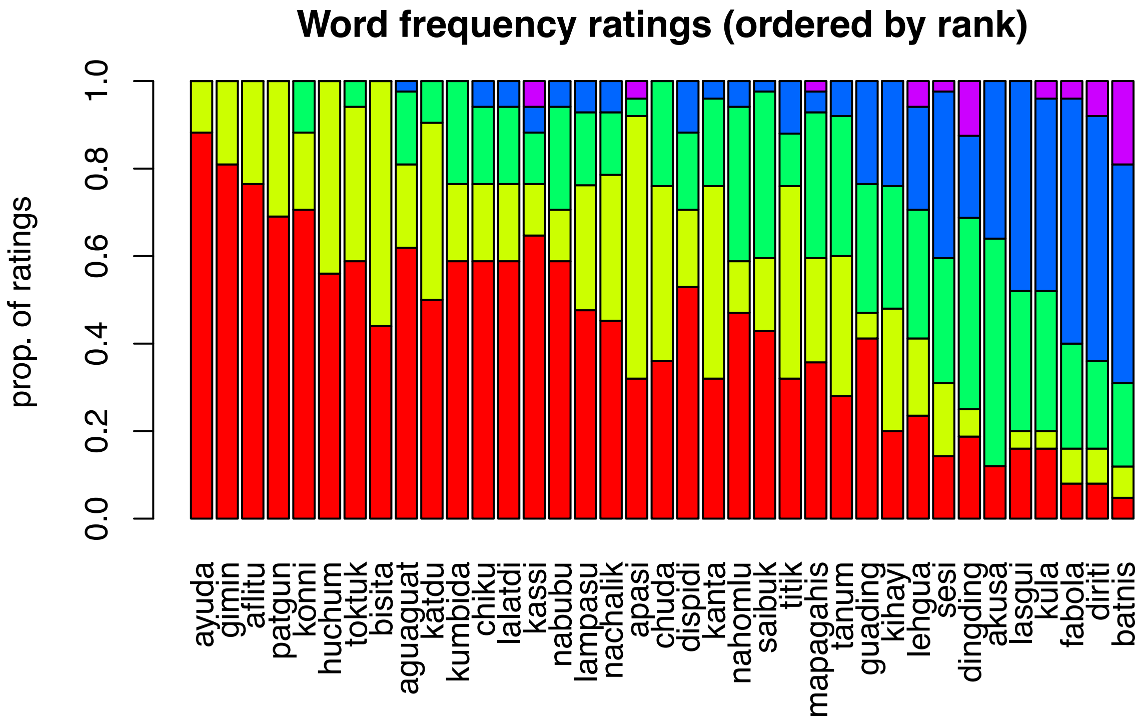
```
# We're going to place the histograms in a grid so we can compare them all at a glance.
# To make it roughly square, we'll need to know how many words there are.
n_words <- nlevels(ratings.df$word)

# Take the square root, and round it up and down to the nearest integer to form
# the dimensions of the grid.
grid_x <- floor(sqrt(n_words))
grid_y <- ceiling(sqrt(n_words))

# Set up the grid, and some appropriate margins around each plot
par(mfrow = c(grid_x, grid_y))
par(mar=c(1,2,1,1))

# Use barplot to spoof a histogram for each word
# I like barplot in this instance, instead of hist() over the raw data,
# because it's easier to control its behavior. #Notice we loop through
# the words according to the ranking we created above
for(w in words_ordered.ix){
  barplot(rating_freq.tab[w,], bty='n', space=0,
          names.arg=NA, ylim=c(0,1),
          main=w, col=rainbow(5))
  # Put the average on each plot
  text(x=4, y=0.8, round(rating_avg[w],2), font=3)
}
# The rainbow() palette will color the ordered categories from high to low:
# red, yellow, green, blue and purple.
```



We can see that words like *ayuda* (‘help’), *gimin* (‘drink’), *patgun* (‘child’) and (*konni’*) (‘catch’, ‘take or bring s.o.’) are at the top of the ranking, whereas words like *lasgui* (‘sharpen’), *kula* (‘sift’) or *batnis* (‘varnish’) are at the bottom.

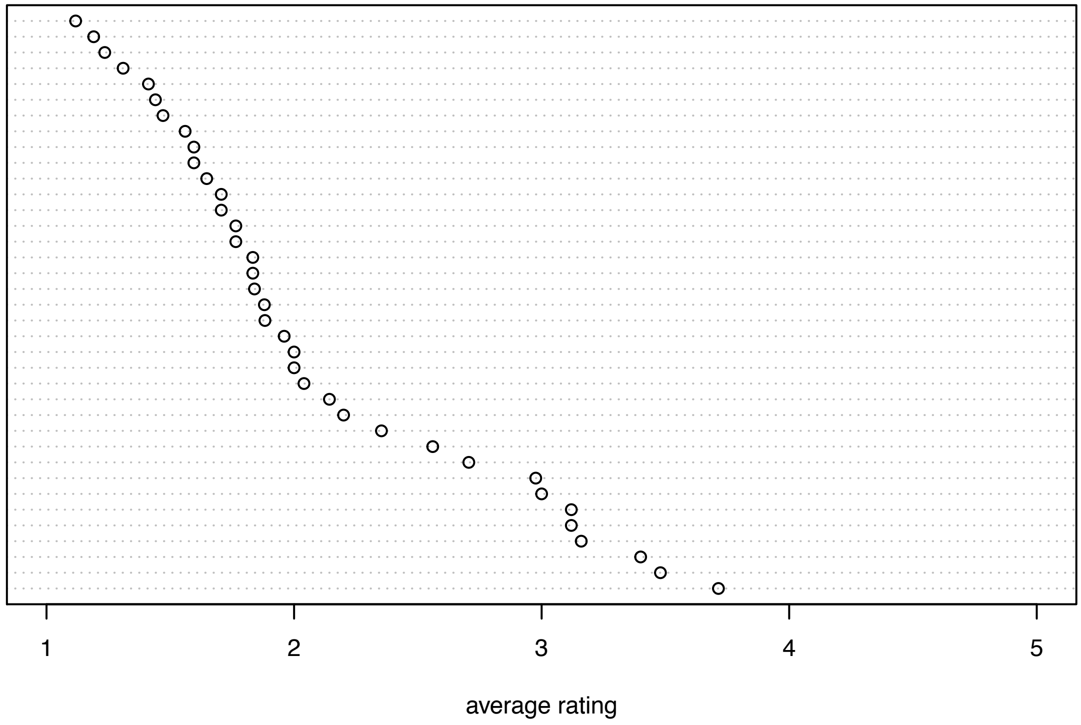
We can use a stacked bar plot as a visualization - this can be perceived somewhat more holistically.

```r
par(mar=c(8,5,3,1)) # Generous margins are needed for the labels.
par(mfrow=c(1,1))    # 1 row, 1 column
# In the call to _barplot_, we transpose the ratings matrix using _t_,
# so that each word is a column. The `las` arg prints the x-axis labels
# vertically so they're legible. The ordered words list is used as an row index.
barplot(t(rating_freq.tab[words_ordered.ix,]),
        col=rainbow(5), las=3)

title(main="Word frequency ratings (ordered by rank)",
      ylab="prop. of ratings")
```



A final kind of visualization:

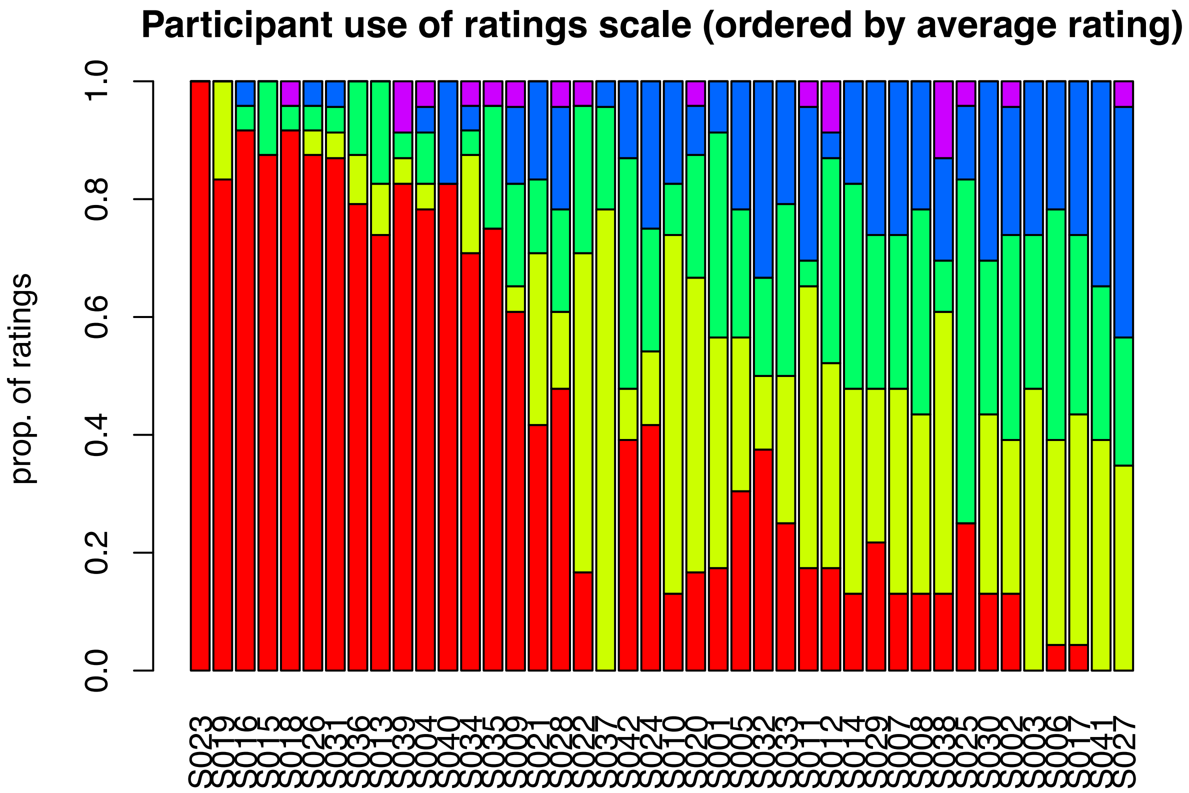
```
dot_ratings <- as.numeric(rating_avg[rev(words_ordered.ix)])
dotchart(dot_ratings,
         cex=0.75,
         xlab="average rating", xlim=c(1,5))
```



2.3 Variation by participant

As discussed above, one of the pitfalls of collecting survey data from a limited population is that we may end up with a small sample. In any case, it is important to understand the kinds of variation in our sample. In this part of the analysis, let's consider whether different participants used the scale differently.

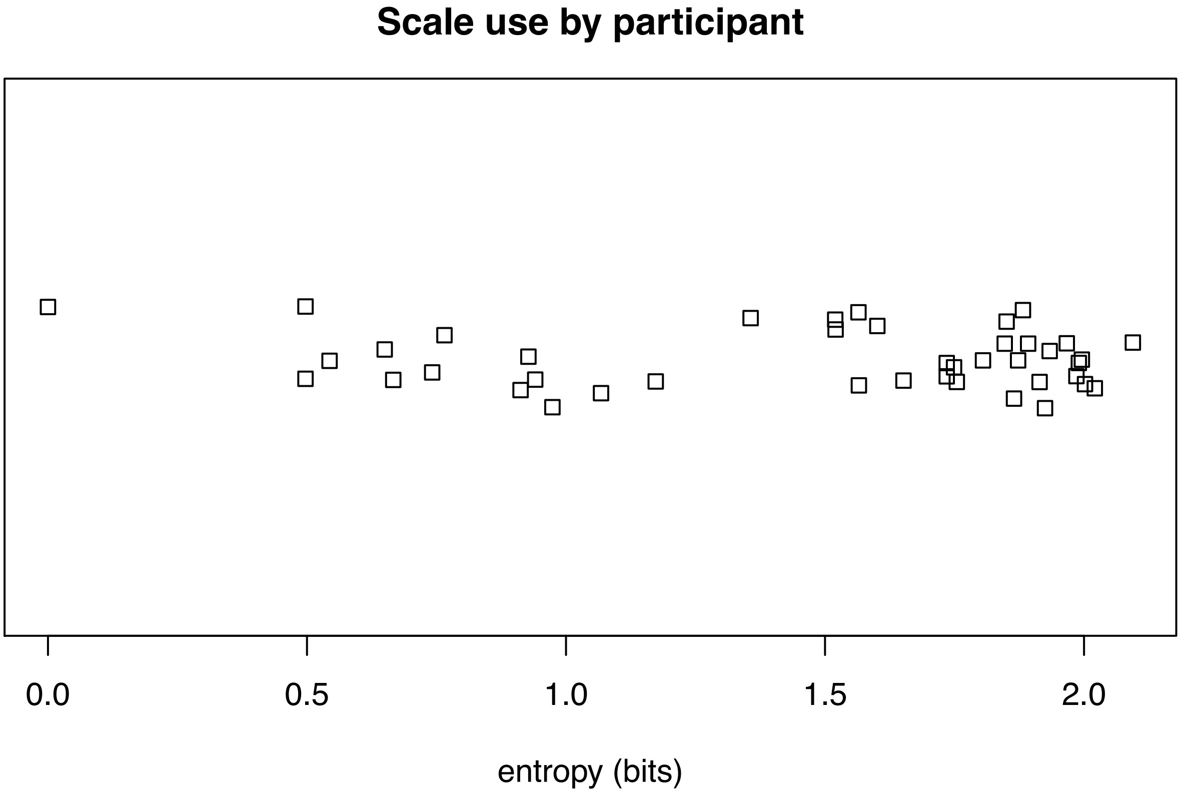
```
# Instead of counting ratings per word, we do it per participant
rating_counts.by_participant.tab <- with(ratings.df, table(participant, rating))
# ... compute frequencies
rating_freq.by_part.tab <- prop.table(rating_counts.by_participant.tab, 1)

# Then calculate the average rating
rating_avg.by_part <- with(ratings.df, tapply(rating, participant, mean, na.rm=1))
# ... and create a list of participants in order of average rating
order(rating_avg.by_part) -> rating_bymean.by_part.ix
participants_ordered.ix <- levels(ratings.df$participant)[rating_bymean.by_part.ix]

# Make a stacked barplotas before
par(mar=c(4,5,3,1))
par(mfrow=c(1,1))  # one row, one column
barplot(t(rating_freq.by_part.tab[participants_ordered.ix,]),
        col=rainbow(5), las=3, cex.lab=0.8)
title(main="Participant use of ratings scale (ordered by average rating)",
      ylab="prop. of ratings")
```



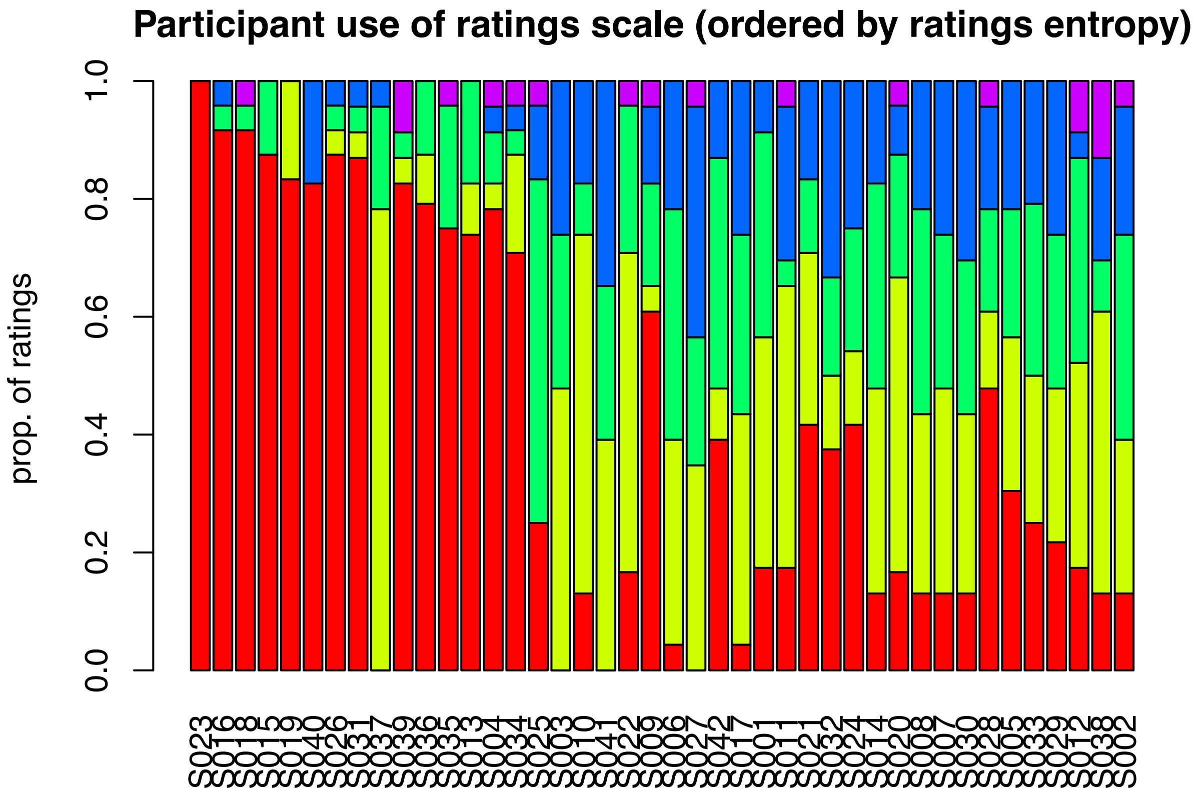
We can see that some participants essentially only assign the highest ranking, and do not use the scale in a uniform fashion. We can compactly quantify the informativity of each participant's responses by using a measure of entropy.

Per participant and rating category, we compute *-p log2(p)*, where *p* is the frequency with which the participant used that rating category. Then we sum those to come up with each person's ratings entropy. A person who used only one rating would have an entropy of 0 bits, whereas a person who used the ratings scale uniformly would have an entropy of 2.32 bits (`5 * (-0.2 * log(0.2, base=2))`).

```
# Calculate entropy per subject by
#... SUM(-p log_2 (p))
#... per each rating category
rating.entropy <- -rating_freq.by_part.tab * log(rating_freq.by_part.tab, base=2)
participant.entropy <- apply(rating.entropy, 1, sum, na.rm=1)

# Visualize with a stripchart
par(mar=c(5,1,3,1))
stripchart(participant.entropy, method="jitter",
           xlab = "entropy (bits)",
           main = "Scale use by participant")
```



If we visualize the ratings with stacked bar charts, and order this by each subject's ratings entropy, we can get a sense of how different individuals use the scale differently: some, with an entropy between 0.5 and ~1 essentially use just 2 ratings categories; others are using all the categories to some extent. No one uses the scale exactly uniformly, but this is not surprising since we did not control our the word stimuli we presented in a way that would make this a sensible outcome.



Immediately this raises the question of why there should be such variety among participants.

Perhaps participants conflate frequency and familiarity, so the highest rating essentially means, "I've heard of this word." Or, participants don't want to admit not knowing a word, or knowing something about a word, so they assign the category "todu i tiempu". Either or both of these are real possibilities in a social context in which there is awareness of language decline, and in which value attaches to being perceived as a good speaker.

We will keep this apparent dichotomy in mind in our future analysis.

We might expect to learn something about the words which these speakers *don't* endorse. I leave identifying these words as an exercise.

3 Cross-validating with corpus measures

3.1 How to obtain corpus counts

In the previous part of the tutorial, we used BootCaT and command-line tools to create a tokenized list of words in our corpus (with counts). To compare those counts to our ratings, we need to identify an equivalence class of word tokens for each of our survey items. For a language like Chamorro, there are two challenges: (1) multiple orthographies, each idiosyncratically used; and (2) rich morphology.

Responding to the orthography challenge requires domain-specific knowledge of how this language is written. The morphology challenge is deepened by morphophonological alternations, dialect variation, and the existence of stress-sensitive reduplication and infixation processes that can distort the stem.

Let's see some examples. For uniformity of presentation, we will use `R`s `system2` function to issue commands to the shell. But really this part of the analysis would be done by writing a program in a more appropriate setting (e.g., Bash, Perl, Python). We will primarily use regular expressions and the `grep` program to locate relevant entries in the corpus. For an excellent tutorial (and tool for testing regular expressions), visit

RegExr: you can put in a short word list and try out your searches there to see exactly how the matching is performed.

First an illustration of orthographic variation. Let's look up the word *huchum* ('close'). (To do so, we will first create a helper function that assembles the search expression and passes it to the command line.)

```
# Where's our corpus
corpus_file <- "tokenized_corpus.txt"

# Define a helper function
grepFromConsole <- function(search_expression){
# ARGS: An (extended) regular expression (string)
# RETURNS: Matches from the corpus, with counts
  # Need to quote the search expression to protect special characters
  search_expression <- paste('\"', search_expression, '\"', sep="")
  system2("grep", c("-E", search_expression, corpus_file),
          stdout="tmpfile")
  # be sure to use the extended (-E) regular expressions
  read.table("tmpfile", col.names=c("count", "surface_form"),
             quote="")
}
# You could do this natively in R by reading the corpus into a table,
# but you probably don't want to because its string functions are wonky.
# This code is just for illustration, and not a full-fledged solution.

# The first search
grepFromConsole("huchum")
```

```
##   count surface_form
## 1     1    umahuchum
## 2     2       huchum
## 3     2     mahuchum
```

Our first search returns 3 surface forms, with a total count of 5. However, in an older orthography, nonlow vowels in the final syllable of a word are generally spelled "e" or "o". So we should also search for *huchom*.

```
grepFromConsole("huchom")
```

```
##   count surface_form
## 1     1     huhuchom
## 2     1   mahuhuchom
## 3     2     mahuchom
## 4     4       huchom
```

Fortunately, we can use character classes to avoid doing disjoint searches. We'll use square brackets to denote an equivalence class.

```
captured_results <- grepFromConsole("huch[ou]m")
print(captured_results)
```

```
##   count surface_form
## 1     1     huhuchom
## 2     1   mahuhuchom
## 3     1    umahuchum
## 4     2       huchum
## 5     2     mahuchom
## 6     2     mahuchum
## 7     4       huchom
```

```
sum(captured_results$count)
```

```
## [1] 13
```

Finally, we need to consider morphological processes that don't concatenate outside of the stem. We will search for *-um-* or *-in-* infixation, using the * notation to capture 0 or more instances and the | to introduce disjunction.

```
captured_results <- grepFromConsole("h(um|in)*uch[ou]m")
# (um|in)* = 0 or more `um` or `in`
print(captured_results)
```

```
##   count surface_form
## 1     1     huhuchom
## 2     1     humuchom
## 3     1   mahuhuchom
## 4     1    umahuchum
## 5     2       huchum
## 6     2     mahuchom
## 7     2     mahuchum
## 8     4       huchom
```

```
sum(captured_results$count)
```

```
## [1] 14
```

Finally, because *-in-* triggers umlaut of the following vowel, we also need to see if *hinichum* is present.

```
captured_results <- grepFromConsole("h(um|in)*[ui]ch[ou]m")
print(captured_results)
```

```
##   count surface_form
## 1     1     huhuchom
## 2     1     humuchom
## 3     1   mahuhuchom
## 4     1    umahuchum
## 5     2     hinichom
## 6     2       huchum
## 7     2     mahuchom
## 8     2     mahuchum
## 9     4       huchom
```

```
sum(captured_results$count)
```

```
## [1] 16
```

At this point, we must be very careful to check our results. Collapsing /u/ and /i/ in the above search was fast and easy, but it was not made contingent upon the presence of *-in-*. Fortunately, there was no *hichum* or *hichom* in the output.

In the best scenario, we would write a program to take an input form and generate a set of complex words from that form. We could also use existing systems like Finite-state Morphology. Alternatively, we can write simple expressions that will overgenerate and then check the results by hand - this will probably be necessary anyhow to deal with idiosyncrasy.

Let's look at a more realistic search and identify any inflected form of the verb *ayuda* ('help').

If we try searching for *ayuda* on its own (try it!), we fill find many, many more non-Chamorro examples because of the nature of the web-scraped corpus and the other data it pulls in. However, we can use the morphotactics of the language to help us "target" a comparison class of Chamorro examples. In the following example, we only allow *ayuda* to be surrounded by acceptable Chamorro prefixes or suffixes (because it is a vowel-initial root, the *um/in* infixes will surface as prefixes).

```
# NB: all slash-escaped characters must be double-escaped (\\)
transitive_agreement <- "\\b([hj]a|u[n]|h{0,1}u|[ie]n){0,1}\\s?"
# Pers agreement for transitives; officially orthographically separated
# but often spelled without a space (\s);
# nothing else can precede these (\b)
vm_prefixes <- "(man*|fan*|na\'?|nina\'?|numa\'?|muna|um|in|a\'?)*"
# voice/mood morphs; some homophonous w number
# ma, man, fa, fan, na('), nina('), numa('), muna('), um, in, a'
possible_suffixes <- "-?([hks]u|mu|[ñn]a|ta|m[å,a]mi|miyu|[n,ñ]iha){0,1}$"
# either the word ends ($) or contains one of these suffixes

# create helper function to glom all these pieces together
searchInflected <- function(root){
  inflected_result <- grepFromConsole(paste(transitive_agreement,
                                       vm_prefixes,
                                       root,
                                       possible_suffixes, sep=""))
  return(inflected_result)
}

root <- "a(yu){1,2}da"
# account for reduplication of stressed syllable
# ayuda or ayuyuda
ayuda_results <- searchInflected(root)
# show some example results
ayuda_results[sample(1:nrow(ayuda_results), 7),]
```

```
##    count surface_form
## 7      2   manmaayuda
## 10     4  fanmanayuda
## 13     4    umayuyuda
## 8      3      ayudata
## 19    38      inayuda
## 14     5       uayuda
## 12     4     manayuda
```

```
# how many hits altogether
print(sum(ayuda_results$count))
```

```
## [1] 371
```

```
# Try another case
root <- "((ch|ñ)(um|in){0,1}i){1,2}ku"
# chiku, with 0 to 1 infixes
# the disjunct w `ñ` accounts for "nasal substitution""
# e.g., man + chiku -> mañiku
searchInflected(root)
```

```
##   count surface_form
## 1     1    chinikuku
## 2     1     fa'chiku
## 3     1       mañiku
## 4     3     umachiku
## 5     5      chiniku
## 6     6        chiku
```

It's important to recognize that we are using approximate, "good enough" solutions for the automation problem at hand. For example, the regular expressions above would allow many ungrammatical words, like *ha chumiku*. This is an issue only insofar as these words correspond, coincidentally, to words in other languages (the ones also scraped up in our corpus).

Moreover, we may not be finding all relevant forms. For example, there are some possible derived words involving stress shift that are not encompassed by the regular expressions above, like *chinikuku-ña*. However, as long as we do not arbitrarily change the search terms for each root of interest, then this probably will not harm us greatly.

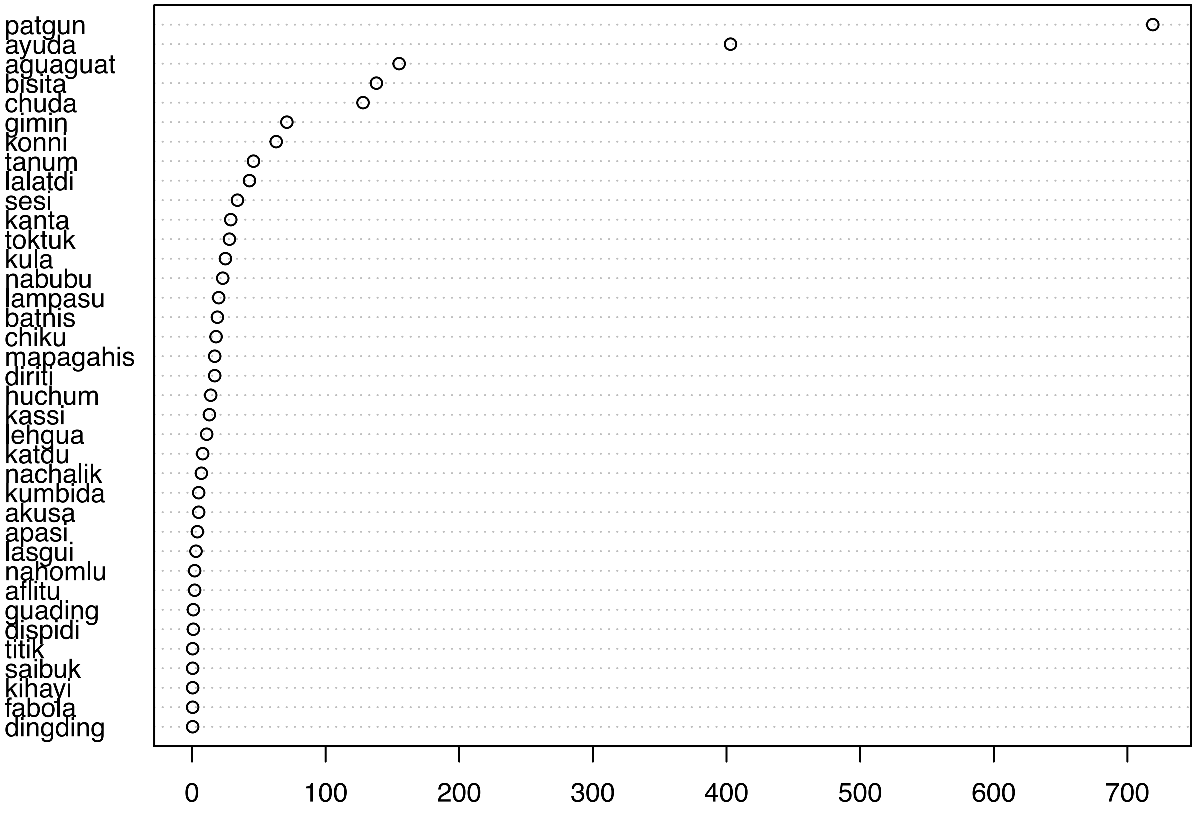
3.2 Comparing corpus counts to subjective frequency ratings

Corpus counts for our test words can be found here: http://people.ucsc.edu/~mwagers/aimm/corpus_counts.csv. These were collected using essentially the same search terms as above, and then the results were hand-inspected to exclude false alarms. Let's create a table with these counts and the ratings.

```
# Read in the counts from file
corpus_comparison.df <- read.csv("corpus_counts.csv")
# Add the average ratings as a column in this table
# important that data be in same order (here they're alphabetized)
corpus_comparison.df <- cbind(corpus_comparison.df, rating_avg)
```

Let's visualize the corpus counts by their rank order.

```
par(mar=c(3,10,1,1))
sorted.ix <- order(corpus_comparison.df$Corpus)
with(corpus_comparison.df,
     dotchart(Corpus[sorted.ix], labels=Word[sorted.ix], cex=0.8))
```



Notice that there are many cases in which we find no examples in the corpus. It is important to keep in mind that without a very large corpus, and one where some care is given to composition, the informativity of an absent word/form is not very great. Therefore, it is unsurprising that these words which are absent from the corpus nonetheless span a range of ratings categories. Here let us refer back to the actual ratings distribution.

```
# Identify words with fewer than 1 instance (coded as 0.5)
zeros <- which(corpus_comparison.df$Corpus<1)
# Print those words
corpus_comparison.df[zeros,1:3]
```

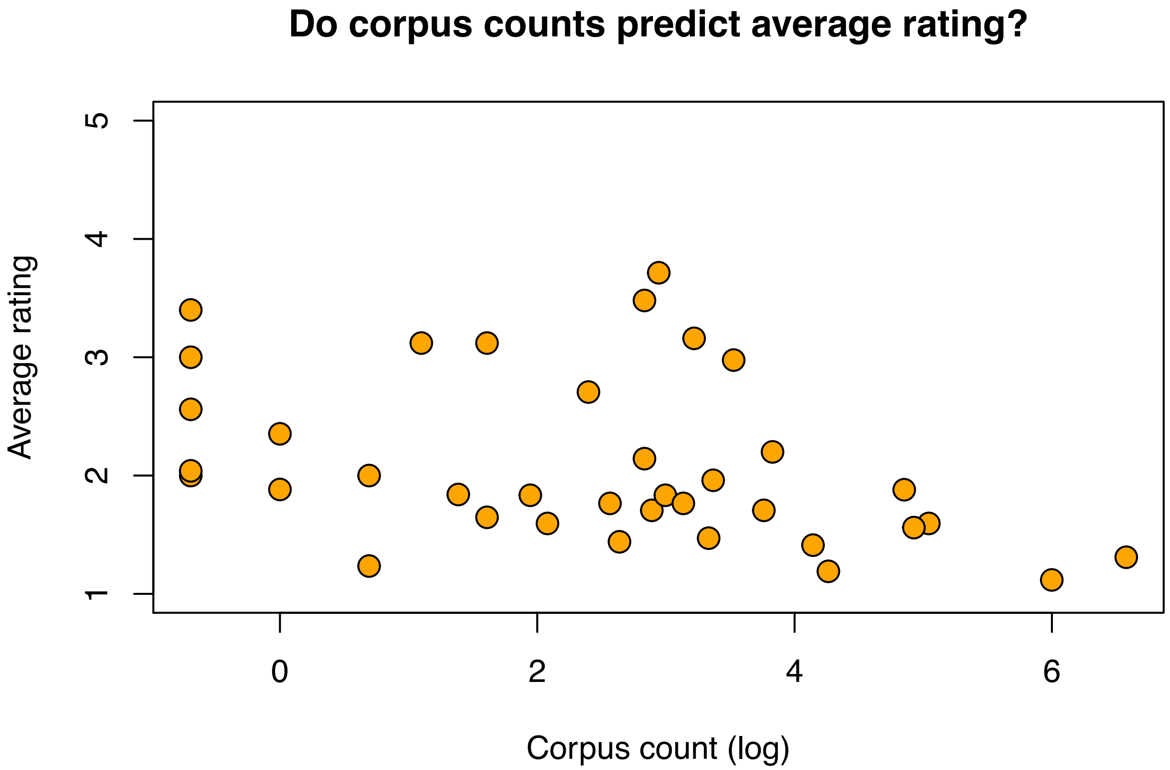
```
##              Word Corpus rating_avg
## dingding dingding    0.5       3.00
## fabola     fabola    0.5       3.40
## kihayi     kihayi    0.5       2.56
## saibuk     saibuk    0.5       2.00
## titik       titik    0.5       2.04
```

```
# Show the distribution of ratings for those words
rating_counts.tab[corpus_comparison.df$Word[zeros],]
```

```
##           rating
## word        1  2  3  4 5
##   dingding  3  1  7  3 2
##   fabola    2  2  6 14 1
##   kihayi    5  7  7  6 0
##   saibuk   18  7 16  1 0
##   titik     8 11  3  3 0
```

OK, keeping in mind that there is already some variation we cannot capture, we now visualize the relationship between average ratings and corpus counts (expressed as a log) using a scatterplot.

```
with(corpus_comparison.df,
     plot(log(Corpus), rating_avg,
          pch = 21, cex=1.5, bg="orange",
          ylim = c(1,5),
          ylab = "Average rating",
          xlab = "Corpus count (log)",
          main = "Do corpus counts predict average rating?"))
```



This looks promising: on average, higher corpus counts lead to lower ratings. But there is also a lot of noise, particularly in the mid-range, where some words seem to be affiliated with much lower ratings than others.

It's now worth recalling the split in how individuals used the ratings scale - some used the scale broadly, while others used only 1-2 categories. The latter participants contributed to a compression of the overall range of ratings, leading to lower ratings on average (i.e., higher estimates of frequency).

Let's see what happens if we only admit individuals into the analysis who used the scale broadly. Referring back to our entropy analysis, we can see an approximate break around 1.5 bits, so we'll use that as our cut-off. (One could do something more formal here …)

```
# Identify the participants with entropy > 1.5
broad_raters <- names(which(participant.entropy>1.5))
length(broad_raters) # how many are there?
```

```
## [1] 27
```

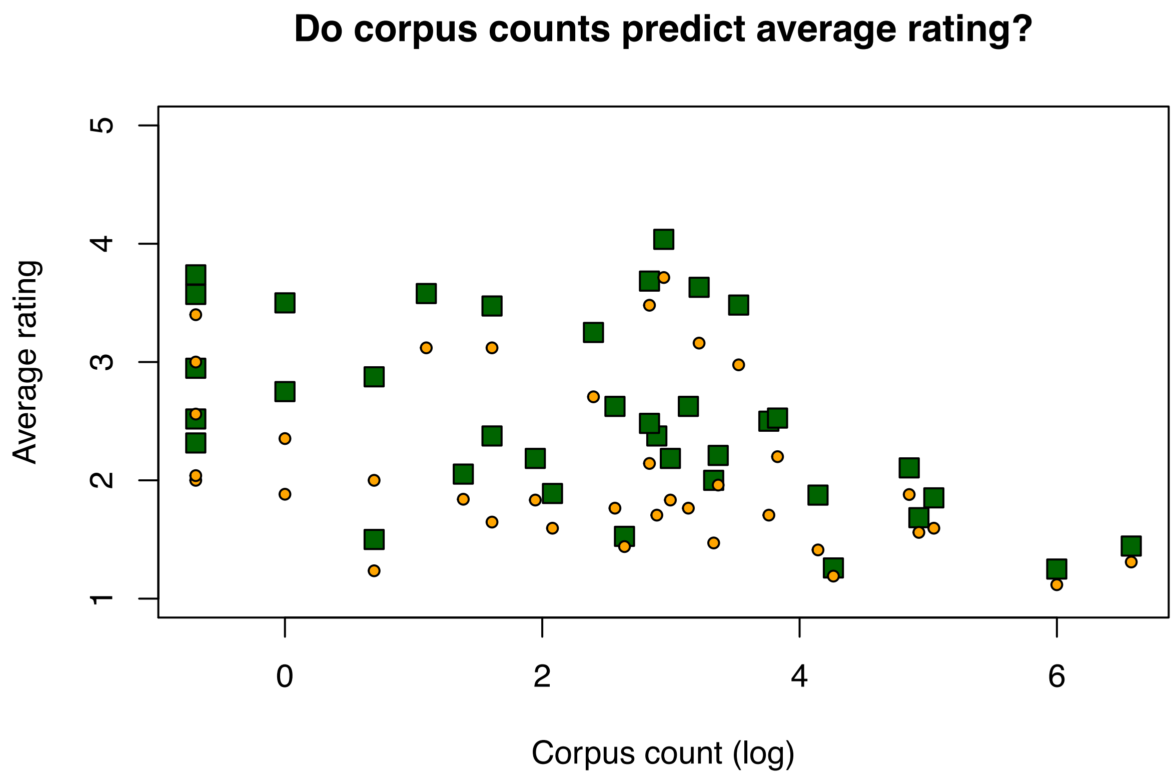
```r
# Recompute the average rating with only those participants
rating_subset <- with(subset(ratings.df, participant %in% broad_raters),
     tapply(rating, word, mean, na.rm=1))
# Add to data table
corpus_comparison.df <- cbind(corpus_comparison.df, rating_subset)
# Verify that these ratings are on average higher (=lower freq)
with(corpus_comparison.df,
     mean(rating_subset - rating_avg))
```

```
## [1] 0.436845
```

```r
# and that there's more variance
with(corpus_comparison.df,
     var(rating_subset)/var(rating_avg))
```

```
## [1] 1.206404
```

```r
# Visualize as before
with(corpus_comparison.df,
     plot(log(Corpus), rating_subset,
          pch = 22, cex=1.5, bg="darkgreen",
          ylab = "Average rating",
                    ylim = c(1,5),

          xlab = "Corpus count (log)",
          main = "Do corpus counts predict average rating?"))

# overlay ratings from all participants
with(corpus_comparison.df,
     points(log(Corpus), rating_avg,
          pch = 21, cex=.75, bg="orange"))
```



What we can see is that there is a greater ‘dynamic range’ of ratings for words in the middle of the corpus count spectrum.

Now we’re ready to test whether the apparent relationship between subjective frequency rating and corpus count is a reliable one. To do so, we’ll perform a simple linear regression.

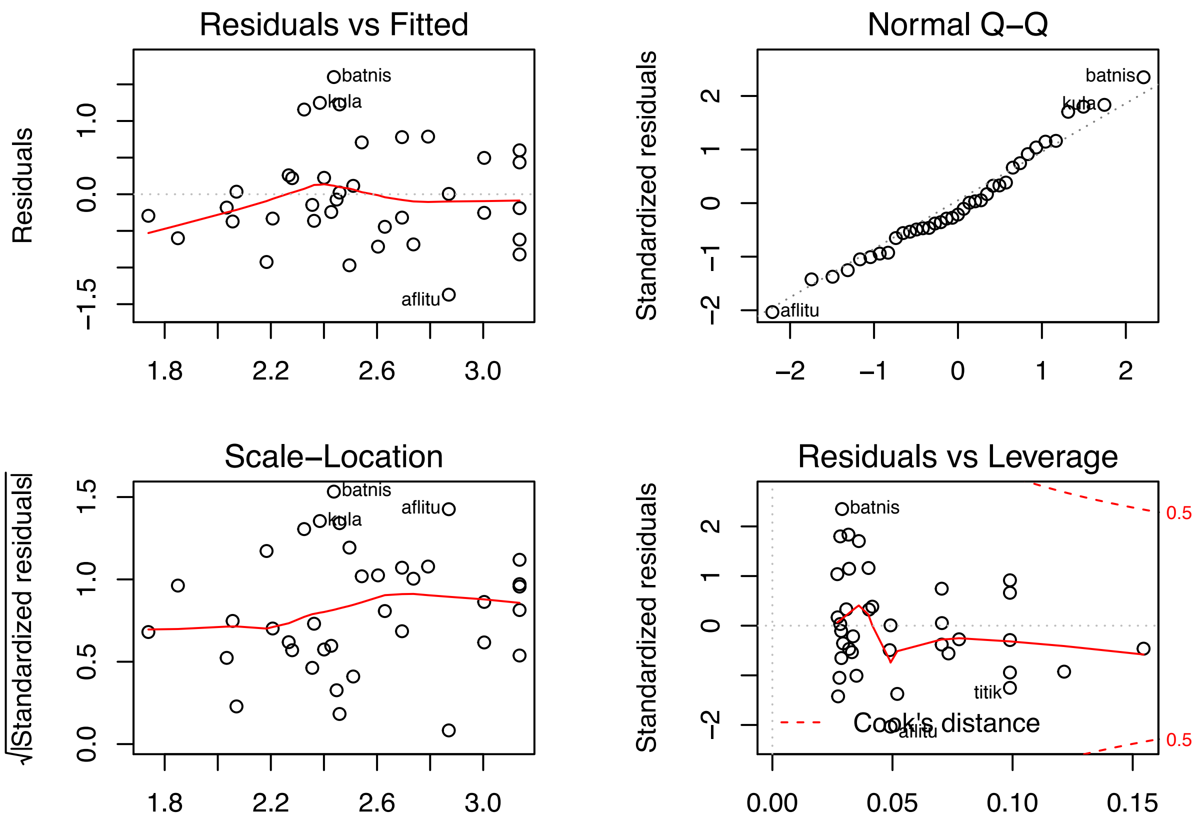
```
basic_model <- lm(rating_subset ~ log(Corpus),
   data=corpus_comparison.df)
# Test whether the average rating (in the broad rater subset)
# is related to the corpus counts (log)
summary(basic_model)
```

```
##
## Call:
## lm(formula = rating_subset ~ log(Corpus), data = corpus_comparison.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3704 -0.3718 -0.1455  0.4344  1.5997
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0037     0.1835  16.373  < 2e-16 ***
## log(Corpus)  -0.1923     0.0594  -3.238  0.00264 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6909 on 35 degrees of freedom
## Multiple R-squared:  0.2305, Adjusted R-squared:  0.2085
## F-statistic: 10.48 on 1 and 35 DF,  p-value: 0.002637
```

What this analysis reveals is that, indeed, as corpus count increases, the average rating reliably decreases. The R-squared statistic lets us know (roughly speaking) how much of the variance in ratings is accounted for by the corpus count predictor. In this case, we'll use adjusted R-squared, which is 0.21. We can compare this to an (adjusted) R-squared of 0.14 for a model based on data from all raters. [Cutting a corner here, which makes this comparison a little less straightforward - and that has to do with incorporating the variance around each word's rating.]

Finally, let's take a look at the residuals to see whether (a) the simple linear model is a sensible one; and (b) any items in particular are skewing the results.

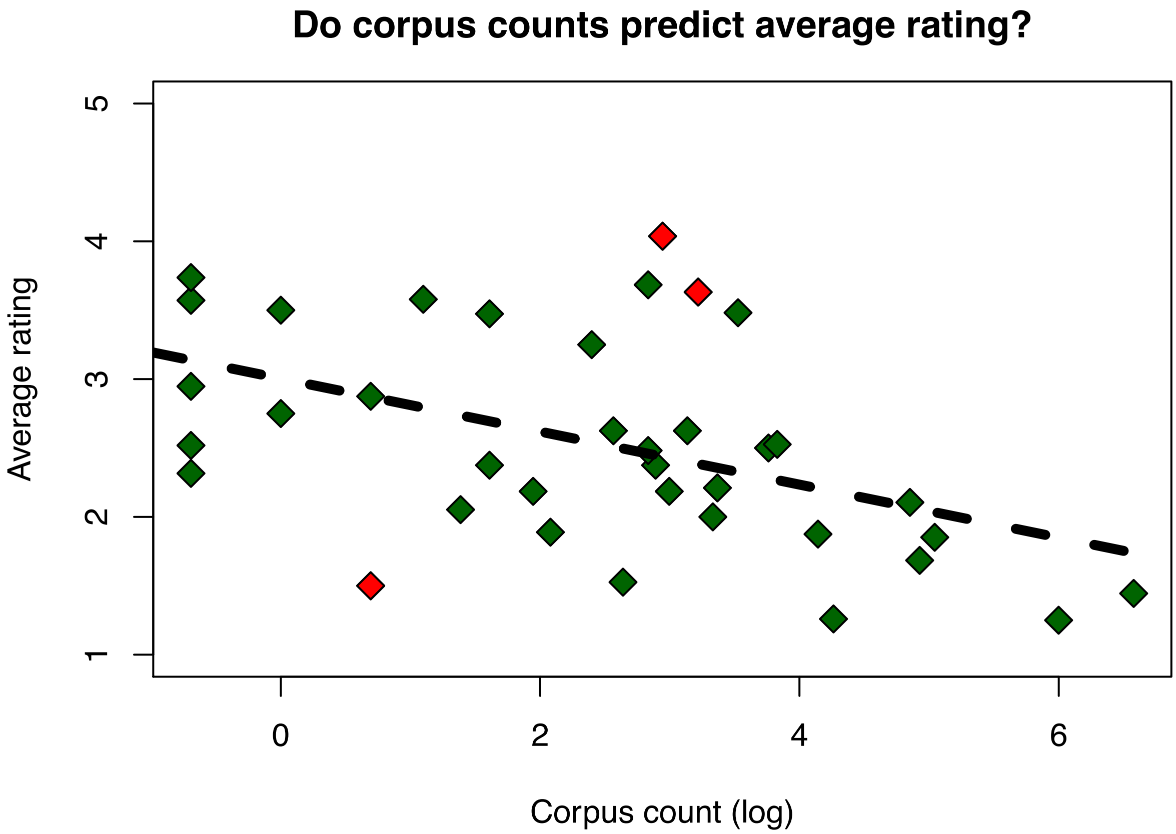
```
par(mfrow=c(2,2))
par(mar=c(3,5,2,2))
plot(basic_model)
```



```
# It's built-in for plot() to show these diagnostics
# when its argument is a fitted model
```

The words *batnis*, *kula* and *aflitu* have the largest residuals (top-left plot). In the case of *batnis* and *kula*, the rating is *higher* (less frequent) than expected based on the corpus count. In the case of *aflitu*, it's much lower (rated as more frequent than the corpus attests.) The following plot visualizes these words as (red) triangles. It also overlays the line represented by our linear regression.

```
par(mar=c(4,5,3,1))

par(mfrow=c(1,1))
with(corpus_comparison.df,
     plot(log(Corpus), rating_subset,
          pch = 23, cex=1.5, bg="darkgreen",
          ylab = "Average rating",
                    ylim = c(1,5),
          xlab = "Corpus count (log)",
          main = "Do corpus counts predict average rating?"))

with(subset(corpus_comparison.df, Word %in% c("batnis", "kula", "aflitu")),
     points(log(Corpus), rating_subset,
            pch = 23, cex=1.5, bg="red"))
abline(basic_model, lwd=5, lty="dashed")
```



Is there anything special about these words? Well, for one, they refer to highly domain-specific kinds of events ("sifting", "frying" - food preparation; "varnishing" - building/crafts). It seems likely that the kinds of knowledge and life experiences that our raters have are not necessarily reflected in the composition of our corpus (a large portion of which is religious text).

A final issue concerns the coding of those words with zero instances in the corpus. What happens if we drop those words from our analysis?

4 Surface form preference ratings: the case of Wh-Agreement

In the previous analysis of word frequency, we counted all the inflected forms of a single root. But sometimes we want to understand the statistical relationship between a particular stem and its affixes. For example, the transitional probability from a word family/lemma to a surface form has played an important role in investigations into how morphologically complex words are processed (Hay, 2001, Hay & Baayen, 2005, Solomyak & Marantz, 2010). It is not likely that we could easily estimate such quantities from such a small corpus as we now have, except perhaps for the most frequent affixes and words. However, we can come up with subjective proxies in certain instances.

In Chamorro questions of an object, there are two verb forms that can be used. In the first form, the verb stem is inflected with ordinary, transitive person/number agreement - *ha* in the example below.

```
Håyi  ha  chiku si Chai'?
who   3P  kiss     Chai'?
"Who did Chai' kiss?"
```

It is also possible to use a special form, called Wh-Agreement (Chung, 1998), in which *-in-* infixation co-occurs with a suffixal form of agreement seen in special verbs and possessor agreement.

```
Håyi  ch<in>iku-ña      si Chai'?
who   WH[Obj].kiss.3P     Chai'?
"Who did Chai' kiss?"
```

It is very difficult to know why speakers sometimes choose to use the "ordinary" form (which is the predominant one) and why they sometimes use the Wh-Agreement form. One natural hypothesis is the Wh-Agreement form may be used preferentially for certain verbs, but not others - perhaps very common verbs. For example, in our web-scraped corpus, the Wh-Agreement form is often observed with verbs like *tungu'* ('know'), *hassu* ('think, remember'), and *guaiya* ('love').

In one survey, speakers were presented with several pairs of sentences - like the pair give above with *chiku* ('kiss) - and asked:

```
Månu  mås   ya-mu?
which more  like-2P?
"Which do you prefer?"
```

An example of the survey can be seen at http://people.ucsc.edu/~mwagers/aimm/manu_mas_yamu.pdf. Data from this survey can be downloaded from http://people.ucsc.edu/~mwagers/aimm/manu_mas_yamu.csv.

First, let's load this data and take a look:

```
pref.df <- read.csv("manu_mas_yamu.csv")

# How many observations per word
table(pref.df$word)
```

```
##
##    aflitu     akusa     apasi     ayuda    bisita     chiku     chuda
##        14         8        11         9        14         8        11
##  dingding    diriti   dispidi    fabola   guading    huchum     kanta
##         9        14         8        11         9        14        11
##     kassi    kihayi     konni      kula   kumbida   lalatdi    lasgui
##         8        11         9        14         8         9        14
##    lehgua    nabubu nachotchu   nahomlu     tanum     titik    toktuk
##         8        11         9        14         8        11         9
```

```r
# How many observations per participant
table(pref.df$subj)
```

```
##
## S001 S002 S003 S004 S005 S006 S007 S008 S009 S010 S011 S012 S013 S014 S015
##    7    7    7    7    7    7    7    7    7    7    7    7    7    7    7
## S016 S017 S018 S019 S020 S021 S022 S023 S024 S025 S026 S027 S028 S029 S030
##    7    7    7    7    7    7    7    7    7    7    7    7    7    7    7
## S031 S032 S033 S034 S035 S036 S037 S038 S039 S040 S041 S042
##    7    7    7    7    7    7    7    7    7    7    7    7
```

```r
# Show a sample of the data
nrow(pref.df) -> n.obs
head(pref.df[sample(n.obs,10),])
```

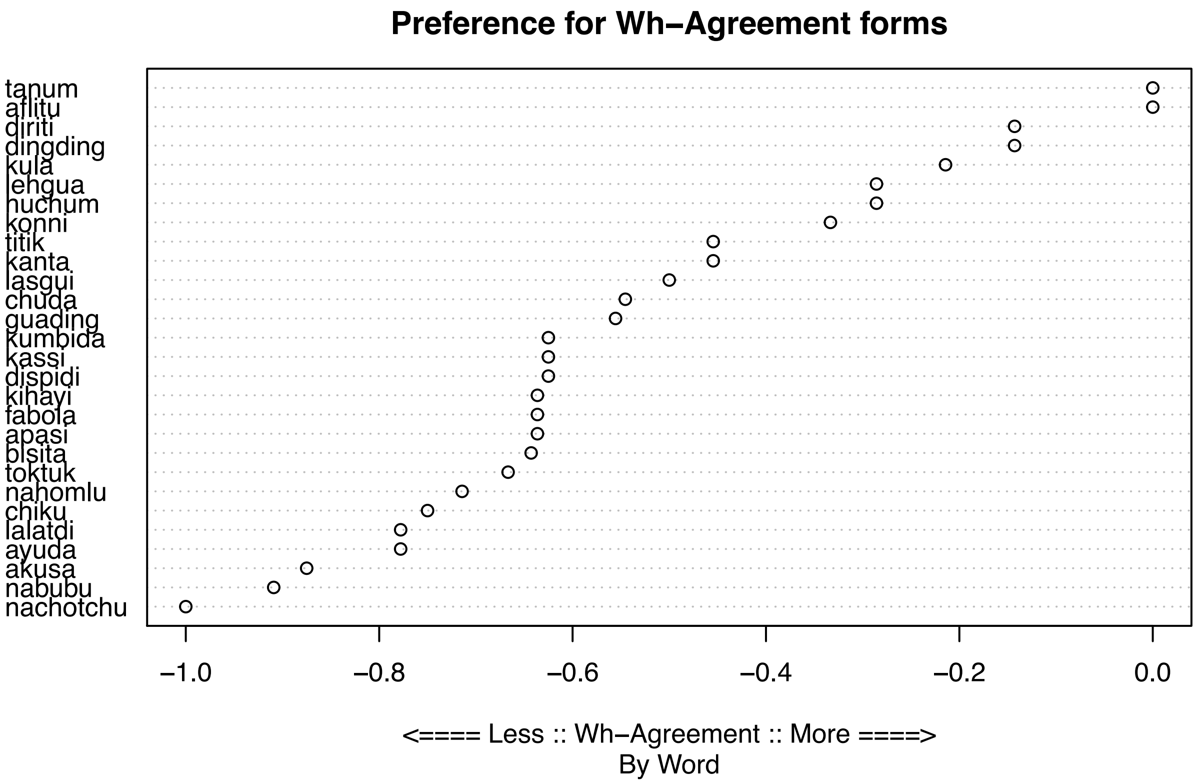
```
##       X    word preference.score subj
## 157 157   titik               -1 S029
## 15   15  lasgui               -1 S002
## 140 140   chiku               -1 S026
## 258 258 lalatdi                0 S024
## 47   47  huchum               -1 S007
## 4     4 nahomlu               -1 S005
```

For sentence pairs in which participants selected the “ordinary” form, the trial was coded as -1; 1 was assigned for the Wh-Agreement form. Some participants - spontaneously and without instruction - selected both answers [perhaps indicating equivocality??]. Those were coded as 0. Sometimes no answer was given, and that was coded as `NA`. It’s worth observing that there’s not nearly as much data from this task.

Let’s compute an average preference score for each word and visualize the results:

```r
# Take an average preference score, classified by the factor 'word'
# Use na.rm=1 because not all trials got an answer
preferences_by_word.tab <- with(pref.df, tapply(preference.score, word, mean, na.rm=1))

# Sort by preference score order and visualize
preference.ix <- order(preferences_by_word.tab)
par(mar=c(7,5,3,1))
dotchart(preferences_by_word.tab[preference.ix], cex=0.8,
         main = "Preference for Wh-Agreement forms", sub="By Word",
         xlab = "<==== Less :: Wh-Agreement :: More ====>")
```

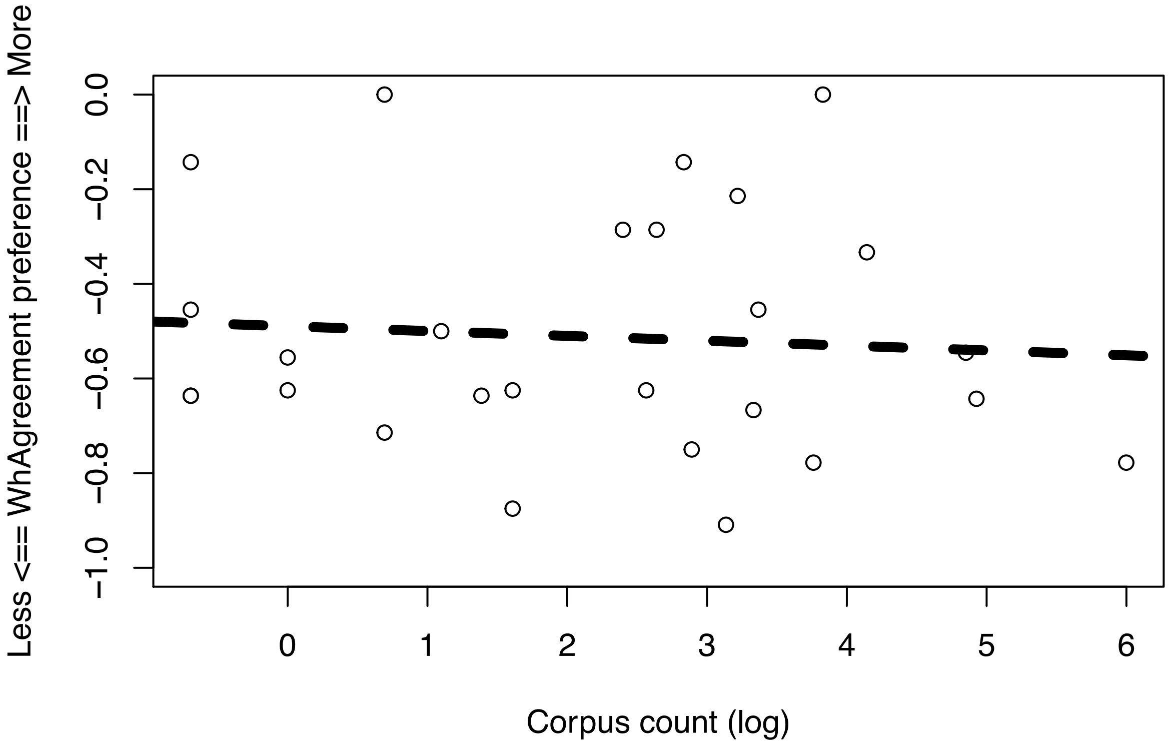
```
## Warning in dotchart(preferences_by_word.tab[preference.ix], cex = 0.8,
## main = "Preference for Wh-Agreement forms", : 'x' is neither a vector nor a
## matrix: using as.numeric(x)
```



Consistent with observations from the field, there is no word for which Wh-Agreement is preferred to ordinary agreement: ordinary agreement predominates. The average preference score is -0.5290146.

Let's see if there's a relationship between word frequency and the Wh-Agreement preference.

```r
# First, create a scatter plot with rating
word.ix <- names(preferences_by_word.tab) # create a word index to match the two data sets
# Plot the corpus data first
plot(log(corpus_comparison.df[word.ix, "Corpus"]), preferences_by_word.tab,
     ylab = "Less <== WhAgreement preference ==> More",
     xlab = "Corpus count (log)")

# Compute a linear regression model to overlay on the plot:
preference_model <- lm(preferences_by_word.tab ~ log(corpus_comparison.df[word.ix, "Corpus"]))
abline(preference_model, lwd=5, lty="dashed")
```



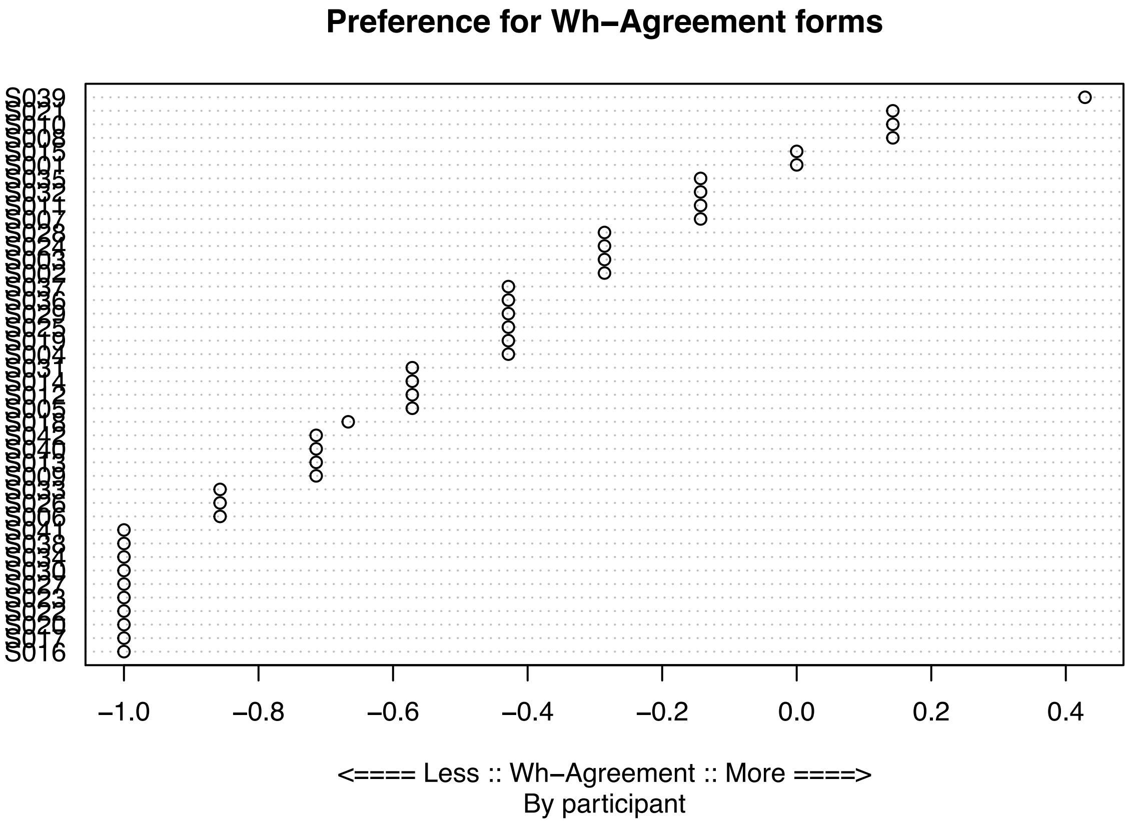
There doesn't appear to be a relationship in this dataset. The correlation coefficient, a measure of the strength and direction of association, is pretty small: -0.0764206. Inspection of the model shows it to be non-significant, and as inspection of the graph suggests, it does not control much variance: there is perhaps a 0.1 change in the preference score over the whole range of the corpus counts.

We leave it as an exercise to see whether using the subjective ratings is any more enlightening.

It is almost certainly true that some individuals might be more or less likely to use Wh-Agreement - but are there any individuals who only use Wh-Agreement? Or who only use ordinary agreement? We can compute the average preference score by individual as well. (Also remembering that the number of observations we have per person is quite small).

```
# Take an average preference score, classified by the factor 'subj'
# Use na.rm=1 because not all trials got an answer
preferences_by_part.tab <- with(pref.df, tapply(preference.score, subj, mean, na.rm=1))

# Sort by preference score order and visualize
preference.ix <- order(preferences_by_part.tab)
dotchart(preferences_by_part.tab[preference.ix],sub="By participant",
         main = "Preference for Wh-Agreement forms", cex=0.8,
         xlab = "<==== Less :: Wh-Agreement :: More ====>")
```

```
## Warning in dotchart(preferences_by_part.tab[preference.ix], sub = "By
## participant", : 'x' is neither a vector nor a matrix: using as.numeric(x)
```



We find that some individuals *never* prefer Wh-Agreement; that some individuals prefer Wh-Agreement in more cases; but that no individuals completely avoid ordinary agreement.

5 Relationship to online measures

A final issue we can take up is whether any of our corpus or survey data can be used to shed light on data more closely related to incremental processing. In what follows, we will address a specific problem about why a morphologically more complex form takes longer to process. But, in general, cross-validation of behavioral and corpus measures in the context of smaller language research is welcome in that it may improve our understanding of the variability of data obtained from experiments conducted in the field, in which item & participant samples are often much more diverse than in the laboratory (e.g., Christianson & Ferreira 2005, Harris & Samuel 2011; Gagliardi & Lidz 2013; Clemens et al. 2014; Wagers, Borja & Chung 2015) or - better yet - lead the way to entirely new questions.

In the following experiments, participants listened to sentences containing movement dependencies which are either marked with Wh-Agreement or not marked with Wh-Agreement. To measure incremental processing, the technique of *self-paced listening* was used (Ferreira et al. 1996). Self-paced listening is a moving-window technique in which a larger expression is spliced into smaller segmeents (ideally phonological phrases). Participants successively press a button to hear the phrases one after the other and they learn to press the buttons at a rate which makes the sentence sounds ~*more or less* natural. Self-paced listening can be implemented on a variety of experimental platforms such as E-Prime, OpenSesame (python) or in the simple, flexible (and free) Linger program (Rohde, 2003). It is a good option when literacy is at issue (for example, because of the orthographic difficulties discussed before).

In this experiment, the sentences have a structure similar to the following example. The | (pipe) marks the segment boundary - in this case, there are 4 audio segments.

```
Bula manhobin | ha kihåyi si Kiku' | dispues di manmumu  |  gi iskuela
```

```
many youths     3P tattle           after      PL.fight   LOC school
"There were many kids who Kiku' tattled on after they were fighting at school."
```

Two factors were manipulated: the identity of the verb, and whether or not it bore Wh-Agreement. In the following example, the verb is anomalous for its object *manhobin*. The Wh-Agreement form is given after the slash:

```
Bula manhobin | ha lehgua'/linigua'-ña si Kiku' ...
many youths     3P stir/WH[Obj].stir     Kiku' ...
"There were many kids who Kiku' stirred ..."
```

At issue in the experiment was whether the morphology on the verb aids in the construction of the Wh-Dependency (and thus speeding the detection of anomaly). At issue for us today is the following observation:

- Participants listened to Wh-Agreement-marked verbs longer.

Let's look at the data (http://people.ucsc.edu/~mwagers/aimm/listening_times.csv).

```
listening_times <- read.csv("listening_times.csv")    # load data

# Show region-by-region (WNUM) averages by the two different
# levels of agreement. RT == listening times/button advancement times.
with(subset(listening_times),
     tapply(RT, list(agreement, WNUM), mean, na.rm=1))
```

```
##              1        2        3        4
## 3-wh 1331.194 1695.497 1274.500 2371.909
## 3+wh 1391.413 1864.525 1300.378 2221.136
```

```
# These are expressed in ms (milliseconds)
```

An obvious concern is that the audio duration of Wh-Agreement marked verbs is simply longer than that of ordinary verbs. However, if we subtract the audio segment duration from the listening time, we still find longer listening times for Wh-Agreement marked verbs:

```
# Subtract audio duration from listening times
listening_times$RTrelative <- (listening_times$RT - listening_times$duration)

# Average listening times by condition and WNUM (segment number)
with(subset(listening_times),
     tapply(RTrelative, list(agreement, WNUM), mean, na.rm = 1))
```

```
##              1        2        3        4
## 3-wh 296.2556 349.9553 460.0778 1356.333
## 3+wh 305.5642 387.1229 506.9278 1218.653
```

The difference in listening times is now substantially reduced on the verb itself, but we still see a slow-down in the region following the verb (a kind of delay or 'spill-over' effect). This effect is significant (uncomment and evaluate the following code chunk to prove it to yourself; you'll need the lme4 library).

```
# library(lme4)
# contrasts(listening_times$agreement) <- -contr.sum(2)/2
# region3_model <- lmer(RTrelative ~ agreement + (agreement|Subj),
#                      data=subset(listening_times, WNUM==3 & accuracy==1))
# summary(region3_model)
```

For our final worked 'vignette', let's ask whether the slow-down for Wh-Agreement verbs is related to the Wh-Agreement preference score we obtained in the previous analysis.

Suppose that verbs are more easily processed in their Wh-Agreement form when, for whatever reason, their Wh-Agreement form is more common or more preferred – for example, because those common forms are (stochastically) available as wholes in a dual-route model of morphological processing. The prediction would then be that the higher the preference score, the less the Wh-Agreement slowdown.

To test this prediction we first need to calculate, for each item, how much it is slowed-down per region in the Wh-Agreement conditions.

```
# Take the mean listening time per condition/segment/word
mean_RTs.by_word.tab <- with(subset(listening_times, accuracy==1),
     tapply(RTrelative, list(agreement, WNUM, Word), mean, na.rm=1))

# Take the difference score between levels of agreement
# The numbers in c(2,3) refer to the margins (in this case, WNUM and Word)
wh_effect.by_word.tab <- apply(mean_RTs.by_word.tab , c(2,3), diff)
```

Let's first see if there's any relation in Region 2:

```
# Create a table matching words with their WhAgreement effect in Region 2
word.ix <- colnames(wh_effect.by_word.tab) # what words do we have
wh_effect.cor_table.reg2 <- cbind(preferences_by_word.tab[word.ix],
                                  wh_effect.by_word.tab[2, ])



# Test for a correlation in Reg2
wh_correlation.reg2_model <- lm(wh_effect.cor_table.reg2[,2] ~ scale(wh_effect.cor_table.reg2[,1]))
# using scale() to center the predictor
summary(wh_correlation.reg2_model)
```

```
##
## Call:
## lm(formula = wh_effect.cor_table.reg2[, 2] ~ scale(wh_effect.cor_table.reg2[,
##     1]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1306.92   -62.72    76.03   175.53   837.58
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -14.62      92.06  -0.159    0.875
## scale(wh_effect.cor_table.reg2[, 1])  -105.81      94.54  -1.119    0.275
##
## Residual standard error: 459.9 on 23 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.05164,    Adjusted R-squared:  0.01041
## F-statistic: 1.252 on 1 and 23 DF,  p-value: 0.2746
```

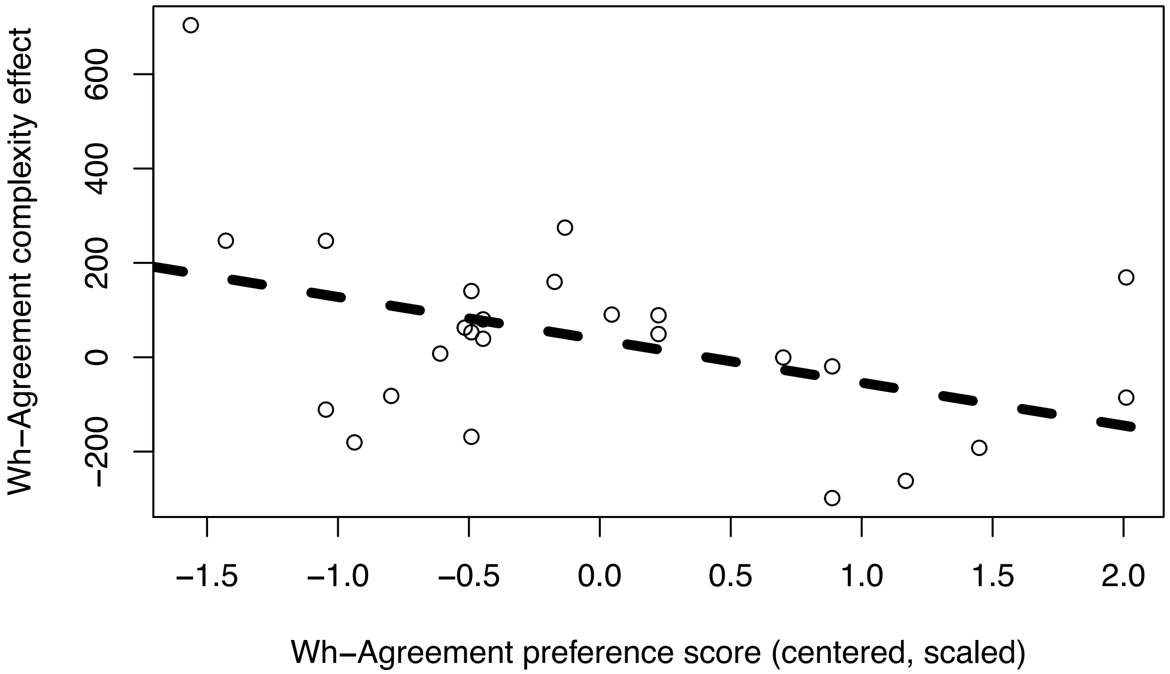
Doesn't look like it. But of course, the effect was only robust on average in Region 3. So now consider that analysis:

```
# Create a table matching words with their WhAgreement effect in Region 3
wh_effect.cor_table.reg3 <- cbind(preferences_by_word.tab[word.ix],
                                  wh_effect.by_word.tab[3, ])

# Test for a correlation in Reg3
wh_correlation.reg3_model <- lm(wh_effect.cor_table.reg3[,2] ~ scale(wh_effect.cor_table.reg3[,1]))
summary(wh_correlation.reg3_model)
```

```
##
## Call:
## lm(formula = wh_effect.cor_table.reg3[, 2] ~ scale(wh_effect.cor_table.reg3[,
##     1]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -302.50  -97.28   24.39   72.46  525.09
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             36.89      38.39   0.961   0.3466
## scale(wh_effect.cor_table.reg3[, 1])   -90.83      39.43  -2.304   0.0306
##
## (Intercept)
## scale(wh_effect.cor_table.reg3[, 1]) *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.8 on 23 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.1875, Adjusted R-squared:  0.1521
## F-statistic: 5.306 on 1 and 23 DF,  p-value: 0.03063
```

The model shows that the Wh-Agreement forms get reliably easier if they come from words with higher Wh-Agreement preference scores.

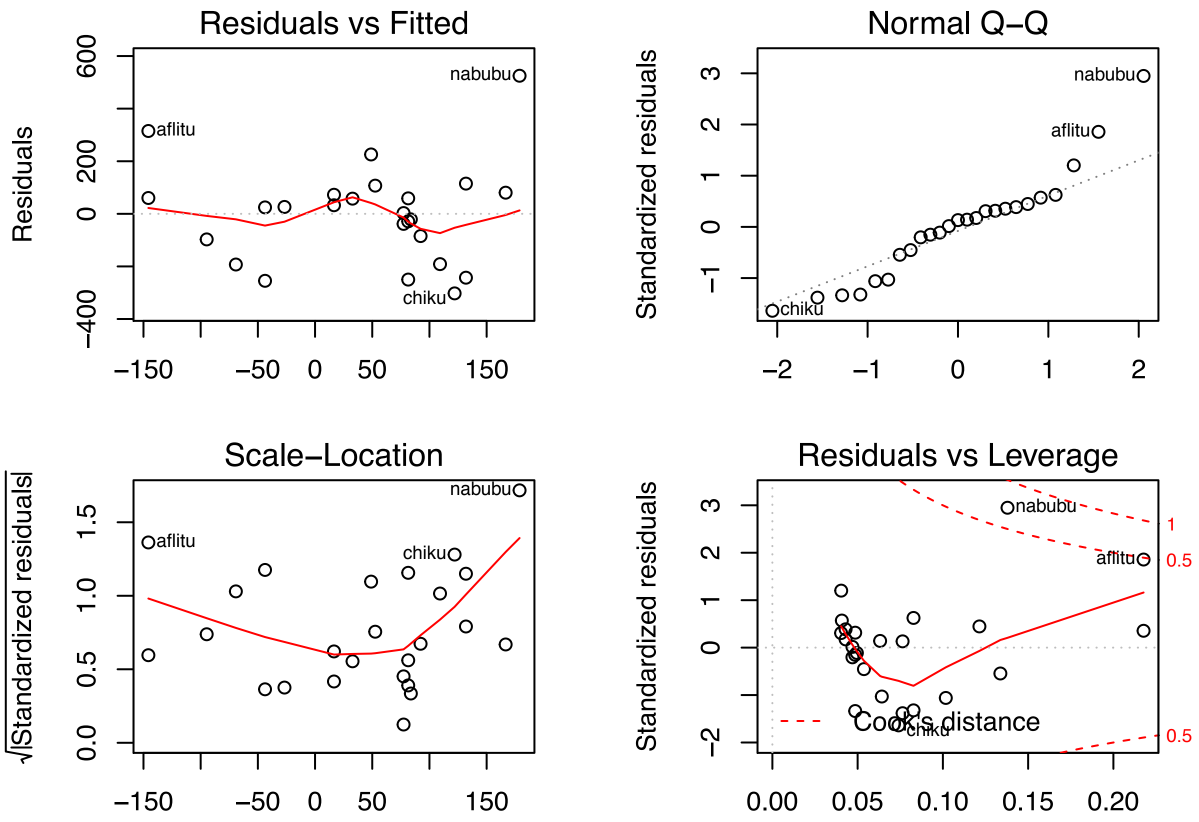
```
par(mfrow=c(1,1))
plot(scale(preferences_by_word.tab[word.ix]), wh_effect.by_word.tab[3,],
     xlab ="Wh-Agreement preference score (centered, scaled)",
     ylab ="Wh-Agreement complexity effect")

abline(wh_correlation.reg3_model, lwd=5, lty="dashed")
```



Now that we've visualized it, it looks as if one data point could be exercising too much power over the fit - the one in top-left corner: *na'bubu/nina'bubu-ña* ('(to) anger').

A residuals plot analysis similarly shows that *na'bubu* and possibly *aflitu* are exerting out-size influence.

```
par(mfrow=c(2,2))
par(mar=c(3,5,2,2))
plot(wh_correlation.reg3_model)
```

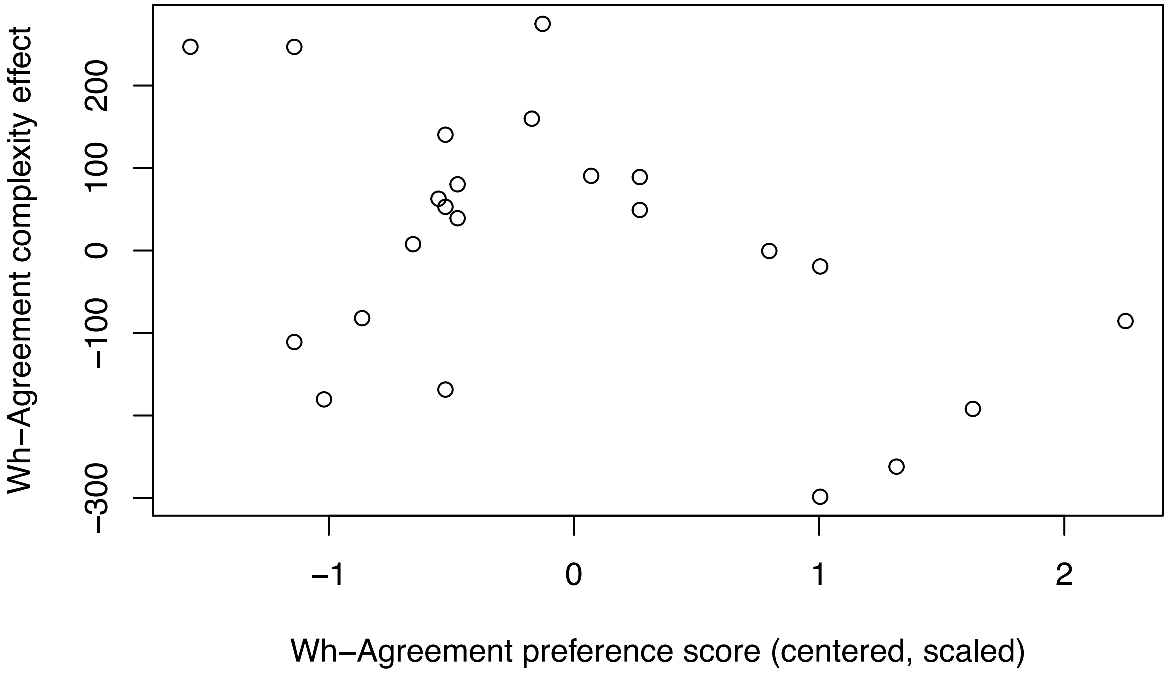


Let's remove these and see what happens.

```
# Remove over-influential data points:
word.ix.2 <- setdiff(word.ix, c("nabubu","aflitu"))
wh_effect.cor_table.reg3 <- cbind(preferences_by_word.tab[word.ix.2],
                                  wh_effect.by_word.tab[3, word.ix.2],
                                  corpus_comparison.df$rating_subset[word.ix.2])
wh_correlation.reg3_model_2 <- lm(wh_effect.cor_table.reg3[,2] ~
                                  scale(wh_effect.cor_table.reg3[,1]))
summary(wh_correlation.reg3_model_2)
```

```
##
## Call:
## lm(formula = wh_effect.cor_table.reg3[, 2] ~ scale(wh_effect.cor_table.reg3[,
##     1]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -259.13 -110.78   42.41   96.09  262.84
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            2.347     30.204   0.078   0.9388
## scale(wh_effect.cor_table.reg3[, 1])  -74.841     31.513  -2.375   0.0272
##
## (Intercept)
## scale(wh_effect.cor_table.reg3[, 1]) *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 144.7 on 21 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.2117, Adjusted R-squared:  0.1742
## F-statistic:  5.64 on 1 and 21 DF,  p-value: 0.02716
```

```
par(mfrow=c(1,1))
plot(scale(preferences_by_word.tab[word.ix.2]), wh_effect.by_word.tab[3,word.ix.2],
     xlab ="Wh-Agreement preference score (centered, scaled)",
     ylab ="Wh-Agreement complexity effect")
```



The effect survives - though we must still exercise caution, as there is much residual variance and potential outliers. However, it provides fodder for thinking about future experiments and analyses. Two follow-up analysis ideas would be to see whether this effect extends to Region 4, and to ask whether overall word frequency interacts with the preference for the WhAgreement form.

Finally, it would be worth figuring out whether other variables that correlate with incremental behavior match up with either our corpus or survey-derived measures. One idea concerns the duration of the speech files: should we expect to find a dependency between pronunciation duration and frequency?

6 References

Borja, Joaquin Flores; Manuel Flores Borja; and Sandra Chung. 2006. Estreyas Marianas: Chamorro. Saipan, CNMI: Estreyas Marianas Publications.

Clothier-Goldschmidt, Scarlett. 2015. The distribution and processing of referential expressions: evidence from English and Chamorro. M.A. Thesis, University of California, Santa Cruz. URL: http://chamorro.sites.ucsc.edu/downloads/clothier-goldschmidt_thesis/.

Connine, Cynthia M.; John Mullennix; Eve Shernoff; and Jennifer Yelen. 1990. Word familiarity and frequency in visual and auditory word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 6.1084–96.

Cooreman, Ann. 1987. Transitivity and Discourse Continuity in Chamorro Narratives. Berlin: Mouton de Gruyter.

Ferreira, Fernanda; John M. Henderson; Michael D. Anes; Phillip A. Week; and David K. McFarlane. 1996. Effects of lexical frequency and syntactic complexity in spoken language comprehension: Evidence from the auditory moving window technique. Journal of Experimental Psychology: Learning, Memory and Cognition 22.324-335.

Gordon, Barry. 1985. Subjective frequency and the lexical decision latency function: Implications for mechanisms of lexical access. *Journal of Memory and Language* 24.631–45.

Hay, Jen. (2001). Lexical frequency in morphology: Is everything relative. *Linguistics* 39.1041-1070.

Hay, Jen; and R. Harald Baayen (2005). Shifting paradigms: gradient structure in morphology. *Trends in Cognitive Sciences* 9. 342-348.

Rohde, Doug 2003. Linger: a flexible platform for language processing experiments, version 2.94. Online: http://tedlab.mit.edu/~dr/Linger/

Solomyak, Olga; and Alec Marantz. 2010. Evidence for early morphological decompo- sition in visual word recognition: A single-trial correlational MEG study. *Journal of Cognitive Neuroscience* 22.2042–57.

Topping, Donald M.; Pedro M. Ogo; and Bernadita C. Dungca. 1975. Chamorro-English Dictionary. Honolulu: University of Hawaii Press.

7 Appendix: Cumulative link model analysis of frequency ratings

We can discuss this, during or after the tutorial, if there is time or interest.