Working with Data Workshop

April 26, 2021

Jamie Yap and Lindsey Potter
jamieyap@umich.edu, Lindsey.Potter@hci.utah.edu

Module: Getting Started

April 26, 2021

Contents

GOAL: By the end of this module, you will have set up your computing environment with the appropriate software versions. Additionally, you would also be able to read and view the curated data files using `R`.

CODE FOR THIS MODULE: `module-getting-started.R`.

1 Set Up of Software

Let's load the two packages we will use throughout: `dplyr` for data manipulation and `lme4` for estimating generalized linear mixed models (GLMMs) with longitudinal data.

```
library(dplyr)
library(lme4)
```

We note that we are using the following combination of versions of software for Module 1: `R 4.0.4`, `dplyr 1.0.5`, `lme4 1.1-25`. The code examples presented in the current and succeeding modules in this workshop may not display output as expected when using older versions of these software. In particular, examples reading csv files using versions of `R` prior to `4.0` or using versions of `lme4` prior to `1.0` will not result in output identical to what will be shown in this module.

Specific versions of `R` itself can be downloaded from the following URLs, depending on your operating system:

- Windows
  - https://cran.r-project.org/bin/windows/base/ for newer versions of `R`
  - https://cran.r-project.org/bin/windows/base/old/ for older versions of `R`
- MACOSX
  - https://cran.r-project.org/bin/macosx/base/ for newer versions of `R`
  - https://cran.r-project.org/bin/macosx/old/ for older versions of `R`

CAUTION: Check the particular version of `R` you are working with prior to starting any analysis. If you are not working with the required version, you may switch versions through the `Tools -> Global Options -> General` tab if you are using R Studio.

You may already have some version of the `dplyr` or `lme4` package installed. However, if you would like to use the specific versions used in this workshop, one of the simplest ways is to first uninstall any existing version you may have using the `remove.packages` function from the `utils` package, and then install specific versions of `R` packages through the `install_version()` function of the `devtools` package, like so:

```
utils::remove.packages("dplyr")
utils::remove.packages("lme4")

devtools::install_version("dplyr", version = "1.0.5", repos = "http://cran.us.r-project.org")
devtools::install_version("lme4", version = "1.1-25", repos = "http://cran.us.r-project.org")
```

As check, you may run `sessionInfo()`. Displayed below is the combination of software versions used for the modules in this workshop.

```
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
## [1] lme4_1.1-25  Matrix_1.3-2 dplyr_1.0.5
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6        pillar_1.5.1      compiler_4.0.4    formatR_1.8
##  [5] nloptr_1.2.2.2    tools_4.0.4       boot_1.3-27       digest_0.6.27
##  [9] statmod_1.4.35    evaluate_0.14     lifecycle_1.0.0   tibble_3.1.0
## [13] nlme_3.1-152      lattice_0.20-41   pkgconfig_2.0.3   rlang_0.4.10
## [17] yaml_2.2.1        xfun_0.22         stringr_1.4.0     knitr_1.31
## [21] generics_0.1.0    vctrs_0.3.7       grid_4.0.4        tidyselect_1.1.0
## [25] glue_1.4.2        R6_2.5.0          fansi_0.4.2       rmarkdown_2.7
## [29] minqa_1.2.4       purrr_0.3.4       magrittr_2.0.1    ellipsis_0.3.1
## [33] htmltools_0.5.1.1 splines_4.0.4     MASS_7.3-53.1     renv_0.13.1
## [37] utf8_1.2.1        stringi_1.5.3     crayon_1.4.1
```

2 Set Up of Locations to Store Data Files and Output

We specify the location of input files and output files in the variables `path_pns_input_data` and `path_pns_output_data`, respectively.

```
path_pns_input_data <- "C:/Users/jamieyap/Desktop/input_data"
path_pns_output_data <- "C:/Users/jamieyap/Desktop/output_data"
```

3 Reading in Data Files

Let's now read in the data files we will be using in Module 1. The following are noteworthy:

- These data files should be within the location specified in `path_pns_input_data`.
- We set the value of `na.strings` to be a blank character, which tells `R` to treat such characters in a csv file as missing values. This is necessary since, by default, `R` treats cells coded as `NA` as missing values, i.e., by default, `na.strings = "NA"`, but all missing values in the curated data files are coded as blanks.

```
dat_big_merged <- read.csv(file.path(path_pns_input_data, "merged.csv"), header = TRUE,
    na.strings = "")
dat_quit_dates <- read.csv(file.path(path_pns_input_data, "quit_dates_final.csv"),
    header = TRUE, na.strings = "")
ema_item_names <- read.csv(file.path(path_pns_input_data, "ema_item_names.csv"),
    header = TRUE, na.strings = "")
```

As a check for whether we have been able to read the files successfully, let's view the first few rows of `ema_item_names`.

```
head(ema_item_names, n = 10)
```

```
##    is_postquit_assessment_type           assessment_type       name_codebook
## 1                            1 Post-Quit Already Slipped  Consume_PastTense1
## 2                            1 Post-Quit Already Slipped  Consume_PastTense2
## 3                            1 Post-Quit Already Slipped          LocationSm
## 4                            1 Post-Quit Already Slipped       SS_PastTense1
## 5                            1 Post-Quit Already Slipped       SS_PastTense2
## 6                            1 Post-Quit Already Slipped       SS_PastTense3
## 7                            1 Post-Quit Already Slipped       SS_PastTense4
## 8                            1 Post-Quit Already Slipped AbSelfEff_PastTense
## 9                            1 Post-Quit Already Slipped   Affect_PastTense1
## 10                           1 Post-Quit Already Slipped  Affect_PastTense10
##                          name_new
## 1   postquit_alreadyslipped_item_1
## 2   postquit_alreadyslipped_item_2
## 3   postquit_alreadyslipped_item_3
## 4   postquit_alreadyslipped_item_4
## 5   postquit_alreadyslipped_item_5
## 6   postquit_alreadyslipped_item_6
## 7   postquit_alreadyslipped_item_7
## 8   postquit_alreadyslipped_item_8
## 9   postquit_alreadyslipped_item_9
## 10 postquit_alreadyslipped_item_10
```

Workshop Kick Off

April 26, 2021

Contents

1 Workshop Goal

The PNS documentation (e.g., see https://github.com/jamieyap/PNS) contains information on

- Study design
- How the data collected differed from the intended design
- How such deviations were addressed in the process of constructing the curated datasets

More than a mere collection of facts, the documentation seeks to organize these diverse set of information into a coherent story, allowing end-users of the curated datasets to more quickly...

1. See to what extent is it feasible to investigate a particular scientific question using the data on-hand
2. See to what extent is it feasible to consider potential advancements in computational methodology

The workshop seeks to complement the PNS documentation. By the end of this workshop, you will be able to:

- Correctly interpret columns in the curated datasets
- Correctly identify rows which need to be included/excluded when working on (1) and/or (2) described above

2 Ground Rules

- Questions and lively discussion are encouraged. However, we ask that workshop participants hold off until 'break points' indicated by the following box in your notes. There are no *dumb questions* in this workshop – we encourage you to raise questions during these break points for the benefit of other workshop participants as well.

BREAK: Any questions?

- Have a pen and paper on hand and get ready to doodle together as we progress through the modules.
- During live demos of R code (Modules 3 and 5), workshop participants are encouraged to use break points to initiate discussion focusing on the logic, rather than syntax, employed in the code.

3 What to Expect

WORKSHOP KICK OFF

- MODULE 1: Anatomy of `merged.csv`
- MODULE 2: Set-Up of Independent Variables
- MODULE 3: Laying the Groundwork for Set-Up of Dependent Variables through Data Visualization

COFFEE/TEA/BAGEL BREAK: 15 minutes

- MODULE 4: Set-Up of Dependent Variable
- MODULE 5: Modeling and Hypothesis Testing

COFFEE/TEA/BAGEL BREAK: 7 minutes

WORKSHOP WRAP UP

Module 1: Anatomy of `merged.csv`

April 26, 2021

Contents

MODULE 1 GOAL: By the end of this module, you will be able to:

- Identify which rows and columns in the `merged.csv` file to select when one wishes to use responses from Pre-/Post- Quit Random EMA Questionnaires as Independent Variables (IV's)
- Identify which rows and columns in the `merged.csv` file to select when cigarette smoking behavior is the primary Dependent Variable (DV) of interest.

1 What do the rows represent?

Each row of `merged.csv` represents a successfully launched EMA questionnaire during the study...

Draw Figure Here.

...and we can tell

- what kind of EMA questionnaire was launched, i.e., what kind of survey was 'pushed' by the smartphone to the participant (`assessment_type`)

- when the EMA questionnaire was launched (`delivered_unixts`)
- whether the participant responded *any* item in the EMA questionnaire (`with_any_response`)
- if the participant responded to *any* item in the EMA questionnaire, when they began providing their responses (`begin_unixts`)

Let's unpack the following:

- What do we mean when we say **successfully launched?**
- What do we mean when we say **having any response?**
- How do we know **which EMA came first?**

Successfully launched?

We say '*successfully* launched' to emphasize that there may be prolonged periods of time when no EMA questionnaires of any type were launched (e.g., due to software issue or when phone was switched off). Hence, during such periods of time, no EMA questionnaires will be displayed on the participant's smartphone. Correspondingly, when performing a visual inspection of the data, such periods of time may be represented by a prolonged time gap between two consecutive EMA questionnaires in the `merged.csv` file.

Having any response?

We say 'having *any* response' when speaking of the variable `with_any-response` since it is possible for participants to partially complete EMA Questionnaires. Hence, fully completed and partially completed EMA Questionnaires will have `with_any_response=1` but EMA Questionnaires having no response whatsoever to all items will have `with_any_response=0`.

Additionally, we note that although it may appear unnecessary to have an indicator for whether the participant responded to Self-Initiated EMA Questionnaires (i.e., the variable `with_any_response` was constructed for all types of EMA Questionnaires), it is possible (but uncommon) for a participant to eventually decide to ignore Self-Initiated EMA Questionnaires (i.e., they perform a button press, and the button press was followed by the launching of an EMA Questionnaire, but for whatever reason, the participant decided to not follow through with responding to the EMA Questionnaire).

Which EMA came first?

We also need a way to chronologically order EMAs. While completed and partially completed EMAs may have a timestamp for Begin Time (`begin_unixts`), EMAs which have no response will not have a timestamp for Begin Time. Even so, we may still be able to chronologically order EMAs if we consider timestamps for Delivered Time (`delivered_unixts`) as well. More specifically, to chronologically order EMAs within `merged.csv`, we may utilize the time variable Aligned Time (`time_unixts`) which is equal to Begin Time (`begin_unixts`) if the participant responded to the EMA (`with_any_response=1`) and equal to Delivered Time (`delivered_unixts`) if the participant did not respond to the EMA (`with_any_response=0`).

Let us place markers chronologically on a timeline. These markers represent Aligned Time for Random EMA (triangles) and Urge EMA (squares).

Draw Figure Here.

Next, let us draw an identical timeline directly on top to visualize how many among the EMAs successfully launched had any response to the EMA questionnaire (`with_any_response=1`).

Draw Figure Here.

2 What information are *not* represented in the rows of `merged.csv`?

Recall that out of the 200 participants enrolled in the PNS study, 37 participants will be excluded from all data analyses. Any responses provided by these 37 participants during the conduct of the study have been excluded from `merged.csv`. Hence, none of the rows `merged.csv` will contain any information from these 37 participants. The IDs of these participants can be referenced in the file `quit_dates_final.csv`. In the `quit_dates_final.csv` file, these participants can be identified by the indicator variable `exclude`, which is equal to 1 if the ID belongs to one of these 37 participants, and equal to 0, if the ID does not belong to one of these 37 participants.

3 What do the columns represent?

The columns suffixed by `_item_XX` refer to items within EMA questionnaires...

Draw Figure Here.

. . . and we can tell

- which EMA questionnaire the column is applicable to by the prefix of the column name (e.g., `postquit_alreadyslipped_item_7` represents an item in the Post-Quit Already Slipped Questionnaire; `postquit_random_item_7` represents an item in the Post-Quit Random EMA Questionnaire)
- what were the original responses provided by participants to the questionnaires; values under the columns suffixed by `_item_XX` were left unchanged from the raw datasets

The curated smoking variables . . .

- `smoking_indicator` - an indicator for whether *any* (1 = at least one; 0 = none) cigarette smoking occurred between the *last assessment* and the *current assessment* (i.e., the EMA Questionnaire from which the value of this variable was derived)
- `smoking_qty` - a count of the number of cigarettes smoked (0, 0.5, 1.5, 3.5, 5.5, 7.5, 9.5) between the *last assessment* and the *current assessment* (i.e., the EMA Questionnaire from which the value of this variable was derived)
- `smoking_delta_minutes` - the number of minutes prior to the *current assessment* (i.e., the EMA Questionnaire from which the value of this variable was derived) when the reported cigarettes smoked in `smoking_qty` occurred; more specifically, if more than one cigarette was smoked, this variable captures the time when that final cigarette stick just prior to the *current assessment* was smoked

. . . are provided side-by-side the original responses to EMA questionnaires.

Draw Figure Here.

Draw Figure Here.

BREAK: Any questions?

Note that not all rows in the `merged.csv` file may be used to construct a dependent variable on smoking behavior. The variable `ema_order` allows us to quickly identify rows which may be used: a missing value represents the fact that the row may not be used but a non-missing value represents the fact that the row may be used. More specifically we have that,

Case #	Type of EMA Questionnaire (`assessment_type`)	Did participant provide any response? (`with_any_response`)	May this row be used?	Value of `ema_order`
1	Pre-/Post- Quit Random EMA	They provided a response (1)	Yes	1, 2, 3, ...
2	Pre-/Post- Quit Random EMA	They did not provide a response (0)	No	missing
3	Pre-/Post- Quit Self-Initiated EMA	They provided a response (1)	Yes	1, 2, 3, ...
4	Pre-/Post- Quit Self-Initiated EMA	They did not provide a response (0)	Yes	1, 2, 3, ...

Recall that participants are asked to report, 'Since *the last assessment*, have you smoked any cigarettes that you did not record?' Hence, the effect of excluding Random EMAs for which the participant did not provide any response is to assume that when the participant recalls smoking information relative to the 'last assessment', that they are only recalling EMAs in Case # 1, 3, 4 above.

Let us reinforce the concept of what it means for a participant to have provided information in a particular EMA Questionnaire *relative to the last assessment* through visual diagrams.

Draw Figure Here.

Draw Figure Here.

BREAK: Any questions?

4 Breaking apart `merged.csv`

Reference file with R code: `module-01.R`

Let's see how these concepts are used as we break apart `merged.csv` into bite-sized chunks of data files which may be used as a starting point for data analysis. The emphasis of this section (and others like it) will be on the workflow and logic of how one would work through grabbing rows/files, rather than `R` syntax.

4.1 Step 1

```
# Read in data file
dat_big_merged <- read.csv(file.path(path_pns_input_data, "merged.csv"),
                           header = TRUE,
                           # required: let R know that missing values
                           # are represented as blanks
                           na.strings = "")
```

```r
# View the first few rows of the data file
head(dat_big_merged)
```

4.2 Step 2

```r
# How do we get from merged.csv to post_quit_random_ema.csv?
dat_postquit_random_ema <- dat_big_merged %>%
  # Select columns corresponding to participant ID and time variables;
  # select columns corresponding to items in Post-Quit Random EMA questionnaire
  select(id:time_unixts, postquit_random_item_1:postquit_random_item_67) %>%
  # Select those rows corresponding to when Post-Quit Random EMA was launched
  filter(assessment_type == "Post-Quit Random") %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)

# How do we get from merged.csv to pre_quit_random_ema.csv?
dat_prequit_random_ema <- dat_big_merged %>%
  # Select columns corresponding to participant ID and time variables;
  # select columns corresponding to items in Pre-Quit Random EMA questionnaire
  select(id:time_unixts, prequit_random_item_1:prequit_random_item_67) %>%
  # Select those rows corresponding to when Pre-Quit Random EMA was launched
  filter(assessment_type == "Pre-Quit Random") %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)
```

Draw Figure Here.

The file `ema_item_names.csv` can be inspected to determine the total number of items that exist in each type of EMA Questionnaire. More specifically, one may inspect the column `name_new`: the largest number `XX` in the suffix `item_XX` for a particular `assessment_type` will be the total number of items that exist in the EMA Questionnaire. In the above code snippet, `postquit_random_item_1:postquit_random_item_67` tells `R` to select those columns in `dat_big_merged` corresponding to items 1, 2, 3, ..., 67 in the Post-Quit Random EMA Questionnaire (i.e., from item 1 all the way through item 67, the last item in this type of EMA Questionnaire). On the other hand, the code snippet `prequit_random_item_1:prequit_random_item_67` tells `R` to select those columns in `dat_big_merged` corresponding to items 1, 2, 3, ..., 67 in the Pre-Quit

Random EMA Questionnaire (i.e., from item 1 all the way through item 67, the last item in this type of EMA Questionnaire).

Although we anticipate that the vast majority of papers will exclusively use responses from Pre-/Post- Quit Random EMA Questionnaires as Independent Variables (IV's), there may be situations where one may want to consider using responses from other kinds of EMA questionnaires. For example, responses from Pre-/Post- Quit Urge EMA Questionnaires when craving (rather than smoking) is the primary Dependent Variable (DV) of interest. In such situations, we may apply a similar logic as the code snippet above to obtain those rows and columns corresponding to Pre-/Post- Quit Urge EMA Questionnaires. An example code snippet applying the logic for Pre-/Post- Quit Urge EMA Questionnaires is displayed below.

```
# How do we get from merged.csv to post_quit_urge_ema.csv?
dat_postquit_urge_ema <- dat_big_merged %>%
  # Select columns corresponding to participant ID and time variables;
  # select columns corresponding to items in Post-Quit Urge EMA questionnaire
  select(id:time_unixts, postquit_urge_item_1:postquit_urge_item_67) %>%
  # Select those rows corresponding to when Post-Quit Urge EMA was launched
  filter(assessment_type == "Post-Quit Urge") %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)

# How do we get from merged.csv to pre_quit_urge_ema.csv?
dat_prequit_urge_ema <- dat_big_merged %>%
  # Select columns corresponding to participant ID and time variables;
  # select columns corresponding to items in Pre-Quit Urge EMA questionnaire
  select(id:time_unixts, prequit_urge_item_1:prequit_urge_item_67) %>%
  # Select those rows corresponding to when Pre-Quit Urge EMA was launched
  filter(assessment_type == "Pre-Quit Urge") %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)
```

4.3 Step 3

```
# How to we get from merged.csv to smoking.csv?
dat_smoking <- dat_big_merged %>%
  select(id:smoking_delta_minutes) %>%
  # We consider the "last assessment" to refer to either of the two situations below:
  # 1. participant-initiated EMAs (any type) having
  # with_any_response=0 or with_any_response=1
  # 2. Random EMA having with_any_response=1
  # In other words, the only situation not included in the "last assessment"
  # are those Random EMAs which the participant did not provide any response
  # All rows in merged.csv having
  filter(!is.na(ema_order)) %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)
```

Draw Figure Here.

BREAK: Any questions?

Module 2: Set-Up of Independent Variables

April 26, 2021

Contents

MODULE 2 GOAL: By the end of this module, you will be able to:

- Identify which rows and columns in the `merged.csv` file to select when one wishes to use responses from Pre-/Post- Quit Random EMA Questionnaires as Independent Variables (IV's) *exclusively from the post-quit period*
- Learn the logic of creating new time variables from existing time variables in the curated datasets

1 Scientific Question

We will begin Module 2 by introducing a scientific question which we will use to motivate our running illustrative example in Modules 2-5.

SCIENTIFIC QUESTION: On average, is self-efficacy *at the current time point* associated with *the proximal occurrence of cigarette smoking* during the post-quit period?

We note that what we will discuss today is a simplified illustrative example of an observational Intensive Longitudinal Data (ILD) analysis.

Draw Figure Here.

2 Potential Issues

What are potential issues we need to consider when investigating our scientific question? In this module, let's consider potential issues concerning our Independent Variable of interest.

Ideally, we would like to infer the associative relationship described in our scientific question of interest at all moments of time during the post-quit period. However,

- we do not have a numerical rating of self-efficacy at *all* time points during the post-quit period
- even if participants provide a rating of self-efficacy through the various different types of EMA Questionnaires, *ratings of self-efficacy provided in Self-Initiated EMA Questionnaires may potentially be influenced by a confounding variable* – namely, a variable simultaneously influencing both the Independent Variable and Dependent Variable in our scientific question

We then ask,

- How is information on self-efficacy collected during the post-quit period?

The question 'I *am* confident in my ability NOT TO SMOKE' is present in 8 out of the 9 types of EMA Questionnaires (i.e., all types of Questionnaires except for the Post-Quit Already Slipped EMA Questionnaire).

Draw Figure Here.

- Among the various types of EMA Questionnaires, which EMA Questionnaires may be utilized for our Independent Variable?

In most cases, we will utilize responses in Post-/Pre- Quit Random EMA Questionnaires for our Independent Variables.

Draw Figure Here.

BREAK: Any questions?

3 Trimming Down the Data Files

Reference file with R code: `module-02.R`

Let's see how these concepts are used as we trim down the data files `dat_postquit_random_ema` and `dat_prequit_random_ema` which we created back in Module 1.

3.1 Step 1

```
subset_dat_postquit_random_ema <- dat_postquit_random_ema %>%
  # Exclude rows corresponding to EMAs having no response to any item
  filter(with_any_response == 1) %>%
  # Exclude rows corresponding to EMAs launched before Quit Date
  filter(use_as_postquit == 1) %>%
  # Select only the columns you will need;
  # this process will result in a smaller data file
  select(id:time_unixts, postquit_random_item_8) %>%
  rename(selfeff = postquit_random_item_8)

subset_dat_prequit_random_ema <- dat_prequit_random_ema %>%
  # Exclude rows corresponding to EMAs having no response to any item
  filter(with_any_response == 1) %>%
  # Exclude rows corresponding to EMAs launched before Quit Date
  filter(use_as_postquit == 1) %>%
  # Select only the columns you will need;
  # this process will result in a smaller data file
  select(id:time_unixts, prequit_random_item_8) %>%
  rename(selfeff = prequit_random_item_8)
```

Draw Figure Here.

The correspondence between the column names in `merged.csv` and the original column names in the codebook `PNS EMA Codebook 07202010.docx` can also be seen in `ema_item_names.csv` by comparing the `name_new` column against the `name_codebook` column.

Let's say you're interested in self-efficacy assessed via Post-Quit Random EMAs as your Independent Variable. The question 'I *am* confident in my ability NOT TO SMOKE' is present in 8 out of the 9 types of EMA Questionnaires (i.e., all types of Questionnaires except for the Post-Quit Already Slipped EMA Questionnaire). Inspection of the codebook `PNS EMA Codebook 07202010.docx` (i.e., you do a control F (find) for the variable `AbsSelfEff`) will show that the variable name `AbsSelfEff` is identical across all types of EMA Questionnaires. Therefore, it is critical to use the file `ema_item_names.csv` to find the variable names with the *prefix* that is appropriate for your analyses. For example, if you are using self-efficacy from Post-Quit Random EMA Questionnaires for your Independent Variable, the appropriate item name would be `postquit_random_item_8`. In other words, you will need to examine the codebook `PNS EMA Codebook 07202010.docx` in parallel with the `ema_item_names.csv` to ensure that you are using the correct variable names.

3.2 Step 2

```
# rbind simply stacks the two data files on top of each other
# rbind will work only if both data files have identical column names
# Hence, there was a need to call the rename() function above prior
# to calling rbind()
dat_analysis <- rbind(subset_dat_postquit_random_ema, subset_dat_prequit_random_ema)
```

Draw Figure Here.

BREAK: Any questions?

4 Creating New Time Variables using Existing Time Variables in the Data Files

Reference file with R code: `module-02.R`

4.1 Step 1

```
# Remember: order according to increasing participant ID
# and within each participant ID, according to increasing time
dat_analysis <- dat_analysis %>% arrange(id, time_unixts)
```

IMPORTANT: The calculations of the time variables in the following step will be incorrect if the data file in the current step has not yet been ordered according to increased participant ID and within each participant ID, according to increased time

4.2 Step 2

```
# Using dat_analysis:
# - Calculate number of hours elapsed between the current EMA and the next EMA

dat_analysis <- dat_analysis %>%
  # The group_by() function is needed to ensure that we do not accidentally
  # use data from another participant to calculate lagged time variables
  # for a particular participant
  group_by(id) %>%
  # When did the participant begin responding to the next EMA?
  # If there is no EMA that follows the current EMA,
  # we will initially set time_unixts_plusone to a missing value
  mutate(time_unixts_plusone = c(tail(time_unixts, n=-1), NA)) %>%
  # If there is no EMA that follows the current EMA, we will set time_unixts_plusone
  # to be equal to end_study_unixts
  mutate(time_unixts_plusone = if_else(is.na(time_unixts_plusone),
                                       end_study_unixts,
                                       time_unixts_plusone)) %>%
  # How many seconds elapsed between the current and next EMA?
  mutate(num_secs_elapsed_since_previous_ema = time_unixts_plusone - time_unixts) %>%
  # Now, make a conversion from seconds to hours
  mutate(num_hrs_elapsed_since_previous_ema = num_secs_elapsed_since_previous_ema/(60*60))
```

Draw Figure Here.

4.3 Step 3

```r
# Using dat_analysis:
# - Calculate number of days elapsed since the beginning of post-quit period
# - Calculate number of days elapsed since the start of the study

dat_analysis <- dat_analysis %>%
  # How many seconds elapsed between the current EMA and Quit Date?
  # Note that we consider 4AM on Quit Date to be the time when participants
  # quit smoking in the PNS study
  mutate(num_secs_elapsed_since_quit = time_unixts - quit_unixts) %>%
  # Now, make a conversion from seconds to hours
  mutate(num_hrs_elapsed_since_quit = num_secs_elapsed_since_quit/(60*60)) %>%
  # Now, make a conversion from hours to days
  mutate(num_days_elapsed_since_quit = num_hrs_elapsed_since_quit/24)

# Create a new time variable that captures the number of days elapsed
# since 12AM of start of study
dat_analysis <- dat_analysis %>%
  mutate(num_secs_elapsed_since_start_study = time_unixts - start_study_unixts) %>%
  mutate(num_hrs_elapsed_since_start_study = num_secs_elapsed_since_start_study/(60*60)) %>%
  mutate(num_days_elapsed_since_start_study = num_hrs_elapsed_since_start_study/24)
```

Draw Figure Here.

BREAK: Any questions?

Module 3: Laying the Groundwork for Set-Up of Dependent Variables through Data Visualization

April 26, 2021

Contents

MODULE 3 GOAL: By the end of this module, you will be able to:

- Use the code in `module-03.R` to create a visual snapshot of the number of EMAs and time between EMAs within the data file `dat_smoking` – the data file which we will be using to construct our Dependent Variable.
- Through a live demo, know what to expect when using `module-03.R` independently.

1 Working with `module-03.R`

1.1 Step 1

We begin with the `dat_smoking` data file we created in earlier modules.

```
# How to we get from merged.csv to smoking.csv?
dat_smoking <- dat_big_merged %>%
  select(id:smoking_delta_minutes) %>%
  # We consider the "last assessment" to refer to either of the two situations below:
  # 1. participant-initiated EMAs (any type) having with_any_response=0
  # or with_any_response=1
  # 2. Random EMA having with_any_response=1
  # In other words, the only situation not included in the "last assessment"
  # are those Random EMAs which the participant did not provide any response
  # All rows in merged.csv having
  filter(!is.na(ema_order)) %>%
  # Remember: order according to increasing participant ID
  # and within each participant ID, according to increasing time
  arrange(id, time_unixts)
```

1.2 Step 2

```r
# Parameters that may be adjusted

# e.g., if this is 2, then we are selecting the 2nd participant ID in the list
choose_idx <- 38
# Minimum number of days since 12AM of start of study date
xlim_min <- 0
# Maximum number of days since 12AM of start of study date
xlim_max <- 28
```

1.3 Step 3

```r
# Do not adjust plotting parameters below this line

if(xlim_min > xlim_max){
  print("Error: xlim_min must be less than or equal to xlim_max")
}

# What are the unique participant IDs which are present in dat_smoking?
participant_ids <- unique(dat_smoking$id)

# Let's visualize the data for one particular participant
current_participant <- participant_ids[choose_idx]

# Take rows corresponding to this particular participant
plotdat_participant <- dat_smoking %>% filter(id == current_participant)

# Create a new time variable (just like we did before)
# that captures the number of days elapsed since 12AM of start of study
plotdat_participant <- plotdat_participant %>%
  mutate(num_secs_elapsed_since_start_study = time_unixts - start_study_unixts) %>%
  mutate(num_hrs_elapsed_since_start_study = num_secs_elapsed_since_start_study/(60*60)) %>%
  mutate(num_days_elapsed_since_start_study = num_hrs_elapsed_since_start_study/24)  %>%
  mutate(roundeddown_num_days_elapsed_since_start_study = floor(num_days_elapsed_since_start_study))

# Layer on each component of the plot
plot(-1, xaxt = "n", yaxt = "n",
     xlab = "Day Since Start of Study ('0' represents midnight on the date when study began)",
     ylab = "",
     xlim = c(xlim_min, xlim_max), ylim = c(0,0.3),
     cex.lab = 2,
     frame.plot = FALSE)

if(xlim_max - xlim_min <=7){
  # Use half-day increments
  axis(1, at = seq(xlim_min, xlim_max + 1 , 0.5), cex.axis = 2, lwd.ticks = 2, gap.axis = 1.2)
}else{
  # Use increments of 7 days
  axis(1, at = seq(xlim_min, xlim_max + 1 , 7), cex.axis = 2, lwd.ticks = 2, gap.axis = 1.2)
}

# Identify which rows correspond to each kind of EMA
```

```r
plotdat_random <- plotdat_participant %>%
  filter(assessment_type == "Post-Quit Random")

plotdat_urge <- plotdat_participant %>%
  filter(assessment_type == "Post-Quit Urge")

plotdat_already_slipped <- plotdat_participant  %>%
  filter(assessment_type == "Post-Quit Already Slipped")

plotdat_part_one <- plotdat_participant %>%
  filter(assessment_type == "Post-Quit About to Slip Part One")

plotdat_part_two <- plotdat_participant %>%
  filter(assessment_type == "Post-Quit About to Slip Part Two")

abline(v = plotdat_random$num_days_elapsed_since_start_study, lty = 2, lwd = 2, col = "red")

# Note that if the number of rows in the plot data is equal to zero,
# then no new points will be added to the existing plot; no error message will be displayed
points(plotdat_random$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_random)),
       pch = 17, cex = 2, col = "black")

points(plotdat_urge$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_urge)),
       pch = 19, cex = 2, col = "orange")

points(plotdat_already_slipped$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_already_slipped)),
       pch = 19, cex = 2, col = "seagreen")

points(plotdat_part_one$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_part_one)),
       pch = 19, cex = 2, col = "lightblue")

points(plotdat_part_two$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_part_two)),
       pch = 19, cex = 2, col = "blue")

# Identify which rows correspond to each kind of EMA
plotdat_random <- plotdat_participant %>%
  filter(assessment_type == "Pre-Quit Random")

plotdat_urge <- plotdat_participant %>%
  filter(assessment_type == "Pre-Quit Urge")

plotdat_part_one <- plotdat_participant %>%
  filter(assessment_type == "Pre-Quit About to Slip Part One")

plotdat_part_two <- plotdat_participant %>%
  filter(assessment_type == "Pre-Quit About to Slip Part Two")

abline(v = plotdat_random$num_days_elapsed_since_start_study, lty = 2, lwd = 2, col = "red")
```

```r
# Note that if the number of rows in the plot data is equal to zero,
# then no new points will be added to the existing plot; no error message will be displayed
points(plotdat_random$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_random)),
       pch = 17, cex = 2, col = "black")

points(plotdat_urge$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_urge)),
       pch = 19, cex = 2, col = "orange")

points(plotdat_already_slipped$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_already_slipped)),
       pch = 19, cex = 2, col = "seagreen")

points(plotdat_part_one$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_part_one)),
       pch = 19, cex = 2, col = "lightblue")

points(plotdat_part_two$num_days_elapsed_since_start_study,
       rep(0.1, nrow(plotdat_part_two)),
       pch = 19, cex = 2, col = "blue")

legend("topright", c("Random", "Urge", "Already Slipped", "Part One", "Part Two"),
       col = c("black", "orange", "seagreen", "lightblue", "blue"),
       pch = c(17, 19, 19, 19, 19), pt.cex = rep(2,5), cex = 1.2)
```

2 Live Demo of `module-03.R`

BREAK: Any questions?

Module 4: Set-Up of Dependent Variable

April 26, 2021

Contents

MODULE 4 GOAL: By the end of this module, you will be able to:

- Learn the logic of constructing the Dependent Variable using scientific considerations (i.e., our motivating scientific question) and practical considerations (i.e., the data collection design).

1 Scientific Question

For convenience, let's display the scientific question we introduced in an earlier module.

SCIENTIFIC QUESTION: On average, is self-efficacy *at the current time point* associated with *the proximal occurrence of cigarette smoking* during the post-quit period?

Draw Figure Here.

2 Implementation

Reference file with R code: `module-04.R`

We will zoom in on specific sections within the implementation in `module-04.R`.

An outer loop goes through each participant. For each participant, the inner loop goes through each of their rows in `dat_analysis`. Recall that, at this point, only Random EMAs having *any* response

(i.e., `with_any_response=1`) are included in `dat_analysis`. Hence, for a particular participant, `total_random_ema` will be the total number of Random EMAs for which the participant provided *any* response.

```
for(i in 1:total_participant_ids){

  # More code here

  for(j in 1:total_random_ema){

    # More code here

  }
}
```

Draw Figure Here.

The inner loop calculates `count_within_bounds`, our Dependent Variable.

```
  for(j in 1:total_random_ema){
    current_lower_bound <- all_lower_bound[j]
    current_upper_bound <- all_upper_bound[j]

    # How many EMAs were launched between the two Random EMAs we
    # are looking at now?
    dat_within_bounds <- current_dat_smoking %>%
      # Note the use of '>' instead of '>=' when checking against left end point
      # We do not include the number of reported cigarettes smoked in the left end point
      # However, we will include the number of reported cigarettes smoked in the right end point
      filter((time_unixts > current_lower_bound) & (time_unixts <= current_upper_bound))

    number_within_bounds <- nrow(dat_within_bounds)
    # Only proceed with further calculations if we have at least one EMA
    if(number_within_bounds > 0){
      number_missing <- sum(is.na(dat_within_bounds$smoking_qty))
      # Only proceed with further calculations if there is no missing value in smoking_qty
      if(number_missing == 0){
        current_count_within_bounds <- sum(dat_within_bounds$smoking_qty)
      } # Mark end of IF STATEMENT
```

```
  } # Mark end of IF STATEMENT

  current_dat_analysis$count_within_bounds[j] <- current_count_within_bounds
} # Mark end of FOR LOOP over Random EMAs having with_any_response=1
```

What are the bounds?

Draw Figure Here.

Which EMAs within these bounds should we *not* use?

Draw Figure Here.

When may missing values come about in our Dependent Variable `count_within_bounds`?

Draw Figure Here.

BREAK: Any questions?

Module 5: Modeling and Hypothesis Testing

April 26, 2021

Contents

MODULE 5 GOAL: By the end of this module, you will be able to:

- Identify which rows to select in a 'Main Analysis' and a 'Sensitivity Analysis'.
- See an example of how the dataset created using the process discussed in Modules 1-4 can be used to test hypothesis.
- Through a live demo, know what to expect when using `module-05.R` independently.

1 Scientific Question

For convenience, let's display the scientific question we introduced in an earlier module.

SCIENTIFIC QUESTION 1: On average, is self-efficacy *at the current time point* associated with *the proximal occurrence of cigarette smoking* during the post-quit period?

We note that the time variables we constructed in earlier modules also allow us to test whether the above associative effect varies with time, i.e.,

SCIENTIFIC QUESTION 2: Does association of self-efficacy *at the current time point* with *the proximal occurrence of cigarette smoking* vary across time during the post-quit period?

2 Dataset for Analysis

Let's now see how the Independent Variables, Dependent Variables, and Time Variables we have grown acquainted with in Modules 1-4 now come together into a dataset we may utilize to investigate the scientific questions above.

The dataset we now have (i.e., `dat_analysis` from Module 4) contains the following columns...

- `id`
- `sensitivity`
- `selfeff`
- `num_days_elapsed_since_quit`
- `num_hrs_elapsed_since_previous_ema`
- `count_within_bounds`

. . . and the information in these columns may be visualized in relation to each other.

Draw Figure Here.

BREAK: Any questions?

3 Models for the Dependent Variable

In terms of variables in `dat_analysis` . . .

MODEL FOR SCIENTIFIC QUESTION 1:

$$\log\left(E\left\{\frac{\text{count\_within\_bounds}}{\text{num\_hrs\_elapsed\_since\_previous\_ema}}\right\}\right) = \beta_0 + \beta_1 \text{selfeff} + \nu_i$$

MODEL FOR SCIENTIFIC QUESTION 2:

$$\begin{aligned}\log\left(E\left\{\frac{\text{count\_within\_bounds}}{\text{num\_hrs\_elapsed\_since\_previous\_ema}}\right\}\right) = \beta_0 &+ \beta_1 \text{selfeff} \\ &+ \beta_2 \text{num\_days\_elapsed\_since\_quit} \\ &+ \beta_3(\text{num\_days\_elapsed\_since\_quit} \times \text{selfeff}) \\ &+ \nu_i\end{aligned}$$

In terms of math . . .

MODEL FOR SCIENTIFIC QUESTION 1:

$$\log\left(E\left\{\frac{Y_{i,t_j}}{L_{i,t_j}}\right\}\right) = \beta_0 + \beta_1 X_{i,t_j} + \nu_i$$

MODEL FOR SCIENTIFIC QUESTION 2:

$$\log\left(E\left\{\frac{Y_{i,t_j}}{L_{i,t_j}}\right\}\right) = \beta_0 + \beta_1 X_{i,t_j} + \beta_2 D_{i,t_j} + \beta_3 (D_{i,t_j} \times X_{i,t_j}) + \nu_i$$

4 Hypothesis Testing

Reference file with R code: `module-05.R`

4.1 Step 1

```
dat_analysis <- dat_analysis %>%
  select(id,
         sensitivity,
         selfeff,
         num_days_elapsed_since_quit,
         num_hrs_elapsed_since_previous_ema,
         count_within_bounds)

# Transform hours elapsed into log-scale
dat_analysis$logged_hrs_elapsed <- log(dat_analysis$num_hrs_elapsed_since_previous_ema)

# Round up totals;
# For example, 0.5 will be rounded up to 1; 1.5 will be rounded up to 2, etc.
dat_analysis$roundedup_count_within_bounds <- as.integer(ceiling(dat_analysis$count_within_bounds))
```

4.2 Step 2

In this step, we will create two data files:

- A data file comprising of those participants who have either *low* or *high* ambiguity in their Quit Date (`dat_main_analysis`); analyses utilizing these participants will be referred to as *'Main Analysis'*
- A data file comprising solely of those participants who *low* ambiguity in their Quit Date (`dat_sensitivity_analysis`); analyses utilizing these participants will be referred to as *'Sensitivity Analysis'*

```
# Create a new data frame, which is essentially dat_analysis copied
dat_main_analysis <- dat_analysis

# Now, using dat_main_analysis, take those rows which will be included
# in Sensitivity Analysis. In other words, drop all those rows which should
# be excluded from Sensitivity Analysis
dat_sensitivity_analysis <- dat_main_analysis %>% filter(sensitivity == 1)
```

4.3 Step 3

We note that we will be using *identical models* for 'Main Analysis' and 'Sensitivity Analysis'. Both types of analyses only differ with respect to which participants will be used to estimate the two models discussed above.

In the remaining steps, we will do a complete-case analysis. Rows having missing values in any of the dependent variables or independent variables will be omitted. In your analysis, you would have to consider how to address missing data in both of these variables, e.g., via an imputation procedure prior to running `glmer`.

```
# Estimate coefficients of the model using glmer
fit_main_1 <- glmer(roundedup_count_within_bounds ~ offset(logged_hrs_elapsed)
                                                  + 1 + selfeff
                                                  + (1 | id),
                    data = dat_main_analysis,
                    family = poisson(link="log"),
                    na.action = na.omit)
```

4.4 Step 4

```
# Estimate coefficients of the model using glmer
fit_sensitivity_1 <- glmer(roundedup_count_within_bounds ~ offset(logged_hrs_elapsed)
                                                         + 1 + selfeff
                                                         + (1 | id),
                           data = dat_sensitivity_analysis,
                           family = poisson(link="log"),
                           na.action = na.omit)
```

4.5 Step 5

A similar logic to Steps 3 and 4 above may be used to estimate the model for Scientific Question 2. To help with convergence of the estimation process, we will rescale the variable `num_days_elapsed_since_quit` prior to estimation (try removing the division by 100!). This rescaling has the effect of estimating the following model:

$$\begin{aligned}\log\left(E\left\{\frac{\text{count\_within\_bounds}}{\text{num\_hrs\_elapsed\_since\_previous\_ema}}\right\}\right) &= \beta_0 + \beta_1 \text{selfeff} \\ &+ \beta_2 \frac{\text{num\_days\_elapsed\_since\_quit}}{100} \\ &+ \beta_3 \left(\frac{\text{num\_days\_elapsed\_since\_quit}}{100} \times \text{selfeff}\right)\end{aligned}$$

```
# Estimate coefficients of the model using glmer
fit_main_2 <- glmer(roundedup_count_within_bounds ~ offset(logged_hrs_elapsed)
                                                  + 1 + selfeff
```

```
                                                + I(num_days_elapsed_since_quit/100)
                                                + selfeff:I(num_days_elapsed_since_quit/100)
                                                + (1 | id),
                        data = dat_main_analysis,
                        family = poisson(link="log"),
                        na.action = na.omit)
```

4.6 Step 6

```
# Estimate coefficients of the model using glmer
fit_sensitivity_2 <- glmer(roundedup_count_within_bounds ~ offset(logged_hrs_elapsed)
                                                          + 1 + selfeff
                                                          + I(num_days_elapsed_since_quit/100)
                                                          + selfeff:I(num_days_elapsed_since_quit/100)
                                                          + (1 | id),
                           data = dat_sensitivity_analysis,
                           family = poisson(link="log"),
                           na.action = na.omit)
```

5 Live Demo of `module-05.R`

BREAK: Any questions?

Workshop Wrap Up

April 26, 2021

Contents

1 Recap

When we kicked off the workshop, we begun with the overarching goal of taking away the following skills from the workshop:

- The ability to correctly interpret columns in the curated datasets
- The ability to correctly identify rows which need to be included/excluded when working on *formulating* one's scientific question of interest

Modules 1 - 5 showed more concretely how we are able to accomplish these goals. Much more, we have also:

- Discussed how the design of the data collection process will raise nuances one needs to consider when constructing Dependent Variables and Independent Variables.

Indeed one cannot construct Dependent Variables and Independent Variables correctly without simultaneous consideration of the data collection process.

- Illustrated one approach to constructing the Dependent Variable.

The approach uses scientific considerations (i.e., our motivating scientific question) and practical considerations (i.e., the data collection design).

- Provided a tool (i.e., code) one may use to create a visual snapshot of the number of EMAs and time between EMAs within the data file which we will be using to construct our Dependent Variable (`dat_smoking`).

The tool may be used to create visual snapshots that zoom in or zoom out of specific moments of time during the study period.

2 Next Steps

- Workshop participants will be asked to complete a workshop evaluation survey.

3 Parting Points

- Draw figures so that you are grasping the logic and implications of decisions made during the data preparation process.
- There are many different ways to conceptualize Independent Variables and Dependent Variables; the approach we discussed today is one of many possibilities. Hence, the code implementation discussed today is not the only implementation you will ever need or use.

4 Open Discussion Session

- What is a question you have about what we did *not* cover today?

'To sum it all up, I feel we are just as confused as ever in some ways, but I believe we are confused at a *higher level* and about *more important things*.' – David A. Peoples