NonlinearityAnalysis

Anonymous

15-1-2021

##Introduction

This notebook presents the code used to create a non-linearity analysis on the basis of keystroke data from the logger Inputlog, and accompanies the paper *Measuring non-linearity of long-term writing processes* (submitted to Reading & Writing special issue on methodology). For context, the abstract of our paper follows below.

Abstract

Non-linearity in writing provides important insight into the dynamics of writing and writing disfluencies. Currently, a range of linearity measures are available. These metrics are calculated based upon the leading edge, and are mostly used for short texts and single writing sessions. However, for longer, multi-session writing processes, the concept of the leading edge, as the singular outer boundary of the text-in-progress, is not enough to distinguish between linear production and non-linear text alterations. Therefore, in the current study, we propose a novel automatized non-linearity analysis. Within this approach, all backwards and forwards cursor and mouse operations from the point of utterance are extracted from keystroke data, and characterized both based on duration and distance. We illustrate this approach by analyzing the writing process of a complete novel based on more than 400 writing sessions totaling 276 hours of writing. The results show that this approach allows us to successfully cluster these writing sessions using the non-linearity characteristics.

Overview of the steps we have taken

We are using Inputlog-General Analysis files as our source material. You could apply the steps to the output of other keystroke loggers as well. The General Analysis is a table in which each keystroke action has its own row. It is quite similar to the raw output from other academic keystroke loggers.

Steps: 1. load all General Analysis files (one for each writing session) into R and merge them into one table, preserving the original session-numbering. (not including in this notebook) 2. aggregate/annotate the keystroke events in this table into larger segments of typing, deleting, jumping/navigating and focus (activities outside of the work doc). 3. Calculate characteristics for each jump event and each typing event 4. Create a summary table with descriptive statistics for each session 5. Running a correlation matrix, removing highly correlation variables, then performing a cluster analysis to explore similarities between sessions. This step is not included in this notebook.

Installation, use and requirements

You need the package Tidyverse to run this notebook.If you do not have it yet, you can use this code:

```
install.packages('tidyverse')
```

```
## Warning: package 'tidyverse' is in use and will not be installed
```

Load data

```
# this is a demo session. Our real data is not shared, due to privacy concerns and agreements with the p

# for example, create a new project in Rstudio, place csv-file in the same folder as this rmd-file.
all_data <- read.csv("demo_data_nonlinearity.csv", stringsAsFactors = F, fileEncoding = 'latin1')
```

Raw data sneakpeak

This is what our raw keystroke files look like. Each line is a keystroke event.

```
# for just the first five lines
head(all_data)
```

```
##   event_type__E_ event_id__E_ event_output__E_ event_charProduction__E_
## 1          mouse            0         Movement                        0
## 2       keyboard            1           LSHIFT                        1
## 3       keyboard            2                T                        1
## 4       keyboard            3                h                        2
## 5       keyboard            4                i                        3
## 6       keyboard            5                s                        4
##   event_actionTime__E_ event_startClock__E_ event_endClock__E_
## 1                  516         00:00:01.953       00:00:02.469
## 2                  225         00:00:03.265       00:00:03.490
## 3                   80         00:00:03.362       00:00:03.442
## 4                  152         00:00:03.634       00:00:03.786
## 5                   96         00:00:03.746       00:00:03.842
## 6                   87         00:00:03.843       00:00:03.930
##   event_startTime__E_ event_endTime__E_ event_pauseLocation__E_
## 1                1953              2469                       8
## 2                3265              3490                       8
## 3                3362              3442                       4
## 4                3634              3786                       1
## 5                3746              3842                       1
## 6                3843              3930                       1
##   event_pauseLocationFull__E_ event_pauseTime__E_ event_intervalNumber__E_
## 1                     INITIAL                1953                        1
## 2                     INITIAL                 796                        1
## 3           BEFORE SENTENCES                  97                        1
## 4               WITHIN WORDS                 272                        1
## 5               WITHIN WORDS                 112                        1
## 6               WITHIN WORDS                  97                        1
##   event_intervalSize__E_ event_x__E_ event_y__E_ event_positionFull__E_
## 1                      1         594         229                       0
## 2                      1          NA          NA                       0
```

```
## 3                      1          NA          NA                      0
## 4                      1          NA          NA                      1
## 5                      1          NA          NA                      2
## 6                      1          NA          NA                      3
##   event_doclengthFull__E_ event_position__E_ event_doclength__E_
## 1                       0                NA                   NA
## 2                       1                 0                    1
## 3                       1                 0                    1
## 4                       2                 1                    2
## 5                       3                 2                    3
## 6                       4                 3                    4
##   sessionID_LogCreationDate__S_ sessionID_Restricted_Logging__SL_
## 1            13-9-2021 15:54:00                                NA
## 2            13-9-2021 15:54:00                                NA
## 3            13-9-2021 15:54:00                                NA
## 4            13-9-2021 15:54:00                                NA
## 5            13-9-2021 15:54:00                                NA
## 6            13-9-2021 15:54:00                                NA
##   session_number
## 1              3
## 2              3
## 3              3
## 4              3
## 5              3
## 6              3
```

```
# for all of the data (fine for our sample session, may crash when you have large files)
view(all_data)
```

Add segmentation (step 2)

We start by labeling several groups of keyboard input. We define the category of arrow keys (as they are used to initiate non-linearity), and we distinguish between visible characters, whitespace, deletion keys and function keys (such as the Control key).

```
##########------------ Add info ----------------------------------------#######
arrow_keys <- c("UP", "DOWN", "LEFT", "RIGHT", "END", "HOME", "PAGE_DOWN",
                "PAGE_UP")

# Check all possible keystrokes
keys <- all_data %>% filter(event_type__E_ =="keyboard")
keyst <- data.frame(event_output__E_ = unique(keys$event_output__E_),
                    stringsAsFactors = F)

keyst_add <- keyst %>%
  mutate(length = nchar(as.character(event_output__E_)),
         type = ifelse((length < 3 & event_output__E_ != "UP") |
                         grepl("OEM_", event_output__E_), "visible_char",
                ifelse(event_output__E_ %in% c("SPACE", "RETURN", "TAB"),
                       "whitespace",
                ifelse(grepl("UP|DOWN|LEFT|RIGHT|END|HOME", event_output__E_),
                       "arrow_key",
                ifelse(event_output__E_ %in% c("BACK", "DELETE"),
```

```
                        "delete_key",
                    "function_key")))))
```

Then, we add boundaries between non-linear jumps and the other types of events. In the #notes in the code below, you can see how we distinguish between 6 different situations in which we are placing a boundary.

```
# Add boundaries for non-linearity
data_add <- all_data %>%
  # Remove keystrokes outside doc (e.g., save as XXX)
  filter(!(event_type__E_ == "keyboard" & is.na(event_position__E_))) %>%

  left_join(keyst_add) %>%
  # Calculate per session/file separately
  group_by(session_number) %>%
  mutate(jump_start = ifelse(
    row_number() == 1 |
    #1) When a typist moves from typing a character to a mouse event
    #   (click, movement, scroll, selection), or vice versa.
    (event_type__E_ == "keyboard" &
       (type %in% c("visible_char", "whitespace") |
          event_output__E_ == "CAPS LOCK" |
                    (type == "function_key" &
                     event_pauseLocationFull__E_ == "COMBINATION KEY"))  &
      (lag(event_type__E_) == "mouse" | event_type__E_ == "replacement")) |
      ((event_type__E_ == "mouse" | lag(event_type__E_) == "replacement")  &
      lag(event_type__E_) == "keyboard" &
         (lag(type) %in% c("visible_char", "whitespace")) |
        event_output__E_ == "CAPS LOCK") |
    #2) When a typists moves from an insertion to another event, or vice versa.
    (lag(event_type__E_) %in% c("insert") &
       event_type__E_ != lag(event_type__E_) |
      (event_type__E_ %in% c("insert")  &
         lag(event_type__E_) != event_type__E_)) |
    #3) When a typist moves from typing a character to typing an arrow key,
    #   or vice versa.
      (event_type__E_ == "keyboard" &
         (type %in% c("visible_char", "whitespace") |
            event_output__E_ == "CAPS LOCK" |
            (type == "function_key" &
               event_pauseLocationFull__E_ == "COMBINATION KEY")) &
         lag(event_type__E_) == "keyboard" & lag(type) == "arrow_key") |
      (event_type__E_ == "keyboard" & type == "arrow_key"
       & (lag(event_type__E_) == "keyboard" &
         lag(type) %in% c("visible_char", "whitespace"))  |
         lag(event_type__E_) == "replacement")  |
    #4) When a typists moves from a keystroke or mouse event to a
    #   delete/backspace keypress, or vice versa.
      (event_type__E_ %in% c("mouse","keyboard", "insert") &
         (is.na(type) | type != "delete_key") &
         lag(event_type__E_) == "keyboard" & lag(type) == "delete_key") |
      (event_type__E_ == "keyboard" & type == "delete_key"
       & lag(event_type__E_) %in% c("mouse","keyboard", "insert") &
         !lag(event_output__E_) %in% c("DELETE", "BACK")) |
```

```
    #5) When a typist moves from one mode of deletion to another (e.g., from
    #   delete key to backspace key press).
      (type == "delete_key" & lag(type) == "delete_key" &
      (event_output__E_) != lag(event_output__E_)) |
    #6) When a typist moves from the main text to a different source
    #   (e.g., online dictionary)
      (lag(event_type__E_) %in% c("focus") &
         event_type__E_ != lag(event_type__E_)) |
         (event_type__E_ %in% c("focus") &
            event_type__E_ != lead(event_type__E_)) , 1, 0),

    #Part B
 # Set to zero if selection is directly followed by insert/delete.
    # (A series of deletions count as one event.)
    #delete - replacement
    jump_start = ifelse((event_type__E_ == "replacement" &
                           lag(event_output__E_) == "DELETE" &
                           event_startClock__E_ == lag(event_startClock__E_) &
                           event_endClock__E_ == lag(event_endClock__E_))
                         | (lag(event_type__E_) == "replacement" &
                              event_output__E_ == "DELETE" &
                                lead(event_type__E_) == "replacement"  &
                          lag(event_output__E_,2) == "DELETE" ) |
                          (event_type__E_ == "replacement" &
                          lag(event_type__E_) == "replacement") |
                          is.na(jump_start), 0, jump_start),
    # Create count number of linear event
    jump_number = ifelse(jump_start == 1,
                         cumsum(jump_start == 1 |
                                  row_number() == 1), NA),
      prev_loc = lag(event_pauseLocationFull__E_),
      next_loc = lead(event_pauseLocationFull__E_),
      endTime_session = max(event_endTime__E_)
    ) %>%
  fill(jump_number)
```

```
## Joining, by = "event_output__E_"
```

```
## Replacing 0 with NA for further processing
data_add %>%
mutate(event_charProduction__E_= ifelse(event_charProduction__E_ ==0, NA, event_charProduction__E_))
```

```
## # A tibble: 728 x 30
## # Groups:   session_number [2]
##    event_type__E_ event_id__E_ event_output__E_ event_charProduction__E_
##    <chr>                 <int> <chr>                                <int>
##  1 mouse                     0 Movement                                NA
##  2 keyboard                  1 LSHIFT                                   1
##  3 keyboard                  2 T                                        1
##  4 keyboard                  3 h                                        2
##  5 keyboard                  4 i                                        3
##  6 keyboard                  5 s                                        4
##  7 keyboard                  6 SPACE                                    5
```

```
##  8 keyboard                  7 i                                          6
##  9 keyboard                  8 s                                          7
## 10 keyboard                  9 SPACE                                      8
## # ... with 718 more rows, and 26 more variables: event_actionTime__E_ <int>,
## #   event_startClock__E_ <chr>, event_endClock__E_ <chr>,
## #   event_startTime__E_ <int>, event_endTime__E_ <int>,
## #   event_pauseLocation__E_ <int>, event_pauseLocationFull__E_ <chr>,
## #   event_pauseTime__E_ <int>, event_intervalNumber__E_ <int>,
## #   event_intervalSize__E_ <int>, event_x__E_ <int>, event_y__E_ <int>,
## #   event_positionFull__E_ <int>, event_doclengthFull__E_ <int>,
## #   event_position__E_ <int>, event_doclength__E_ <int>,
## #   sessionID_LogCreationDate__S_ <chr>,
## #   sessionID_Restricted_Logging__SL_ <lgl>, session_number <int>,
## #   length <int>, type <chr>, jump_start <dbl>, jump_number <int>,
## #   prev_loc <chr>, next_loc <chr>, endTime_session <int>
```

In part B (in the code block above), we add a specific rule to aggregate deletions of multiple characters that are done by selecting each character separately - this is something which occurred quite a lot for one particular writer, it may not be necessary for other materials.Also, we add an identification number for each segment (note that although the code calls everything 'jump' here it is actually all segments in between the boundaries we set before.)

preview of segmented data

```
head(data_add)
```

```
## # A tibble: 6 x 30
## # Groups:   session_number [1]
##   event_type__E_ event_id__E_ event_output__E_ event_charProdu~ event_actionTim~
##   <chr>                <int> <chr>                        <int>            <int>
## 1 mouse                    0 Movement                         0              516
## 2 keyboard                 1 LSHIFT                           1              225
## 3 keyboard                 2 T                                1               80
## 4 keyboard                 3 h                                2              152
## 5 keyboard                 4 i                                3               96
## 6 keyboard                 5 s                                4               87
## # ... with 25 more variables: event_startClock__E_ <chr>,
## #   event_endClock__E_ <chr>, event_startTime__E_ <int>,
## #   event_endTime__E_ <int>, event_pauseLocation__E_ <int>,
## #   event_pauseLocationFull__E_ <chr>, event_pauseTime__E_ <int>,
## #   event_intervalNumber__E_ <int>, event_intervalSize__E_ <int>,
## #   event_x__E_ <int>, event_y__E_ <int>, event_positionFull__E_ <int>,
## #   event_doclengthFull__E_ <int>, event_position__E_ <int>,
## #   event_doclength__E_ <int>, sessionID_LogCreationDate__S_ <chr>,
## #   sessionID_Restricted_Logging__SL_ <lgl>, session_number <int>,
## #   length <int>, type <chr>, jump_start <dbl>, jump_number <int>,
## #   prev_loc <chr>, next_loc <chr>, endTime_session <int>
```

```
# Or for the full table
# view(data_add)
```

Now we will label each segment - adding a column with their type (delete, focus, jumps, typing et cetera). This enables us to later calculate certain values for only the jumps or only the typing chunks, for example.

Add characteristics for each jump (step 3)

For a description of the characteristics, please see our paper.

Add characteristics for each typing segment (step 3B)

Session-level summary table (step 4)

```
# Summary statistics for each session (all, time and word count)
sum_session <- data_add %>%
  group_by(session_number)%>%
  summarize(
    total_time = last(event_endTime__E_) - first(event_startTime__E_),
    total_time_seconds = total_time/1000,
    total_time_minutes = total_time_seconds/60,
    total_charproduced = last(event_charProduction__E_) - first(event_charProduction__E_)
  ) %>%
  mutate(
    char_produced2 = total_charproduced - lag(total_charproduced))


#added total jump time per session
sum_session2 <- jump_filt %>%
  group_by(session_number)%>%
  summarize(
    Jump_time = sum(jump_duration, na.rm = T)
  ) %>% left_join(sum_session)
```

```
## Joining, by = "session_number"
```

```
#view(sum_session2)

## Added TYPING chunks (size & duration & position relative to leading edge, also total amount of chars
sum_session3 <- sum_typing %>%
  group_by(session_number)%>%
  summarise(
    MeanTypingChars = mean(typing_size_chars),
    sdTypingChars = sd(typing_size_chars),
    totalTypingChars = sum(typing_size_chars, na.rm = T),
    MeanDurationTyping = mean(typing_duration),
    sdDurationTyping = sd(typing_duration),
    MeanTypingPositionRel = mean(start_position_rel),
    sdTypingPositionRel = sd(start_position_rel),
    MeanTypingPositionEdge = mean(start_position_edge),
    sdTypingPositionEdge = sd(start_position_edge),
    totalTypingChars = sum(typing_size_chars)
  ) %>% left_join(sum_session2)
```

```
## Joining, by = "session_number"
```

Next, we are adding information on where in the text a jump starts - for example: within a word, or after a sentence -. We then calculate the % of jumps that fall into each location-category for each session.

```
pivottableStartLoc <- count(jump_filt, start_location)

# Pivot_wider helps to glue the two tables together at the appropriate junctions
pivotwide <- pivot_wider(pivottableStartLoc, names_from = start_location, values_from = n)
pivotfancy <- pivotwide %>%
   rename(
      StartPos_AfterWords = `AFTER WORDS`,
     # StartPos_BeforeParagraphs = `BEFORE PARAGRAPHS`,
     # StartPos_BeforeWords = `BEFORE WORDS`,
   #   StartPos_Change = CHANGE,
      StartPos_Deletion = REVISION,
    #  StartPos_WithinWords = `WITHIN WORDS`,
      StartPos_BeforeSentences = `BEFORE SENTENCES`,
      StartPos_AfterSentences = `AFTER SENTENCES`)
   #   StartPos_unknown = UNKNOWN)

   ## a number of categories do not occur in our sample sessions. You can remove the hashtag to include


# Adding count of instances within each category



pivotfancy2 <- pivotfancy %>%
   mutate(N = sum(StartPos_AfterWords,
                  # StartPos_BeforeParagraphs, StartPos_BeforeWords, StartPos_Change,
# StartPos_WithinWords, StartPos_unknown,
                  StartPos_Deletion,  StartPos_BeforeSentences,
                  StartPos_AfterSentences, na.rm = T))

# And changing raw counts to percentages ##
pivotfancy3 <- pivotfancy2 %>%
   mutate( StartPos_AfterWords_perc = 100/(N/StartPos_AfterWords),
          # StartPos_BeforeParagraphs_perc = 100/(N/StartPos_BeforeParagraphs),
         #  StartPos_BeforeWords_perc = 100/(N/StartPos_BeforeWords),
         #  StartPos_Change_perc = 100/(N/StartPos_Change),
           StartPos_Deletion_perc = 100/(N/StartPos_Deletion),
        #   StartPos_WithinWords_perc = 100/(N/StartPos_WithinWords),
           StartPos_BeforeSentences_perc = 100/(N/StartPos_BeforeSentences),
           StartPos_AfterSentences_perc = 100/(N/StartPos_AfterSentences))


## Removing unnecessary columns ##
pivotfancy31 <- pivotfancy3 %>%
   select(session_number, StartPos_AfterWords_perc,
        #  StartPos_BeforeParagraphs_perc,
        #  StartPos_BeforeWords_perc,
        #  StartPos_Change_perc,
          StartPos_Deletion_perc,
```

```
        #  StartPos_WithinWords_perc,
          StartPos_BeforeSentences_perc,
          StartPos_AfterSentences_perc)

## Replacing missing values by zero (0% instead of NA)
pivotfancy4 <- mutate_at(pivotfancy31, vars(StartPos_AfterWords_perc:StartPos_AfterSentences_perc), ~rep
```

```
sum_session4 <- sum_session3 %>%
  group_by(session_number)%>%
  mutate( Perc_time_jumps = ifelse(sum(Jump_time > 0) == 0, 0,
                             100/( total_time / Jump_time))) %>%

  left_join(pivotfancy4)
```

```
## Joining, by = "session_number"
```

And now on to the most important bit - adding descriptive writing session statistics from the jump characteristics.

```
descriptives <- jump_filt %>%
  group_by(session_number)%>%
      summarize(
    # Time-based
    MeanDurationJumps = mean(jump_duration, na.rm = TRUE),
    SD_DurationJumps = sd(jump_duration, na.rm = TRUE),

    # Position change
    MeanJumpsize_chars = mean(jump_size_chars, na.rm = T),
    SDJumpsize_chars = sd(jump_size_chars, na.rm = T),
    MeanJumpsize_rel = mean(jump_size_rel, na.rm = T),
    SDJumpsize_rel = sd(jump_size_rel, na.rm = T),

    # All distances transformed to positive values
    MeanJumpsize_chars_Plus = mean(jump_size_charsPlus, na.rm = T),
    sdJumpsize_chars_Plus = sd(jump_size_charsPlus, na.rm = T),

    # Log transform of MeanJumpsize
    logmeanJumpsize_chars_Plus = log(MeanJumpsize_chars_Plus),

    # Total jumpsize
    TotalJumpsize =sum(jump_size_charsPlus),
    logTotalJumpsize = log(TotalJumpsize),

    # Size of backwards jumps
    MeanJumpSize_BackW = mean(jump_size_chars[jump_size_chars < 0],
                              na.rm = TRUE),
    SDJumpSize_BackW = sd(jump_size_chars[jump_size_chars < 0],
                          na.rm = TRUE),

    # Size of forwards jumps
    MeanJumpSize_Forw = mean(jump_size_chars[jump_size_chars > 0],
                          na.rm = TRUE),
```

```
    SDJumpSize_Forw = sd(jump_size_chars[jump_size_chars > 0],
                         na.rm = TRUE),

    # Percentile of jumps that is backwards
    Countrows = n(),
  PercentageBackwardsJumps = ifelse( sum(jump_size_chars < 0) == 0,0,
                                100/(Countrows / sum(jump_size_chars < 0))),


    # Slope = jump size in chars / duration
  MeanJumpSlope = mean(jump_slope),
  SDJumpSlope = sd(jump_slope),

  # Start position relative to the leading edge (in percentile )
  MeanStartPos_rel = mean(start_position_rel),
  sdStartPos_rel = sd(start_position_rel),

  # Start position in characters from leading edge
  MeanStartPos_edge = mean(start_position_edge),
  sdStartPos_edge = sd(start_position_edge),

  # End position relative to the leading edge (in percentile)
  MeanEndPos_rel = mean(end_position_rel),
  sdEndPos_rel = sd(end_position_rel),

  # Content of jumps
  Mean_n_events = mean(n_events),
  sd_n_events = sd(n_events),
  Mean_n_scroll_movements = mean(n_scroll_movements),
  sd_n_scroll_movements = sd(n_scroll_movements))
```

Sneakpeak at the output from the previous code block

```
# Viewing the entire table
view(descriptives)

## Overview of the added variables
colnames(descriptives)
```

```
##  [1] "session_number"             "MeanDurationJumps"
##  [3] "SD_DurationJumps"           "MeanJumpsize_chars"
##  [5] "SDJumpsize_chars"           "MeanJumpsize_rel"
##  [7] "SDJumpsize_rel"             "MeanJumpsize_chars_Plus"
##  [9] "sdJumpsize_chars_Plus"      "logmeanJumpsize_chars_Plus"
## [11] "TotalJumpsize"              "logTotalJumpsize"
## [13] "MeanJumpSize_BackW"         "SDJumpSize_BackW"
## [15] "MeanJumpSize_Forw"          "SDJumpSize_Forw"
## [17] "Countrows"                  "PercentageBackwardsJumps"
## [19] "MeanJumpSlope"              "SDJumpSlope"
## [21] "MeanStartPos_rel"           "sdStartPos_rel"
```

```
## [23] "MeanStartPos_edge"          "sdStartPos_edge"
## [25] "MeanEndPos_rel"             "sdEndPos_rel"
## [27] "Mean_n_events"              "sd_n_events"
## [29] "Mean_n_scroll_movements"    "sd_n_scroll_movements"
```

Merging two tables and adding a few other relative variables using information from one of the generic tables.

```
# Merging two tables
descriptivesBig <- sum_session4 %>%
  group_by(session_number) %>%
left_join(descriptives)
```

```
## Joining, by = "session_number"
```

```
# Relative jump count added
descriptivesBigger <- descriptivesBig %>%
  group_by(session_number) %>%
  mutate(RelCountJumps = totalTypingChars/Countrows)


# Detour added, it's a ratio (total jumpsize /characters typed)
descriptivesFinal <- descriptivesBigger %>%
   group_by(session_number) %>%
   mutate(Detour = TotalJumpsize/char_produced2, Author = "GB")
```

inspecting the output table

```
view(descriptivesFinal)

# and to obtain a first impression of the dataset
summary(descriptivesFinal)
```

```
##  session_number MeanTypingChars sdTypingChars   totalTypingChars
##  Min.   :3.00   Min.   :11.00   Min.   :10.61   Min.   :154.0
##  1st Qu.:3.25   1st Qu.:11.01   1st Qu.:10.84   1st Qu.:162.5
##  Median :3.50   Median :11.03   Median :11.07   Median :171.0
##  Mean   :3.50   Mean   :11.03   Mean   :11.07   Mean   :171.0
##  3rd Qu.:3.75   3rd Qu.:11.04   3rd Qu.:11.30   3rd Qu.:179.5
##  Max.   :4.00   Max.   :11.06   Max.   :11.54   Max.   :188.0
##
##  MeanDurationTyping sdDurationTyping MeanTypingPositionRel sdTypingPositionRel
##  Min.   :2066       Min.   :2109     Min.   :0.7573        Min.   :0.3145
##  1st Qu.:2193       1st Qu.:2281     1st Qu.:0.7573        1st Qu.:0.3145
##  Median :2321       Median :2453     Median :0.7573        Median :0.3145
##  Mean   :2321       Mean   :2453     Mean   :0.7573        Mean   :0.3145
##  3rd Qu.:2448       3rd Qu.:2626     3rd Qu.:0.7573        3rd Qu.:0.3145
##  Max.   :2575       Max.   :2798     Max.   :0.7573        Max.   :0.3145
##                                      NA's   :1             NA's   :1
##  MeanTypingPositionEdge sdTypingPositionEdge   Jump_time      total_time
##  Min.   :27.64          Min.   :34.91        Min.   :3186   Min.   : 93281
```

```
##  1st Qu.:29.32          1st Qu.:37.16        1st Qu.:3545   1st Qu.: 98289
##  Median :31.00          Median :39.41        Median :3904   Median :103297
##  Mean   :31.00          Mean   :39.41        Mean   :3904   Mean   :103297
##  3rd Qu.:32.68          3rd Qu.:41.66        3rd Qu.:4262   3rd Qu.:108305
##  Max.   :34.35          Max.   :43.91        Max.   :4621   Max.   :113313
##
##  total_time_seconds total_time_minutes total_charproduced char_produced2
##  Min.   : 93.28     Min.   :1.555      Min.   :199.0      Min.   :41
##  1st Qu.: 98.29     1st Qu.:1.638      1st Qu.:209.2      1st Qu.:41
##  Median :103.30     Median :1.722      Median :219.5      Median :41
##  Mean   :103.30     Mean   :1.722      Mean   :219.5      Mean   :41
##  3rd Qu.:108.31     3rd Qu.:1.805      3rd Qu.:229.8      3rd Qu.:41
##  Max.   :113.31     Max.   :1.889      Max.   :240.0      Max.   :41
##                                                           NA's   :1
##  Perc_time_jumps StartPos_AfterWords_perc StartPos_Deletion_perc
##  Min.   :2.812   Min.   : 0.0             Min.   : 0.0
##  1st Qu.:3.347   1st Qu.:12.5             1st Qu.:12.5
##  Median :3.883   Median :25.0             Median :25.0
##  Mean   :3.883   Mean   :25.0             Mean   :25.0
##  3rd Qu.:4.418   3rd Qu.:37.5             3rd Qu.:37.5
##  Max.   :4.954   Max.   :50.0             Max.   :50.0
##
##  StartPos_BeforeSentences_perc StartPos_AfterSentences_perc MeanDurationJumps
##  Min.   : 0.0                  Min.   : 0.0                 Min.   :1593
##  1st Qu.:12.5                  1st Qu.:12.5                 1st Qu.:1772
##  Median :25.0                  Median :25.0                 Median :1952
##  Mean   :25.0                  Mean   :25.0                 Mean   :1952
##  3rd Qu.:37.5                  3rd Qu.:37.5                 3rd Qu.:2131
##  Max.   :50.0                  Max.   :50.0                 Max.   :2310
##
##  SD_DurationJumps MeanJumpsize_chars SDJumpsize_chars MeanJumpsize_rel
##  Min.   :1570     Min.   :-36.0      Min.   : 9.899   Min.   :0.05893
##  1st Qu.:1670     1st Qu.:-28.5      1st Qu.:22.627   1st Qu.:0.14527
##  Median :1769     Median :-21.0      Median :35.355   Median :0.23161
##  Mean   :1769     Mean   :-21.0      Mean   :35.355   Mean   :0.23161
##  3rd Qu.:1868     3rd Qu.:-13.5      3rd Qu.:48.083   3rd Qu.:0.31794
##  Max.   :1967     Max.   : -6.0      Max.   :60.811   Max.   :0.40428
##
##  SDJumpsize_rel    MeanJumpsize_chars_Plus sdJumpsize_chars_Plus
##  Min.   :0.06987   Min.   : 7              Min.   : 8.485
##  1st Qu.:0.17803   1st Qu.:16              1st Qu.:19.092
##  Median :0.28619   Median :25              Median :29.698
##  Mean   :0.28619   Mean   :25              Mean   :29.698
##  3rd Qu.:0.39435   3rd Qu.:34              3rd Qu.:40.305
##  Max.   :0.50252   Max.   :43              Max.   :50.912
##
##  logmeanJumpsize_chars_Plus TotalJumpsize logTotalJumpsize MeanJumpSize_BackW
##  Min.   :1.946              Min.   :14    Min.   :2.639    Min.   :-79.0
##  1st Qu.:2.400              1st Qu.:32    1st Qu.:3.093    1st Qu.:-62.5
##  Median :2.854              Median :50    Median :3.547    Median :-46.0
##  Mean   :2.854              Mean   :50    Mean   :3.547    Mean   :-46.0
##  3rd Qu.:3.307              3rd Qu.:68    3rd Qu.:4.001    3rd Qu.:-29.5
##  Max.   :3.761              Max.   :86    Max.   :4.454    Max.   :-13.0
##
```

```
##  SDJumpSize_BackW MeanJumpSize_Forw SDJumpSize_Forw   Countrows
##  Min.   : NA      Min.   :1.0       Min.   : NA     Min.   :2
##  1st Qu.: NA      1st Qu.:2.5       1st Qu.: NA     1st Qu.:2
##  Median : NA      Median :4.0       Median : NA     Median :2
##  Mean   :NaN      Mean   :4.0       Mean   :NaN     Mean   :2
##  3rd Qu.: NA      3rd Qu.:5.5       3rd Qu.: NA     3rd Qu.:2
##  Max.   : NA      Max.   :7.0       Max.   : NA     Max.   :2
##  NA's   :2                          NA's   :2
##  PercentageBackwardsJumps MeanJumpSlope        SDJumpSlope
##  Min.   :50               Min.   :-0.008630   Min.   :0.006581
##  1st Qu.:50               1st Qu.:-0.006398   1st Qu.:0.010049
##  Median :50               Median :-0.004166   Median :0.013517
##  Mean   :50               Mean   :-0.004166   Mean   :0.013517
##  3rd Qu.:50               3rd Qu.:-0.001935   3rd Qu.:0.016986
##  Max.   :50               Max.   : 0.000297   Max.   :0.020454
##
##  MeanStartPos_rel sdStartPos_rel   MeanStartPos_edge sdStartPos_edge
##  Min.   :0.5863   Min.   :0.3294   Min.   :34.50     Min.   :47.38
##  1st Qu.:0.6291   1st Qu.:0.3904   1st Qu.:36.75     1st Qu.:50.56
##  Median :0.6719   Median :0.4514   Median :39.00     Median :53.74
##  Mean   :0.6719   Mean   :0.4514   Mean   :39.00     Mean   :53.74
##  3rd Qu.:0.7146   3rd Qu.:0.5123   3rd Qu.:41.25     3rd Qu.:56.92
##  Max.   :0.7574   Max.   :0.5733   Max.   :43.50     Max.   :60.10
##
##  MeanEndPos_rel    sdEndPos_rel    Mean_n_events    sd_n_events
##  Min.   :0.4010   Min.   :0.2438   Min.   :3.500   Min.   :1.414
##  1st Qu.:0.4359   1st Qu.:0.3067   1st Qu.:4.375   1st Qu.:1.591
##  Median :0.4708   Median :0.3695   Median :5.250   Median :1.768
##  Mean   :0.4708   Mean   :0.3695   Mean   :5.250   Mean   :1.768
##  3rd Qu.:0.5057   3rd Qu.:0.4323   3rd Qu.:6.125   3rd Qu.:1.945
##  Max.   :0.5406   Max.   :0.4952   Max.   :7.000   Max.   :2.121
##
##  Mean_n_scroll_movements sd_n_scroll_movements RelCountJumps       Detour
##  Min.   :0               Min.   :0             Min.   :77.00   Min.   :0.3415
##  1st Qu.:0               1st Qu.:0             1st Qu.:81.25   1st Qu.:0.3415
##  Median :0               Median :0             Median :85.50   Median :0.3415
##  Mean   :0               Mean   :0             Mean   :85.50   Mean   :0.3415
##  3rd Qu.:0               3rd Qu.:0             3rd Qu.:89.75   3rd Qu.:0.3415
##  Max.   :0               Max.   :0             Max.   :94.00   Max.   :0.3415
##                                                                NA's   :1
##     Author
##  Length:2
##  Class :character
##  Mode  :character
##
##
##
##
```

Final remarks

We hope to have shown the ingredients of our analysis and the steps we took to demarcate keystroke logging files into non-linear jumps, texts bursts, focus and deletion events, followed by the application of descriptive

statistics at the writing session-level.