9.59 Lab in Psycholinguistics: Problem Set #2

Reading and processing Turk data

In this problem, you will get some slightly preprocessed data from Amazon's Mechanical Turk from an experiment run in the lab in 2012 (now published in Mahowald, Fedorenko, Piantadosi, & Gibson, 2013). (It will be a good idea to save this code to re-use for your own Turk project.)

In this project, Mahowald et al. looked at word pairs like chimp/chimpanzee and math/mathematics. A prediction from information theory is that words that are predictable in context should be shorter. If you already know what the next word will be (as in "To be or not to. . . "), there is no reason to spend a lot of time and effort saying a long word. On the other hand, if the word is entirely unpredictable ("The next word I am going to say is . . ."), the word should be longer and thus more robust to noise. That is, if one syllable gets garbled or misunderstood, the meaning can still be recovered. To test whether English is efficient in this regard, they gave subjects on Turk sentences like "Susan loves the apes at the zoo, and she even has a favorite . . . " (supportive context) or "During a game of charades, Susan was too embarrassed to act like a . . . " (neutral context) in which they were asked to complete the sentence with either the word chimp or chimpanzee.

Here is a sample trial from the supportive condition:

> Susan loves the apes at the zoo, and she even has a favorite. . .
>
> 1. chimp 2. chimpanzee

There were 40 longshort word pairs, and they were presented to the subjects in random order such that each subject saw a different order of sentences. They also varied which form appeared first (the short one or the long one). They predicted that the shorter form would be more common in the predictable construction than in the unpredictable one for a whole list of such word pairs. We want you to look and see if this prediction is correct!

- Download the csv files long_short_turk.csv and decode_long_short.csv and put them in the same folder as the R script you are creating. This data set is from a behavioral experiment run on Turk in the Gibson lab.
- Read in the results data from the csv file so that you have a data frame containing everything from long_short_turk.csv. This is your main results file. HINT: Make sure you have the `tidyverse` package loaded using `library()`
- You now have a data frame with many columns. As is typical of Turk output, there is a row for each subject. This is not ideal for analysis in R. Rearrange the data using `gather()` so that there is one row for each individual trial. That is, because there are 80 items on the survey, each Subject will have 80 Answer Choice rows.
- You should now have the following columns:
  - WorkerId: unique Turk ID for each subject
  - Input.list: Each subject receives her own random order (identified by the list number).
  - Answer.country: Country of subject
  - Answer.English: Whether the subject's native language is English
  - variable: This is output by the `gather()` function. See below.
  - value: The value for the variable output by `gather()`.

The column variable consists of two parts separated by an underscore. The first part indicates whether that row's value specifies the input, a question, or the answer choice. For our purposes, you are interested in when the value is 'AnswerChoice'. The part after the underscore specifies where on the subject's list that individual item appeared. So when a row has 'AnswerChoice_1' in the variable column, that means that the

value column will say whether that subject picked Choice 1 or Choice 2 for their first question on the survey. Note that the first question on the survey is likely to be different for every subject.

- To get them into their own columns, use `separate()` to split the 'variable' column at the underscore into 2 columns: 'Type' and 'PresentationOrder'
- When Type is either InputTrial or InputQuestion, that means that the row contains either the prompt given to the subject or a question asked to see if the subject was paying attention. You can ignore these for now by filtering out everything in the dataframe except the rows where Type is AnswerChoice.
- Read in the file decode_long_short.csv into a separate data frame. This new data frame contains the information you need to actually match up the items and words with the Turk output. Merge the decode dataframe with your main data frame, using `left_join()`, based on the columns PresentationOrder and Input.list. The Item column is consistent across subjects. That is, Item 1 refers uniquely to the pair math/mathematics.
- For some trials, the subject may have failed to answer the question. In this case, an NA will appear in the dataframe. Remove all NA rows from the data frame using `na.omit()` on the data frame.
- If you have column names like "value" or "variable", you should change them since those words are used by R and you might get errors. Change "value" to AnswerChoice using `rename()`.
- You should also remove the items that fall into the Filler condition. These were used in the experiment to prevent the subjects from noticing that the task was about long and short forms of the same word. Filter the subjects so that we have only English speakers and only people from the US.
- To get the data in a meaningful form, you should add a column called PickedShort that is 1 if the subject picked the short form and is 0 otherwise. You can do this by using AnswerChoice and Condition to determine what the subject picked for each trial. You might want to use `separate()` on Condition (similar to what we did a few steps above). This will give you a Cond column that is either "neut" for neutral or "supp" for supportive, and a First column that is either "longfirst" or "shortfirst" (depending on whether the long form or short form was shown first).
- AnswerChoice tells you 1 or 2 for which option was chosen. If the long form was first (i.e. First == "longfirst") and the AnswerChoice is 1, that means that the person picked the long form. If the short form was first and the AnswerChoice is 2, that means that the person chose the long form. Use this knowledge to set the PickedShort column to 1 if the subject picked the short form or 0 if they picked the long form. HINT: you can use `mutate()` and `if_else()`.
- Now that the data are "tidy", you can start analyzing and looking for patterns. The item of interest is whether the person picked the long or short form and what the condition was. Give a short quantitative discussion of the results. Make sure to answer the following questions. There is some flexibility in how you answer the question, but be sure to report numbers (no inferential statistics necessary) and be clear about which numbers you are reporting.
  - How many subjects remain in the experiment after exclusion?
  - Averaging across all trials, on what proportion of trials did someone pick the short form?
  - How does the proportion who picked short differ in the neutral condition vs. the supportive condition?
  - Which word pair had the highest overall PickedShort percentage?
  - Which was least often used in the short form?
  - Which word had the biggest difference between the supportive and neutral condition?
  - Do subjects seem to choose the item that appears first more often than the one that appears second?

- Does it look like there is a shift in preference for long vs. short words over the course of the experiment?
- What else do you notice in this data set that is worth pointing out?
- What sorts of things still need to be controlled for in this analysis?
- What conclusions do you draw from this experiment?

9.59J/24.905J Lab in Psycholinguistics
Spring 2017