Script 3b

Pain progression: individual data

Peter Kamerman

06 December 2019

Contents

Analysis notes

Definitions of missingness

Data were regarded as **missing** when *pain in the last week* data were not present for one or more of weeks 0, 12, 24, 36, 48. Data also were classified as **missing** when there were inconsistencies in the data across the variables collected within a week.

Definition of data inconsistencies

Pain was defined as *pain in the last week* being 'Yes', and *pain at its worst* being > 0. These two measurements were then the "gatekeeper" measurements, such that the two measurements both had to be positive ('Yes' and '> 0', respectively) in order for there to be any entries for *site of pain* and *site of worst pain.* Were the data were inconsistent (e.g., when there was no *pain in the last week* and *pain at its worst* $= 0$, but there were entries for *site of pain* and *site of worst pain*), then the *site of pain* and *site of worst pain* entries were marked as **inconsistent**.

Data also were considered **inconsistent** when *pain in the last week* = 'Yes', but *site of worst pain* = 'None'.

Lastly, data were considered **inconsistent** when *site of worst pain* was not listed as one of the pain locations for a given measurement week.

For analysis purposes, missing data in the *site of pain* columns were changed to **'No'** (pain not present in the site). This approach was conservative, but we believed that the approach would have the least effect on the outcome, while still retaining as many participants as possible.

Import data

```
df <- read_rds('data-cleaned/data-ADVANCE.rds') %>%
    select(ranid, interval_name, pain_in_the_last_week,
           any_missing, interval_numeric)
```

Quick look

```
head(df)
```

```
## # A tibble: 6 x 5
##   ranid   interval_name pain_in_the_last_week any_missing interval_numeric
##   <chr>   <ord>         <chr>                 <chr>                  <dbl>
## 1 01-0001 0 weeks       No                    No                         0
## 2 01-0001 12 weeks      No                    No                        12
## 3 01-0001 24 weeks      No                    No                        24
## 4 01-0001 36 weeks      No                    No                        36
## 5 01-0001 48 weeks      No                    No                        48
## 6 01-0002 0 weeks       No                    No                         0
```

```
glimpse(df)
```

```
## Observations: 5,265
## Variables: 5
## $ ranid                 <chr> "01-0001", "01-0001", "01-0001", "01-000...
## $ interval_name         <ord> 0 weeks, 12 weeks, 24 weeks, 36 weeks, 4...
## $ pain_in_the_last_week <chr> "No", "No", "No", "No", "No", "No", "Yes...
## $ any_missing           <chr> "No", "No", "No", "No", "No", "No", "No"...
## $ interval_numeric      <dbl> 0, 12, 24, 36, 48, 0, 12, 24, 36, 48, 0,...
```

Basic clean

```
# Clean and process data
df %<>%
    filter(any_missing == 'No') %>%
    select(-any_missing) %>%
    rename(time_weeks = interval_numeric)
```

Quick tabulation

Analysis data set for the period 0 to 48 weeks

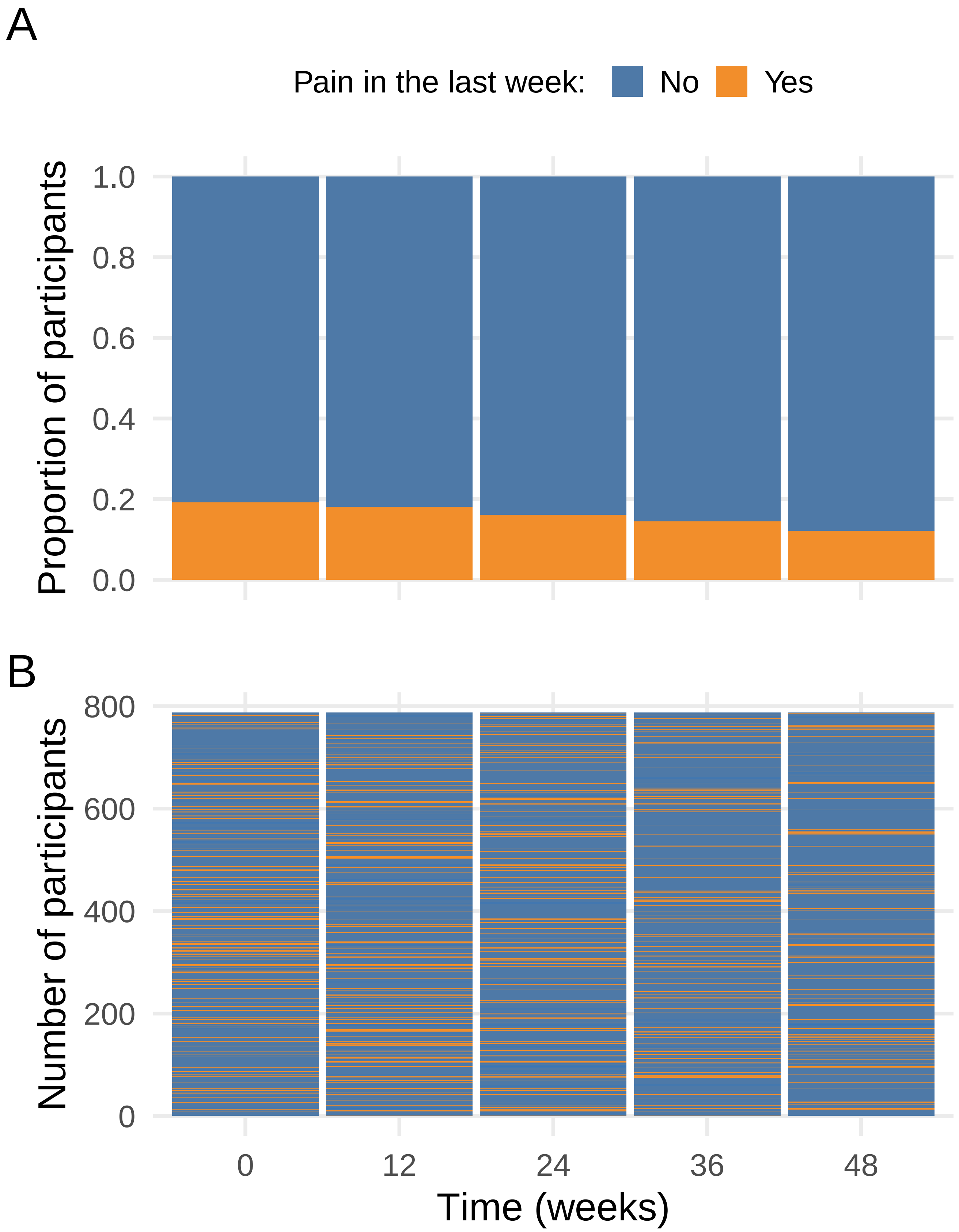
```
# Tabulate data
xtabs(~interval_name, data = df)
```

```
## interval_name
##  0 weeks 12 weeks 24 weeks 36 weeks 48 weeks
##      787      787      787      787      787
```

Pain per visit:

Plot summary data vs participant-level data

```r
# Breaks
n <- length(unique(df$ranid))

# Generate summary plot (for potential publication plot)
plot_summary <- ggplot(data = df) +
    aes(factor(time_weeks),
        fill = pain_in_the_last_week) +
    geom_bar(width = 0.95) +
    scale_fill_tableau(name = 'Pain in the last week: ') +
    scale_y_continuous(breaks = c(n * 0, n * 0.2, n * 0.4, n * 0.6, n * 0.8, n),
                       labels = c('0.0', '0.2', '0.4', '0.6', '0.8', '1.0')) +
    labs(x = 'Time (weeks)',
         y = 'Proportion of participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'top',
          legend.title = element_text(size = 16),
          legend.text = element_text(size = 16),
          panel.grid.minor = element_blank(),
          axis.text.x = element_blank(),
          axis.title.x = element_blank())

# Generate participant-level plot (for potential publication plot)
plot_individual <- df %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(time_weeks),
        y = id) +
    geom_tile(aes(fill = pain_in_the_last_week),
              width = 0.95) +
    scale_fill_tableau(name = 'Pain in the last week: ') +
    labs(x = 'Time (weeks)',
         y = 'Number of participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          panel.grid.minor = element_blank())

plot_individual2 <- df %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(time_weeks),
        y = id) +
    geom_tile(aes(fill = pain_in_the_last_week),
              width = 1) +
    scale_fill_tableau(name = 'Pain in the last week: ') +
    labs(x = 'Time (weeks)',
         y = 'Number of participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'top',
          panel.grid.minor = element_blank())

# Patchwork plot
plot_patchwork <- plot_summary + plot_individual +
    plot_layout(ncol = 1) + plot_annotation(tag_levels = 'A')

# Plot output
plot_patchwork
```



```r
# Save outcome
ggsave(filename = 'figures/figure-3.png', plot_individual2,
       width = 10.2, height = 7.72)
```

Tabulate pain progression patterns

```r
# Generate the pain sequences
df_sequence <- df %>%
    select(ranid, interval_name, pain_in_the_last_week) %>%
    mutate(pain_in_the_last_week = as.character(pain_in_the_last_week)) %>%
    pivot_wider(names_from = interval_name,
                values_from = pain_in_the_last_week) %>%
    unite(col = 'sequence', -ranid, sep = ' ') %>%
    mutate(any_pain = str_detect(sequence, pattern = 'Yes'))

# x-tabulate the sequences
df_xtab <- as.data.frame(xtabs(~sequence, data = df_sequence)) %>%
    arrange(desc(Freq)) %>%
    rename(frequency = Freq) %>%
    mutate(percent = round(100 * frequency / sum(frequency), 2))

# Print table
knitr::kable(df_xtab,
             caption = 'Pain in the last week for weeks 0, 12, 24, 36 and 48')
```

Table 1: Pain in the last week for weeks 0, 12, 24, 36 and 48

sequence	frequency	percent
No No No No No	393	49.94
Yes No No No No	67	8.51
No Yes No No No	50	6.35
No No Yes No No	44	5.59
No No No Yes No	32	4.07
No No No No Yes	28	3.56
Yes Yes No No No	22	2.80
Yes No Yes No No	16	2.03
No Yes No Yes No	15	1.91
No No No Yes Yes	13	1.65
No No Yes Yes No	13	1.65
No Yes Yes No No	13	1.65
No Yes No No Yes	10	1.27
Yes No No No Yes	10	1.27
No Yes Yes Yes No	9	1.14
No No Yes No Yes	8	1.02
Yes No No Yes No	6	0.76
Yes Yes No Yes Yes	6	0.76
Yes No Yes No Yes	4	0.51
Yes No Yes Yes No	4	0.51
No No Yes Yes Yes	3	0.38
No Yes No Yes Yes	3	0.38
Yes Yes Yes No No	3	0.38
Yes Yes Yes No Yes	3	0.38
No Yes Yes Yes Yes	2	0.25
Yes No Yes Yes Yes	2	0.25
Yes Yes No No Yes	2	0.25
Yes Yes No Yes No	2	0.25
Yes Yes Yes Yes No	2	0.25

sequence	frequency	percent
Yes No No Yes Yes	1	0.13
Yes Yes Yes Yes Yes	1	0.13

Number of sequences with 'yes' in series

```r
# Extract sequences
df_yes <- df_sequence %>%
    mutate(yes_2 = str_detect(sequence, pattern = 'Yes Yes')) %>%
    mutate(yes_3 = str_detect(sequence, pattern = 'Yes Yes Yes')) %>%
    mutate(yes_4 = str_detect(sequence, pattern = 'Yes Yes Yes Yes')) %>%
    mutate(yes_5 = str_detect(sequence, pattern = 'Yes Yes Yes Yes Yes'))

# Create filters
vec_filter_2yes <- df_yes %>%
    filter(yes_2 == TRUE) %>%
    .$ranid

vec_filter_3yes <- df_yes %>%
    filter(yes_3 == TRUE) %>%
    .$ranid

vec_filter_4yes <- df_yes %>%
    filter(yes_4 == TRUE) %>%
    .$ranid

vec_filter_5yes <- df_yes %>%
    filter(yes_5 == TRUE) %>%
    .$ranid

# Extract data
df_2yes <- df_yes %>%
    filter(!ranid %in% vec_filter_5yes) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(!ranid %in% vec_filter_3yes) %>%
    filter(ranid %in% vec_filter_2yes)

df_3yes <- df_yes %>%
    filter(!ranid %in% vec_filter_5yes) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(ranid %in% vec_filter_3yes)

df_4yes <- df_yes %>%
    filter(!ranid %in% vec_filter_5yes) %>%
    filter(ranid %in% vec_filter_4yes)

df_5yes <- df_yes %>%
    filter(ranid %in% vec_filter_5yes)

# Sequences with 2 'yes' in series ONLY
knitr::kable(data.frame('type' = c('Sequence count', 'Percent of total sequences'),
                        'value' = c(round(sum(df_2yes$yes_2)),
                                    round(100 * (sum(df_2yes$yes_2) /
                                                     sum(df_xtab$frequency)), 2))),
             caption = "Sequences with 2 'yes' (pain present) in series")
```

Table 2: Sequences with 2 ‘yes’ (pain present) in series

type	value
Sequence count	79.00
Percent of total sequences	10.04

```r
# Sequences with 3 'yes' (pain present) in series ONLY
knitr::kable(data.frame('type' = c('Sequence count', 'Percent of total sequences'),
                        'value' = c(round(sum(df_3yes$yes_3)),
                                    round(100 * (sum(df_3yes$yes_3) /
                                                 sum(df_xtab$frequency)), 2))),
             caption = "Sequences with 3 'yes' (pain present) in series")
```

Table 3: Sequences with 3 ‘yes’ (pain present) in series

type	value
Sequence count	20.00
Percent of total sequences	2.54

```r
# Sequences with 4 'yes' (pain present) in series ONLY
knitr::kable(data.frame('type' = c('Sequence count', 'Percent of total sequences'),
                        'value' = c(round(sum(df_4yes$yes_4)),
                                    round(100 * (sum(df_4yes$yes_4) /
                                                 sum(df_xtab$frequency)), 2))),
             caption = "Sequences with at least 4 'yes' (pain present) in series")
```

Table 4: Sequences with at least 4 ‘yes’ (pain present) in series

type	value
Sequence count	4.00
Percent of total sequences	0.51

```r
# Sequences with at least 5 'yes' (pain present) in series
knitr::kable(data.frame('type' = c('Sequence count', 'Percent of total sequences'),
                        'value' = c(round(sum(df_5yes$yes_5)),
                                    round(100 * (sum(df_5yes$yes_5) /
                                                 sum(df_xtab$frequency)), 2))),
             caption = "Sequences with at least 5 'yes' (pain present) in series")
```

Table 5: Sequences with at least 5 ‘yes’ (pain present) in series

type	value
Sequence count	1.00
Percent of total sequences	0.13

Session information

```r
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] patchwork_0.0.1 ggthemes_4.2.0  magrittr_1.5    forcats_0.4.0
##  [5] stringr_1.4.0   dplyr_0.8.3     purrr_0.3.3     readr_1.3.1
##  [9] tidyr_1.0.0     tibble_2.1.3    ggplot2_3.2.1   tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5 xfun_0.10        haven_2.1.1      lattice_0.20-38
##  [5] colorspace_1.4-1 vctrs_0.2.0      generics_0.0.2   htmltools_0.4.0
##  [9] yaml_2.2.0       utf8_1.1.4       rlang_0.4.0      pillar_1.4.2
## [13] glue_1.3.1       withr_2.1.2      modelr_0.1.5     readxl_1.3.1
## [17] lifecycle_0.1.0  munsell_0.5.0    gtable_0.3.0     cellranger_1.1.0
## [21] rvest_0.3.4      evaluate_0.14    labeling_0.3     knitr_1.25
## [25] fansi_0.4.0      highr_0.8        broom_0.5.2      Rcpp_1.0.2
## [29] scales_1.0.0     backports_1.1.5  jsonlite_1.6     hms_0.5.1
## [33] digest_0.6.22    stringi_1.4.3    grid_3.6.1       cli_1.1.0
## [37] tools_3.6.1      lazyeval_0.2.2   crayon_1.3.4     pkgconfig_2.0.3
## [41] zeallot_0.1.0    ellipsis_0.3.0   xml2_1.2.2       lubridate_1.7.4
## [45] assertthat_0.2.1 rmarkdown_1.16   httr_1.4.1       rstudioapi_0.10
## [49] R6_2.4.0         nlme_3.1-141     compiler_3.6.1
```