Por Nuestra Salud Study: Sanity Checks on Curated Data

March 24, 2020

1 About

In this file, we document sanity checks on the Por Nuestra Salud (PNS) study curated data. Curated data are expected to (1) be consistent with the study design, and (2) be internally consistent, i.e., having no conflicting information between rows or columns in the dataset. To this end, we use the `testthat` package to organize *programmatic checks* on the curated data and the `ggplot2` package to create *visual checks* on the curated data.

```r
use.samp.size <- 20
pns.quit.dates <- read.csv(file.path(path.pns.input_data, "pns_quit_dates.csv"), header = TRUE)
use.df.ids <- SampleAndRename(df = pns.quit.dates, use.seed = 754369, samp.size = use.samp.size)
```

All visual checks are displayed for the same sample of 20 randomly chosen PNS study participants: `SampleAndRename()` chooses a sample of participant ID's of size `use.samp.size` and provides new participant ID's beginning and ending at 1 and `use.samp.size`, respectively, for plotting. Functions used to create displays of visual checks, such as `SampleAndRename()`, `PlotSmokingOutcome()`, `PlotPostQuitEMATime()`, and `PlotPostQuitNumericResponses()` are defined in the script `pns-plot-utils.R`.

2 Smoking Outcome Curated Datasets

Let us read in the smoking outcome curated datasets. These curated datasets differ in the type of information from the raw PNS study data that are used to construct the smoking outcome. More details how these curated datasets were constructed can be found in `PNS_documentation.pdf`.

```r
df.smoking.01 <- read.csv(file.path(path.pns.output_data, "pns.smoking.01.csv"), header = TRUE)
df.smoking.02 <- read.csv(file.path(path.pns.output_data, "pns.smoking.02.csv"), header = TRUE)
```

2.1 Programmatic Checks

All smoking outcome curated datasets have the same structure, hence, a common set of checks can be performed. The call `test_file(file.path(path.pns.code, "pns-test-file-01.R"))` triggers the following checks on the internal consistency of `df`, a given smoking outcome curated dataset: checks on whether . . .

1. lower bound of time interval should come before upper bound of time interval
2. lower bound of time intervals are in ascending order
3. upper bound of time intervals are in ascending order
4. lower bound of current interval equals upper bound of previous interval
5. YES- or NO-labelled intervals are at most 24 hours in length

If a given smoking outcome curated dataset passes all checks in `pns-test-file-01.R`, then the number of tests `Failed` will be displayed as `0`.

```
df <- df.smoking.01
test_file(file.path(path.pns.code, "pns-test-file-01.R"))
```

```
## v |  OK F W S | Context
## / |   0       | Construction of smoking intervals: internal consistency of curated data| |   3
##
## == Results =========================================================================================
## Duration: 0.3 s
##
## OK:       5
## Failed:   0
## Warnings: 0
## Skipped:  0
```

```
df <- df.smoking.02
test_file(file.path(path.pns.code, "pns-test-file-01.R"))
```

```
## v |  OK F W S | Context
## / |   0       | Construction of smoking intervals: internal consistency of curated data| |   3
##
## == Results =========================================================================================
## Duration: 0.2 s
##
## OK:       5
## Failed:   0
## Warnings: 0
## Skipped:  0
```

2.2 Visual Checks

As the various smoking outcome curated datasets differ in the type of information from the raw PNS study data that are used to construct the smoking outcome, we expect that (1) the total number of rows will vary across smoking outcome curated datasets (2) in visual checks on the smoking outcome curated datasets, we expect to see a change in length of some YES- and NO-labelled intervals across the smoking outcome curated datasets.

```
nrow(df.smoking.01)
```

```
## [1] 7928
```

```
nrow(df.smoking.02)
```

```
## [1] 8537
```

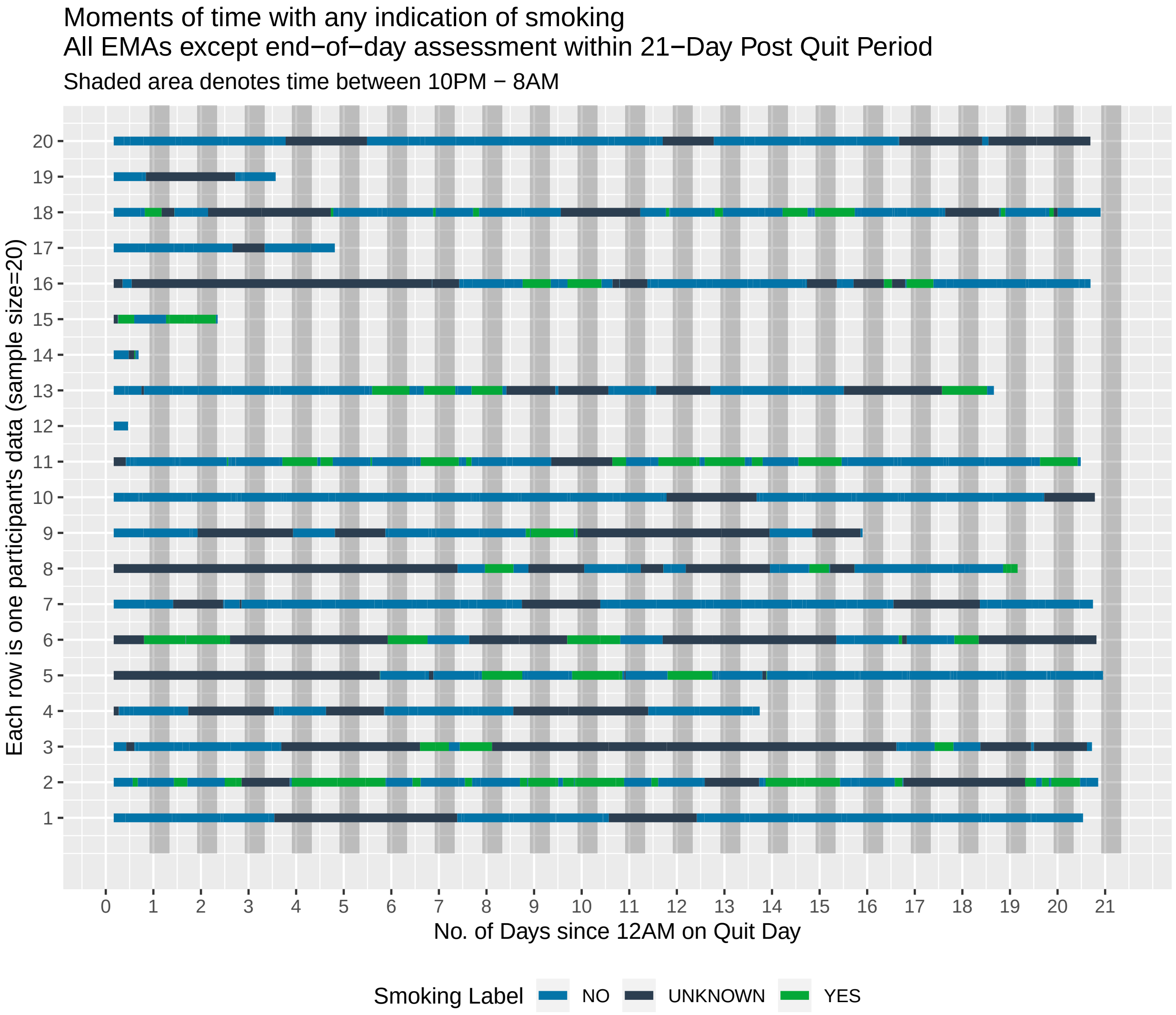
```
df.smoking.01 %>%
  mutate(interval.length = (UB.ts - LB.ts)/(60*60)) %>%
  group_by(smoking.label) %>%
  summarise(count = n(),
            mean.interval.length = mean(interval.length))
```

```
## # A tibble: 3 x 3
##   smoking.label count mean.interval.length
##   <fct>         <int>                <dbl>
## 1 NO             6246                 6.46
## 2 UNKNOWN         856                24.8
## 3 YES             826                 6.04
```

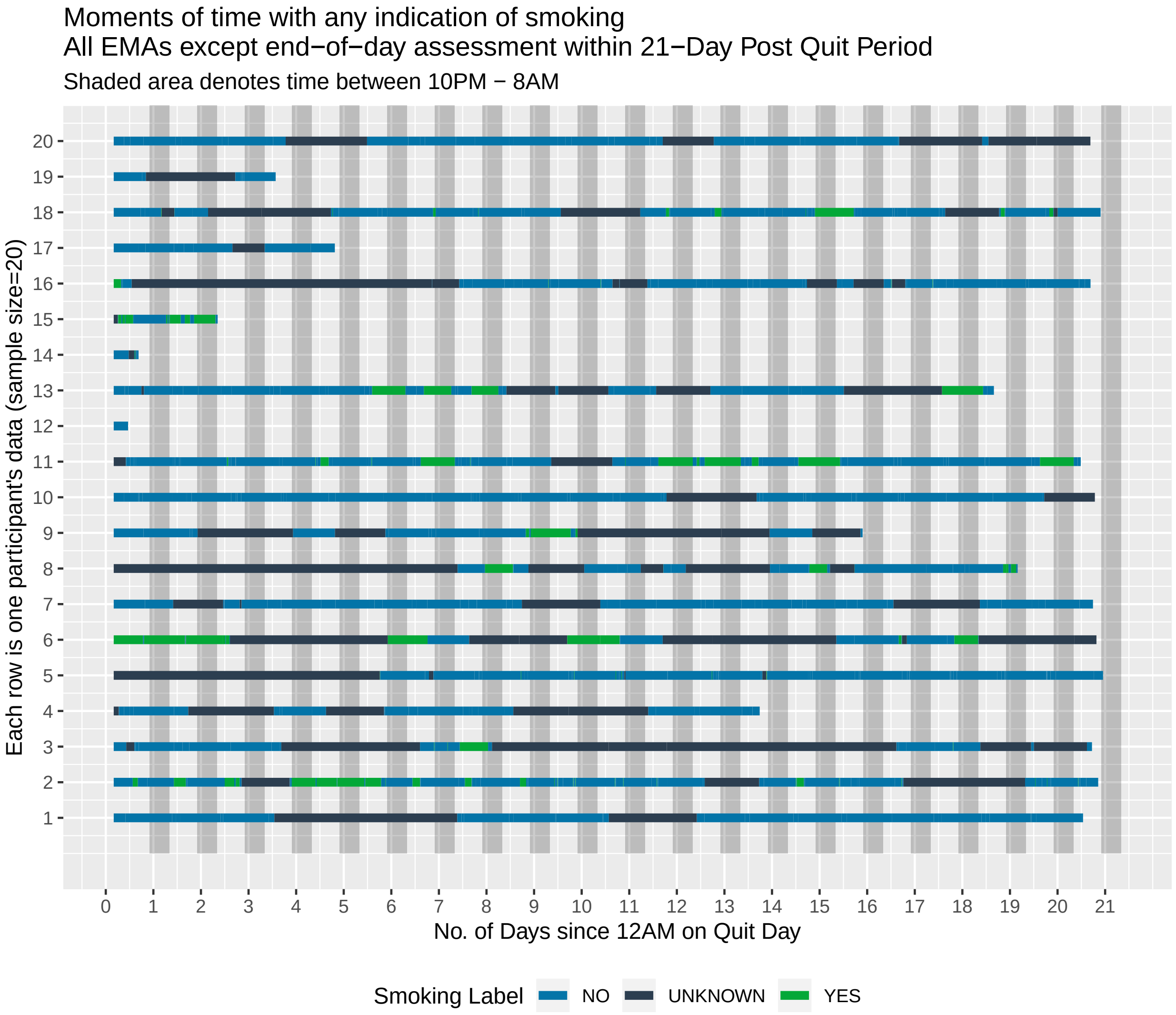
```
df.smoking.02 %>%
  mutate(interval.length = (UB.ts - LB.ts)/(60*60)) %>%
  group_by(smoking.label) %>%
  summarise(count = n(),
            mean.interval.length = mean(interval.length))
```

```
## # A tibble: 3 x 3
##   smoking.label count mean.interval.length
##   <fct>         <int>                <dbl>
## 1 NO             6855                 6.10
## 2 UNKNOWN         837                25.1
## 3 YES             845                 4.38
```

```
gg.smoking.01 <- PlotSmokingOutcome(df.smoking = df.smoking.01, df.ids = use.df.ids)
gg.smoking.01
```



```r
gg.smoking.02 <- PlotSmokingOutcome(df.smoking = df.smoking.02, df.ids = use.df.ids)
gg.smoking.02
```



3 Post-Quit Random Curated Datasets

The post-quit random curated datasets, named as `pns.postquitrandom.XX.csv`, only differ in which subset of post-quit random EMA items are included as columns, while the manner of constructing timestamps corresponding to each row are identical across all post-quit random curated datasets. We choose one post-quit random curated dataset on which to perform checks on consistency with study design and internal consistency.

```
# pns.postquitrandom.01.csv is the minimal post-quit random curated dataset
df.post.quit.random.01 <- read.csv(file.path(path.pns.output_data,
                                             "pns.postquitrandom.01.csv"),
                                   header = TRUE)
```

3.1 Programmatic Checks

The call `test_file(file.path(path.pns.code, "pns-test-file-02.R"))` triggers the following checks on the internal consistency of `df`, a given Post-Quit Random curated dataset: checks on whether …

1. order of total prompts since start should match order of time.unixts timestamps

If a given Post-Quit Random curated dataset passes all checks in `pns-test-file-02.R`, then the number of tests `Failed` will be displayed as `0`.

```
df <- df.post.quit.random.01
test_file(file.path(path.pns.code, "pns-test-file-02.R"))
```

```
## v |  OK F W S | Context
## / |   0       | Construction of Post-Quit Random EMA Datasetsv |   1       | Construction of Post-Qui
##
## == Results =========================================================================================
## OK:       1
## Failed:   0
## Warnings: 0
## Skipped:  0
##
## You are a coding rockstar!
```

To check consistency of the Post-Quit Random curated dataset with the study design, we tabulate the number of Post-Quit Random EMAs in the curated dataset per participant.

```
df <- df.post.quit.random.01
```

Across the 21-day post quit period, let us tabulate the mean, minimum, and maximum number of Post-Quit Random EMAs per participant in the Post-Quit Random curated data. The design of the study was such that participants would be prompted to 3 Random EMAs per day for each of the 21 days during the Post-Quit study period via a mobile device which a participant could choose to switch off during the course of the study, so that we expect at most $21 \times 3 = 63$ per participant.

```
df %>%
  group_by(id) %>%
  summarise(count = n()) %>%
  summarise(mean.count = mean(count),
            min.count = min(count),
            max.count = max(count),
            more.than.expected = sum(1*(count>=64)))
```

```
## # A tibble: 1 x 4
##   mean.count min.count max.count more.than.expected
##        <dbl>     <int>     <int>              <dbl>
## 1       45.5         1        67                  5
```

Now, per paticipant, let us tabulate the number of Post-Quit Random EMAs in the curated dataset with `engaged.yes=0` and `engaged.yes=1` and then calculate the mean, minimum, and maximum across participants.

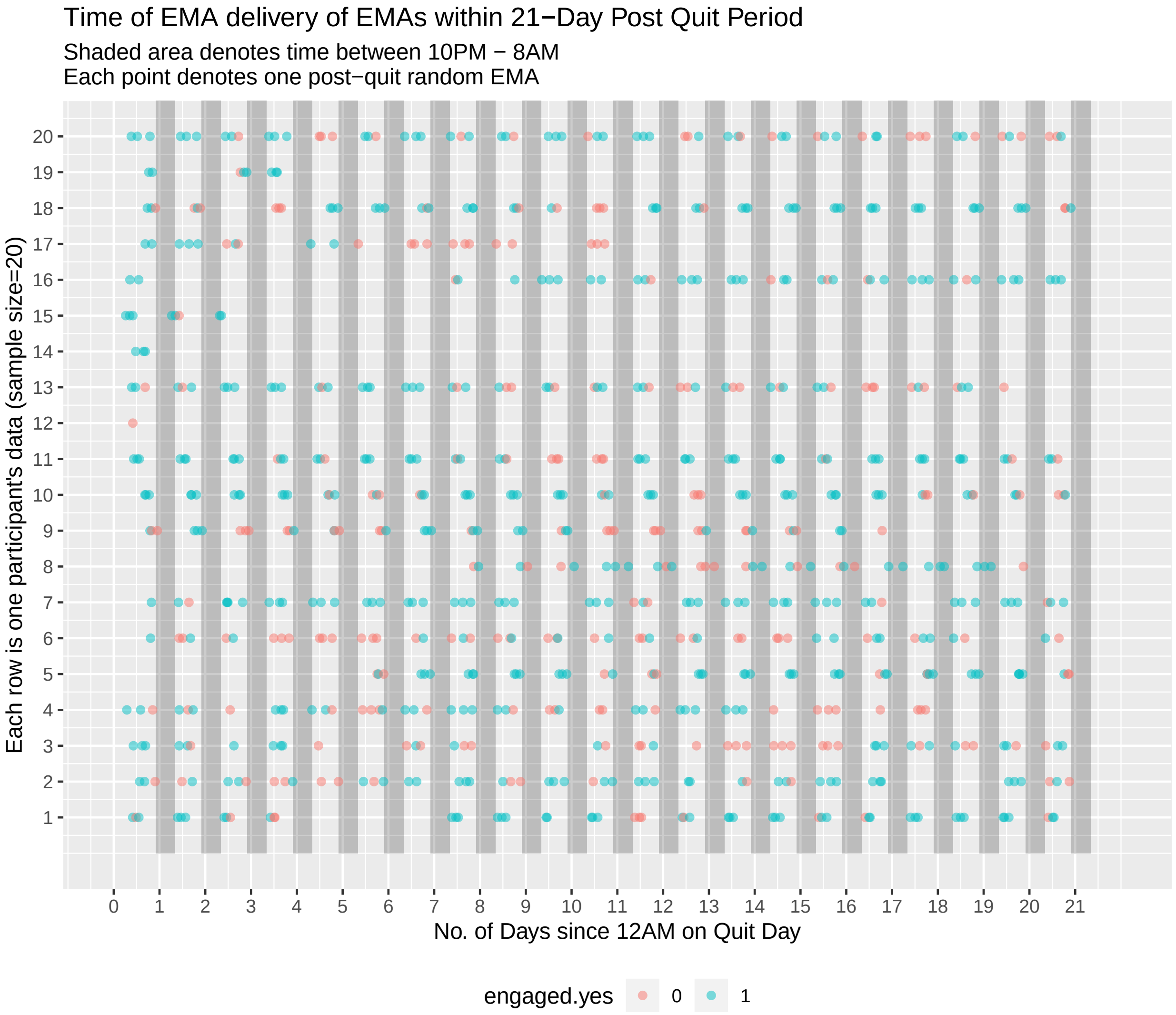
```
df %>%
  group_by(id, engaged.yes) %>%
  summarise(count = n()) %>%
  group_by(engaged.yes) %>%
  summarise(mean.count = mean(count),
            min.count = min(count),
            max.count = max(count),
            more.than.expected = sum(1*(count>=64)))
```

```
## # A tibble: 2 x 5
##   engaged.yes mean.count min.count max.count more.than.expected
##         <int>      <dbl>     <int>     <int>              <dbl>
## 1           0       14.5         1        56                  0
## 2           1       32.7         1        63                  0
```

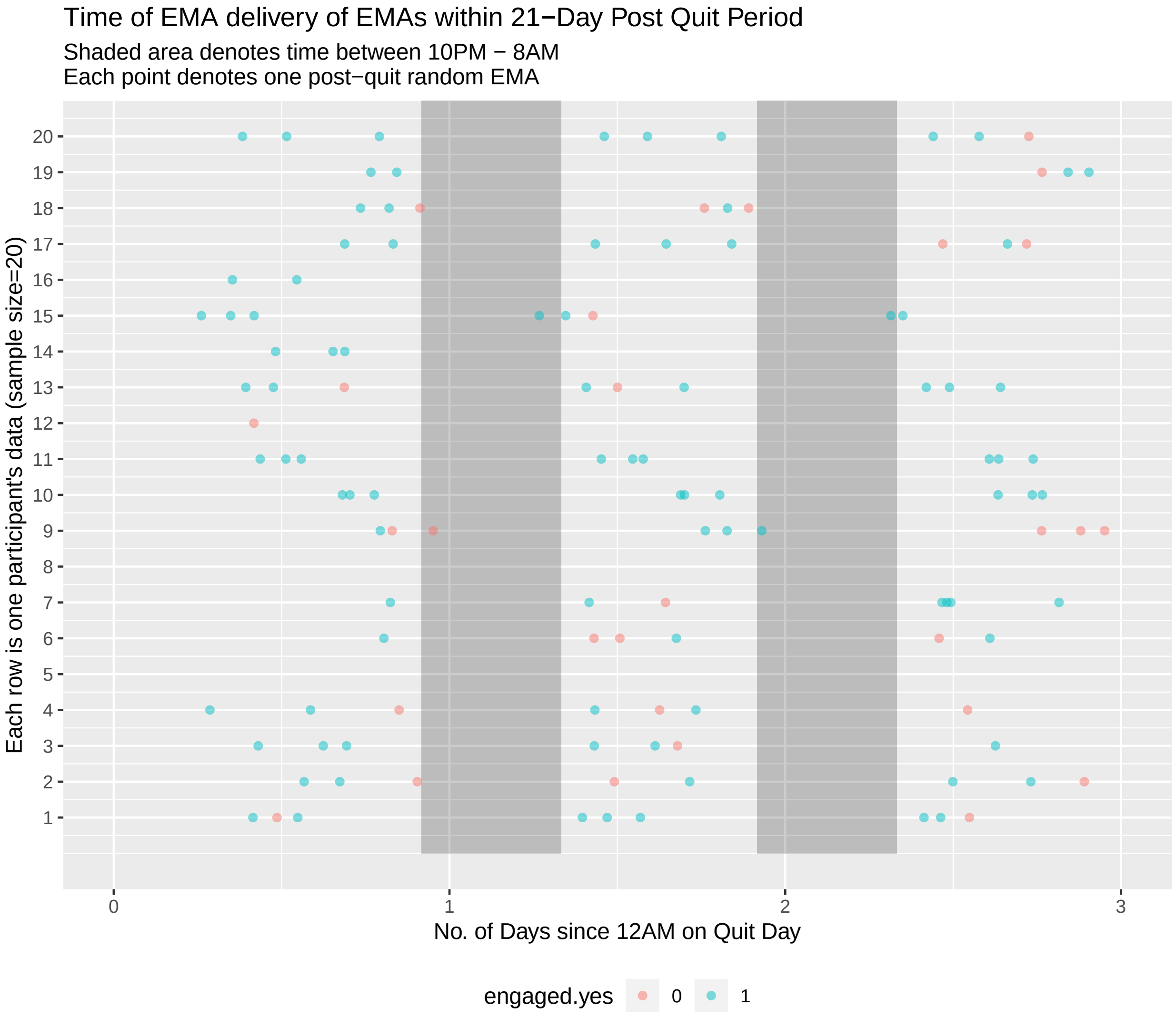
3.2 Visual Checks

The design of the study was such that participants would not be prompted to Random EMAs during the evening. Timing or Random EMA delivery (for Random EMA's with `engaged.yes=0`) or timing of when an individual began completing a Random EMA (for Random EMA's with `engaged.yes=1`) are displayed with time of day marked in the background. The vast majority of Random EMA delivery or completion are expected to occur outside the 10pm-8am time window.

```
gg.pq.random <- PlotPostQuitEMATime(df.post.quit = df.post.quit.random.01,
                                    df.ids = use.df.ids,
                                    plot.days = 22,
                                    ema.type="random")

gg.pq.random
```



```
gg.pq.random.zoom <- PlotPostQuitEMATime(df.post.quit = df.post.quit.random.01,
                                         df.ids = use.df.ids,
                                         plot.days = 3,
                                         ema.type="random")

gg.pq.random.zoom
```



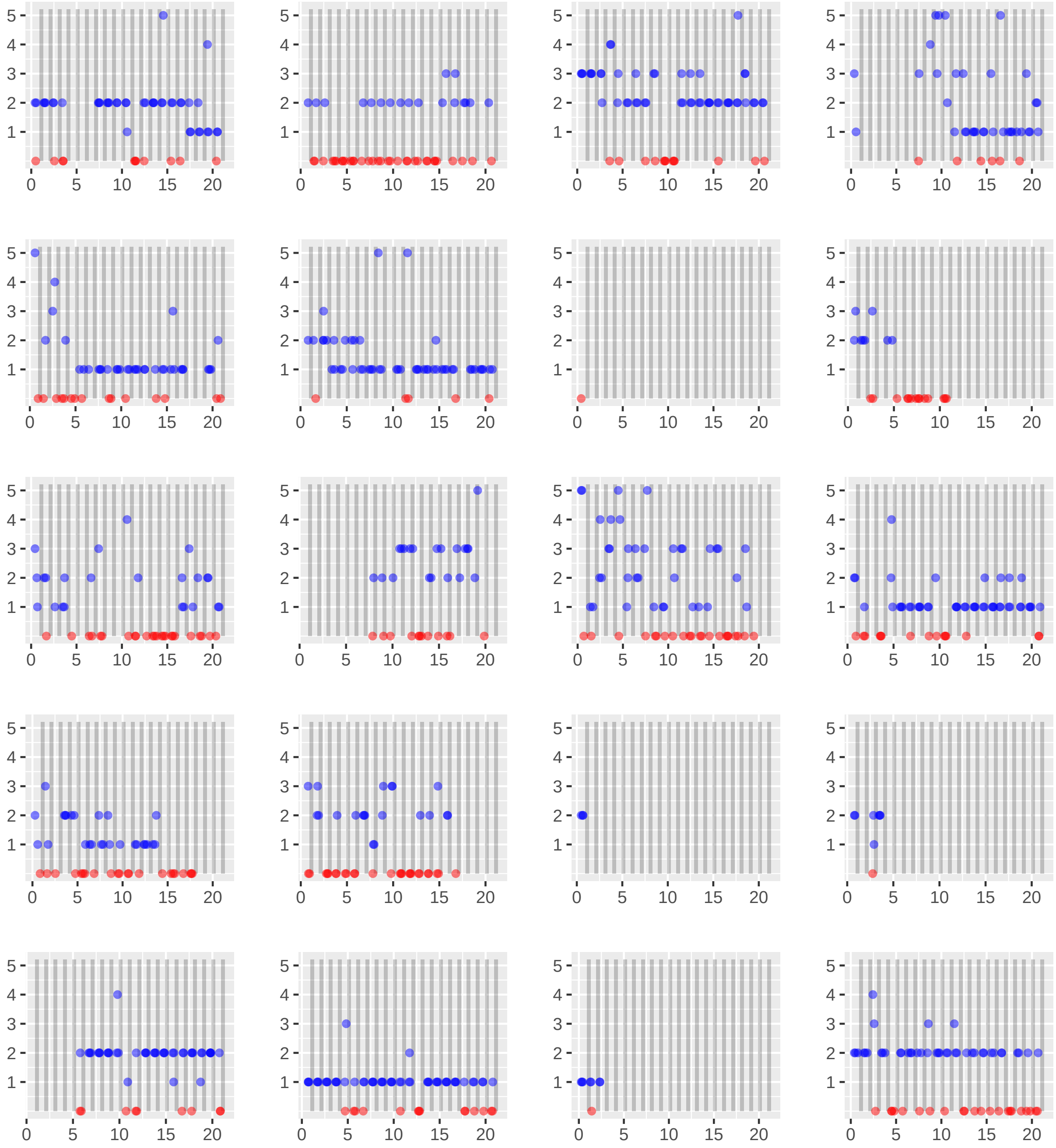
Let us visually inspect the pattern of responses during the 21 day Post-Quit study period for select Post-Quit Random EMA items.

Variable Name	Question
Affect6	I feel angry.
Affect7	I feel anxious.
Affect8	I feel restless.

```r
all.vars <- c(paste("Affect",c(6,7,8),sep=""))
collect.plot.grid <- list()

for(i in 1:length(all.vars)){
  use.var.name <- all.vars[i]

  collect.plots <- PlotPostQuitNumericResponses(df.post.quit = df.post.quit.random.01,
                                                var.name = use.var.name,
                                                df.ids = use.df.ids)

  text.top <- paste(use.var.name,
                    "Time of EMA delivery versus response on a 5-point Likert scale",
                    "All Random EMAs within 21-Day Post Quit Period",
                    sep="\n")
  text.bottom <- paste("Shaded area denotes time between 10PM - 8AM",
                       "Each point denotes one random EMA",
                       "red dots: engaged.yes=0, blue dots: engaged.yes=1",
                       sep="\n")

  plot.grid <- marrangeGrob(grobs = collect.plots,
                            ncol=4,
                            nrow = 5,
                            top = textGrob(text.top,gp=gpar(fontsize=11,font=3)),
                            bottom = textGrob(text.bottom,gp=gpar(fontsize=11,font=3))
                            )

  collect.plot.grid <- append(collect.plot.grid, list(plot.grid))
}
```

```
collect.plot.grid[[1]]
```

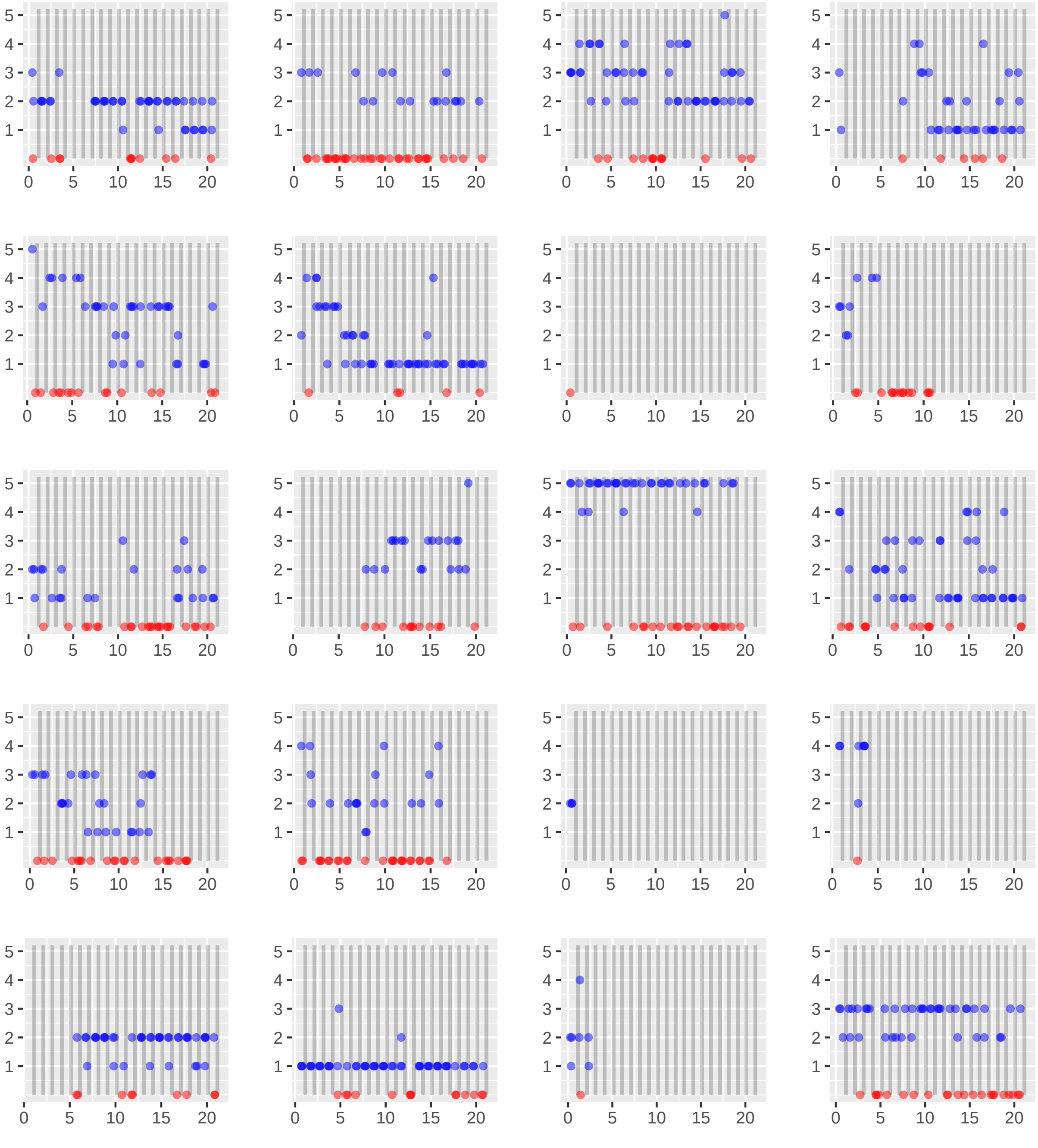
Affect6
Time of EMA delivery versus response on a 5−point Likert scale
All Random EMAs within 21−Day Post Quit Period



Shaded area denotes time between 10PM − 8AM
Each point denotes one random EMA
red dots: engaged.yes=0, blue dots: engaged.yes=1

```
collect.plot.grid[[2]]
```

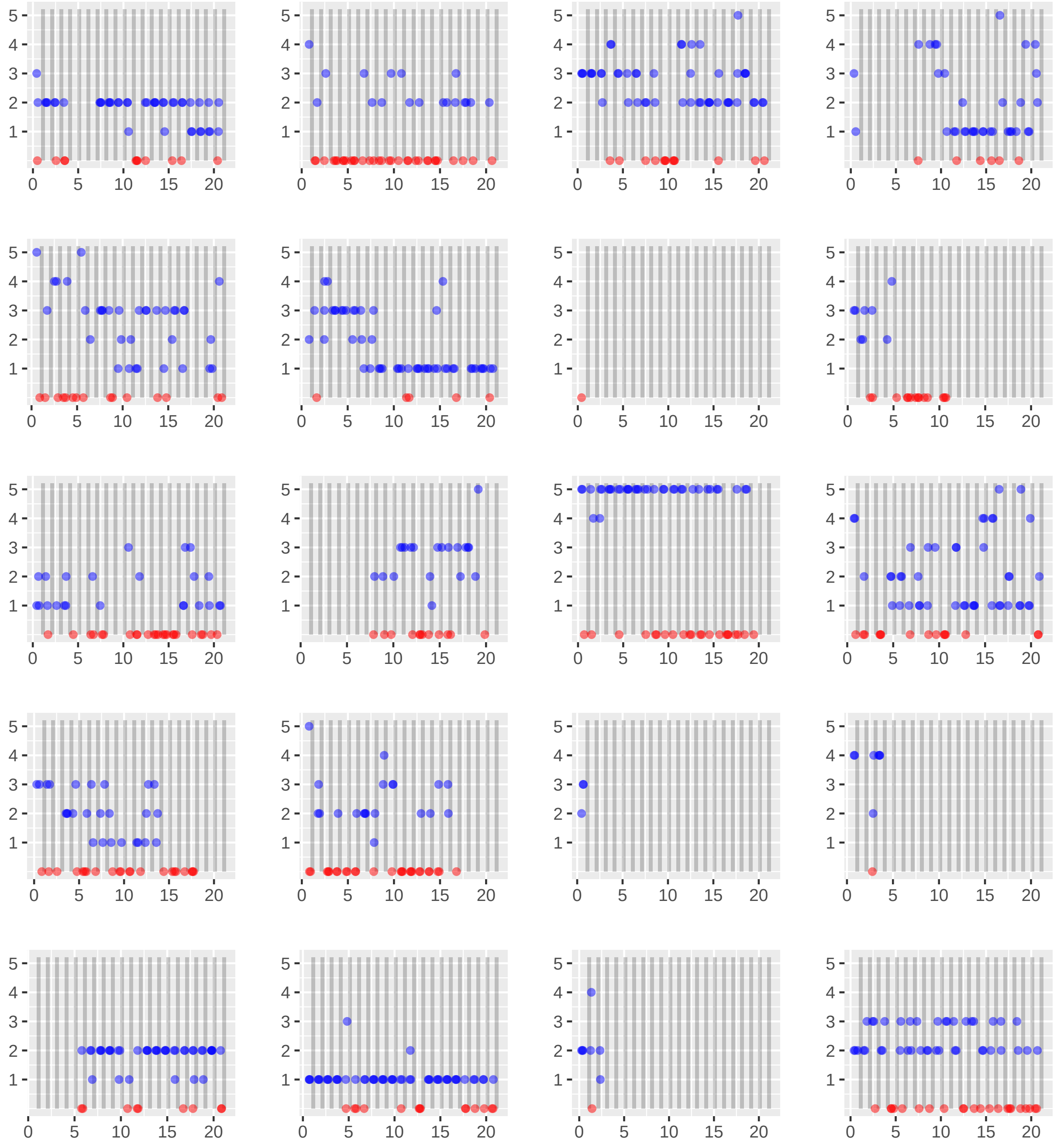
Affect7
Time of EMA delivery versus response on a 5−point Likert scale
All Random EMAs within 21−Day Post Quit Period



Shaded area denotes time between 10PM − 8AM
Each point denotes one random EMA
red dots: engaged.yes=0, blue dots: engaged.yes=1

```
collect.plot.grid[[3]]
```

Affect8
Time of EMA delivery versus response on a 5−point Likert scale
All Random EMAs within 21−Day Post Quit Period



Shaded area denotes time between 10PM − 8AM
Each point denotes one random EMA
red dots: engaged.yes=0, blue dots: engaged.yes=1