III. EYEDRY

**Note: See EYEWARN.DOC for information about the necessary sequencing of condition numbers in a file of materials used in an experiment.**

**And see the stuff at the end of the file….**

A. Overview. Eyedry computes first fixation duration, first pass duration, total reading time, the percentage of regressions out of each region, the percentage of regressions into each region, the percentage of times each region was fixated on the first pass, the second pass duration, etc., for each specified region of each input sentence in an experiment, as well as the concentrated looks at a single region, which used to be called EYEPOKE and EYEGAZE (see below). It also reports the counts of the numbers of subjects who contributed observations to each region. A region can be a single word, or it can be a series of words, or it can be a portion of a word. Imagine that you have reading times for sentences like (1), in which regions are separated by '^':

(1a). The big bully^ always thought^ he knew^ girls^ bite back^ sometimes.^

(1b). Friends and neighbors^ claim^ that^ my time^ has^ finally^ come.^

You may want to compare the reading times for the "disambiguating" regions of these sentences, "bite back" in (1a) and "has" in (1b). This region is the 5th region in each sentence (regions are separated by ^). You would like to get the mean reading times for (a) this critical region, (b) the region right after it, (c) the region right before it, and (d) the disambiguating complementizer "that" in the second sentence. That's what EYEDRY does for you: It will write out disk files that contain this information for each condition of an experiment on both a subject-by-subject basis and an item-by-item basis.

B. First, prepare a .CNT file. Copy the boilerplate EYEDRY.TEM from C:\DATA onto your subdirectory, rename it (e.g., expnmED.C), stick the core of your experiment-running program where it says "YOUR PROGRAM STARTS HERE," compile it with ETCC or ETCC206. Then run it once. (You can make separate .CNT files for separate subexperiments by making copies of the relevant parts of the overall .CNT file, or you could run the expnmED program several times, or you can leave the overall .CNT file intact and have EYEDRY select the appropriate parts from it.) The program expnmWP runs basically like your experiment-running program ran, except (a) you should tell it 1 for subject condition and counterbalancing condition, and (b) you should indicate no data file. When it asks for the position count output file, tell it the name you want to give your .CNT file (include the extension). If you like, the program will give you a hard copy of the sentences as seen by your subjects, broken up into regions.

If you want your regions to be single words, just press the SPACE bar when the program asks for delimiters (thereby telling it that the delimiter is ' '). Otherwise, prepare a file that is a modification of your experiment .SEN file. You should indicate the **end** of each region you want to analyze in this file, by inserting some convenient character at the end of each region of each sentence (like the ^ in (1), above), and use this as the input file. (NOTE CAREFULLY: The delimiter comes right after the last character in a region, and before the first character in the next region - which in general will be the space before the first word in that region.) Tell the program the delimiting character you used. Make sure that the regions you want to compare across sentences have the same ordinal positions in each sentence (in the example above, the critical region had position 5). The program will create a 20-character region following the last delimiter.

Note: EYEDRY normally assumes contiguous item numbers for an experiment. If your experiment has noncontiguous item numbers (e.g. if you used a randomization in which the experimental items occurred scattered through the whole list, with accordingly scattered item numbers), you can either modify eyedry by adding a vector to map the actual item numbers onto a contiguous series (e.g., 1, 2, 3,..,24), or you can just leave things alone and let PUTAWAY clean up your mess. It really is better though to have contiguous item numbers; lots of memory and disk space is wasted otherwise -- eg the .itm and the .ixs files will have lots of lines with nothing but zeros in them if you have noncontiguous item numbers. You can let the experiment running program randomize things or you can use prepared lists to do the randomizing.

C. Prepare a file of data file names. Call it (by default) DATA.LST. One data file name (the output of EYEWASH) should be on each line. For example:

T1.DA1

T2.DA1

T3.DA1

D. Examine a data file to see what column contains the condition number, the item number, the "number of observations" number (number of fixations in the sentence), and the start of the vector of reading times. (Typically, these will be columns 2, 3, 8 and 9.)

D.Run EYEDRY. Type a convenient label, which will be printed on the printed output as an identifier. The program types out the date automatically. If you have run EYEDRY before in an experiment, you can tell in to load a "control" file (generally, .CON or .CRN) in which you have saved your answers to a bunch of questions. Otherwise, you will have to answer some questions (after which you will have a chance to save your answers in a disk file). The first question asks you about debug level. Type 0. Tell it no, there are not questions after every item (unless there are, and they are on a different line than the item; and even here, you can generally tell it no anyway). Answer the rest of the questions. Tell it the smallest and the largest sentence numbers and the smallest and largest condition numbers you plan to analyze. NOTE: This lets you analyze a big experiment with multiple subexperiments in it component by component. You can have separate .CNT files for each subexperiment, or you can have just one .CNT file. Tell it the names of your file of data file names (DATA.LST) and the .CNT file (e.g., expnm.CNT) when it asks. Tell EYEDRY which analysis you want it to do; it will loop through the possible analyses until you tell it you are done. For each analysis, give EYEDRY the names of the files you want it to output data to. These will be inputs to the subjects and the items ANOVAS. You will generally not want to have it write a file of subjects x items combinations, unless each subject is tested in all conditions on each item.

E. EYEDRY allows you to have an exceptions file (which has entries like:

item# +-changetocondition

item# +-changetocondition

This lets you change the condition number of specified items (e.g., to correct errors in the original sentence file, or -- by specifying +99 as changetocondition -- to delete an item entirely). EYEDRY also allows you to discard trials when a question was answered incorrectly, if you wish.

F. The EYEPOKE and EYEGAZE routines of EYEDRY have some peculiar characteristics. EYEPOKE computes the duration of 3 fixations before and 3 fixations after the first fixation in a single specified region, or the mean of the signed values of the distances of each of these fixations from the first fixation specified in the region. The first distance can be interpreted as the mean absolute value of the length of the first saccade, either regressive or progressive, after the fixation; but the other distances can only be interpreted as distances, since each fixation position results from a mixture of regressive and progressive saccades. EYEGAZE computes the first pass duration on a single specified region (which, if it is a single word, results in gaze duration for that word).

EYEPOKE's and EYEGAZE's big trick is that, if they don't find a fixation in the specificed region, they move the left boundary of a region left, character by character, until they find a fixation or until they have moved the region as far left as you permitted it to; then they try once again, moving the right boundary one character right. Then they give up. (They never move a boundary around a line break). Thus, EYEGAZE and EYEPOKE minimize the problem of missing data (under the assumption that information from a region can be acquired from just outside that region.

The output of EYEDRY is suitable for input directly to COLANOV, a very simple-minded ANOVA program (and by adding group numbers, can be used in SYSTAT, ECSTATIC, etc.). As well as containing the data you asked for, most data files also contain the number of observations that contributed to each datum. The data are formated as "(10x,n(f6.0,4x))" for for n regions of an item. They are arranged subject-by-subject (or item-by-item), subjects (items) moving most slowly, and within each subject (item), they are arranged by condition, lowest to highest, one condition per line.

**Note on PUTAWAY.** PUTAWAY is a filter program that takes the items-by-subjects output of EYEDRY and transforms it into a form suitable for input to commercial stats programs. It provides one line per subject (or per item), containing a vector of 8-digit (2 decimal point) numbers, ordered so that all the conditions for analysis region 1 come first on a line, then the conditions for analysis region 2, etc., with conditions ordered first, second, third...within each region. PUTAWAY contains several options, e.g., you can keep or discard means whose value is zero and you can truncate a trial when some specified number of values of zero are found in successive regions of a sentence (e.g., when 2 regions are skipped).

**Addition 12/97:** An option was added to allow EYEDRY to deal with double-spaced displays. Simply tell EYEDRY ‘y’ that an experiment used double-spacing (works for triple-spacing, etc. as well) and it will filter out the blank line between successive lines of text when calculating region length. Beware, though, that region length is not computed correctly when a region is allowed to wrap between two lines; see below.

**Warning**: Because of the way EYEDRY calculates the starts and ends of regions, region length will be calculated incorrectly when a region wraps between two lines. Only the end of a region is indicated in your .DEL file; the next region is assumed to start immediately thereafter. Therefore, you cannot indicate where text ends at the end of a line without starting a new analysis region – which you don’t want to do if you want regions to wrap across lines. This has two consequences: (a) ms/char will be calculated incorrectly for regions that wrap across lines, and (b) the parameters of the regression equation used in the deviation from predicted measure will be calculated incorrectly. You can deal with the latter problem by preparing a delimited file that does NOT wrap across lines, using it to compute your regression parameters, and moving back to your line-wrapped .DEL and .CNT file for final computation. However, even this move will result in incorrect values for the specific regions that actually wrap across lines.

char version[] = "EYEDRY VERSION OF 7/26/11 (compiled with mingw-GCC)";

/\*...................................................

copyright 1990 Charles Clifton. Users of this program are permitted

to modify and distribute it at no charge but may not sell it or

include any of its code in a commercial program.

Modification of old SEGMENT program.

Assumes data consisting of:

... cond# ... item# ... #fixations in sentence x1 y1 st1 et1 x2 y2 st2 et2 ...

optionally with question to be discarded after each trial

VERSION DESIGNED TO TAKE EXCEPTIONS FILE, IN FORMAT:

item# +-changetocond#

item# +-changetocond#

...

6/88: if changetocond == 99, item is thrown out

it does the following things:

1. writes a file of mean values for each combination

of subject,item, subcondition, and region (where

subcondition refers to the first, second,...nth

condition used for an item)

2. writes a file of mean values for each combination

of subject, condition, and region, averaged over items

3. writes a file of mean values for each combination

of item, condition, and region, averaged over subjects

4. writes to the screen and option`y the printer the

means over items, and the means over subjects and

items, of the data for each condition and region

To run it, you need a control file with one line for each

item/condition combination used in the experiment,

containing the following information:

item# cond# #regions start of 1st region (=0) start of 2nd region

.... start of nth region

When the control file gets read in, the condition number for

a sentence gets stored as the start of the first region

which must always be 0 anyway.

Note, the numbers in the control .cnt file are the STARTS of each

region, counting from 0 as the start of the first word.

Note, the condition numbers indicated are those in the datafile

before adjustment from the exceptions file.

The control file can be generated by combining the program EYEDRY.TEM

and your experiment-running program, and compiling and running the

resulting program.

1/99 Added options to permit contingency on prior fixtion plus other

options listed below with \*\*

\*\*Newly implemented

1 = first fixation

option: raw or ms/ch

option: conditionalize on presence or absence of regression from a specified region

\*\*option: conditionalize on there being exactly one fixation in region

\*\*option: conditionalize on there being more than one fixations in region

\*\*option: second of two contiguous fixations in region

\*\*option: conditionalize on prior fixation being between i and j characters

before beginning of region

\*\*option: report fixation immediately before entering region

option: conditionalize on presence or absence of fixation in a specified region

2 = first pass time

option: keep vs. discard trials with no appropriate fixations

option: raw, ms/ch, deviation times

option: conditionalize on presence or absence of regression from a specified region

\*\*option: conditionalize on prior fixation being between i and j characters

before beginning of region

option: conditionalize on presence or absence of fixation in a specified region

3 = total time

option: keep vs. discard trials with no appropriate fixations

option: raw, ms/ch, deviation times

option: conditionalize on presence or absence of regression from a specified region

\*\*option: conditionalize on prior fixation being between i and j characters

before beginning of region

option: conditionalize on presence or absence of fixation in a specified region

4 = first pass regressions out (Percentage of trials where one or more

legal first pass fixations in region were followed by a legal

fixation to earlier point in the sentence.)

5 = regressions in (Percentage of trials where one or more legal fixations

in region were preceded by legal fixation in a later region of the sentence.");

6 = probability of a first-pass fixation

\*\*option: probability of there being exactly one fixation

\*\*option: conditionalize on prior fixation being between i and j characters

before beginning of region; reports pr(fix|prior fix OK) but does not

count trials where the "prior fix" was the last fixation in the sentence

i.e. it gives the p(fix|fix or skip)

\*\*option: duration of fixation just before or just after skipping region

or landing in region

7 = number of fixations

1 = count fixations prior to leaving region

2 = count all fixations in region

8 = second pass times

option: raw, ms/ch, deviation times

1 = include only time in region after going past it

2 = include all re-reading time after leaving region, left or right

3 = include only time in region after leaving it to left, before going past

4 = include only re-reading time in region after entering second specified region (Tim Slattery)

9 = eyepoke

option: raw or ms/ch

1 = fixation durations

2 = all saccade lengths

3 = nonregressive saccades, before first regression

4 = probability of a forward saccade

10 = eyegaze

option: raw or ms/ch

11 = landing position

option 1: position of first fixation in a region

\*\*option 2: position of the second fixation in a region

\*\*option 3: launch site (position of previous saccade, charc back from start of region)

12 = position as function of time

Report fixation positions relative to

0 = start of region

1 = first fixation in region

Cumulate or don't cumulate position (cumulate = no going back; Scheepers)

Sample size in msec 10 or 20 or whatever

13 = time from entering region to first going PAST it

option: keep vs. discard trials with no appropriate fixations

option: raw or ms/ch

1 = All from first entering region to first going past

2 = Time spent in region from first entering to first going past

3 = Time spent outside of region after entering to first going past

14 = first fixation or saccade after LEAVING region (spillover)

option: eliminate the last fixation on a sentence

option: record fixation duration or saccade length of the first saccade

out of each region

option: permit only forward-going saccades or allow regressions as well

option: raw or ms/ch

15 = saccade matrix (from region to region)

1 = all saccades

2 = only saccades not following regression (first pass saccades)

16 = total gaze

option: raw or ms/ch

17 = landing site matrix (word length by position) for Jane Ashby

plus launch site vector (letters before start of word, for the

same person.

18 = inhibition of return analysis (report location of regressions and

the following forward saccades together with duration of preceding

fixations)

19 = vector of entering and leaving (for readaloud experiments)

20 = lengths of regions