Script 8

Treatment group publication plots

Peter Kamerman

08 August 2020

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,
           group, pain_in_the_last_week, pain_worst,
           any_missing, interval_numeric)
```

Quick look

```
head(df)
```

```
## # A tibble: 6 x 7
##   ranid interval_name group pain_in_the_las~ pain_worst any_missing
##   <chr> <ord>         <chr> <chr>                 <dbl> <chr>
## 1 01-0~ 0 weeks       DTG ~ No                        0 No
## 2 01-0~ 12 weeks      DTG ~ No                        0 No
## 3 01-0~ 24 weeks      DTG ~ No                        0 No
## 4 01-0~ 36 weeks      DTG ~ No                        0 No
## 5 01-0~ 48 weeks      DTG ~ No                        0 No
## 6 01-0~ 0 weeks       DTG ~ No                        0 No
## # ... with 1 more variable: interval_numeric <dbl>
```

```
glimpse(df)
```

```
## Rows: 5,265
## Columns: 7
## $ ranid                 <chr> "01-0001", "01-0001", "01-0001", "01-0001", "...
## $ interval_name         <ord> 0 weeks, 12 weeks, 24 weeks, 36 weeks, 48 wee...
## $ group                 <chr> "DTG + TAF + FTC", "DTG + TAF + FTC", "DTG + ...
## $ pain_in_the_last_week <chr> "No", "No", "No", "No", "No", "No", "Yes", "Y...
## $ pain_worst            <dbl> 0, 0, 0, 0, 0, 0, 3, 3, 5, 0, 0, 0, 0, 0, 0, ...
## $ any_missing           <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ interval_numeric      <dbl> 0, 12, 24, 36, 48, 0, 12, 24, 36, 48, 0, 12, ...
```

Basic clean

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

Quick tabulations

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
```

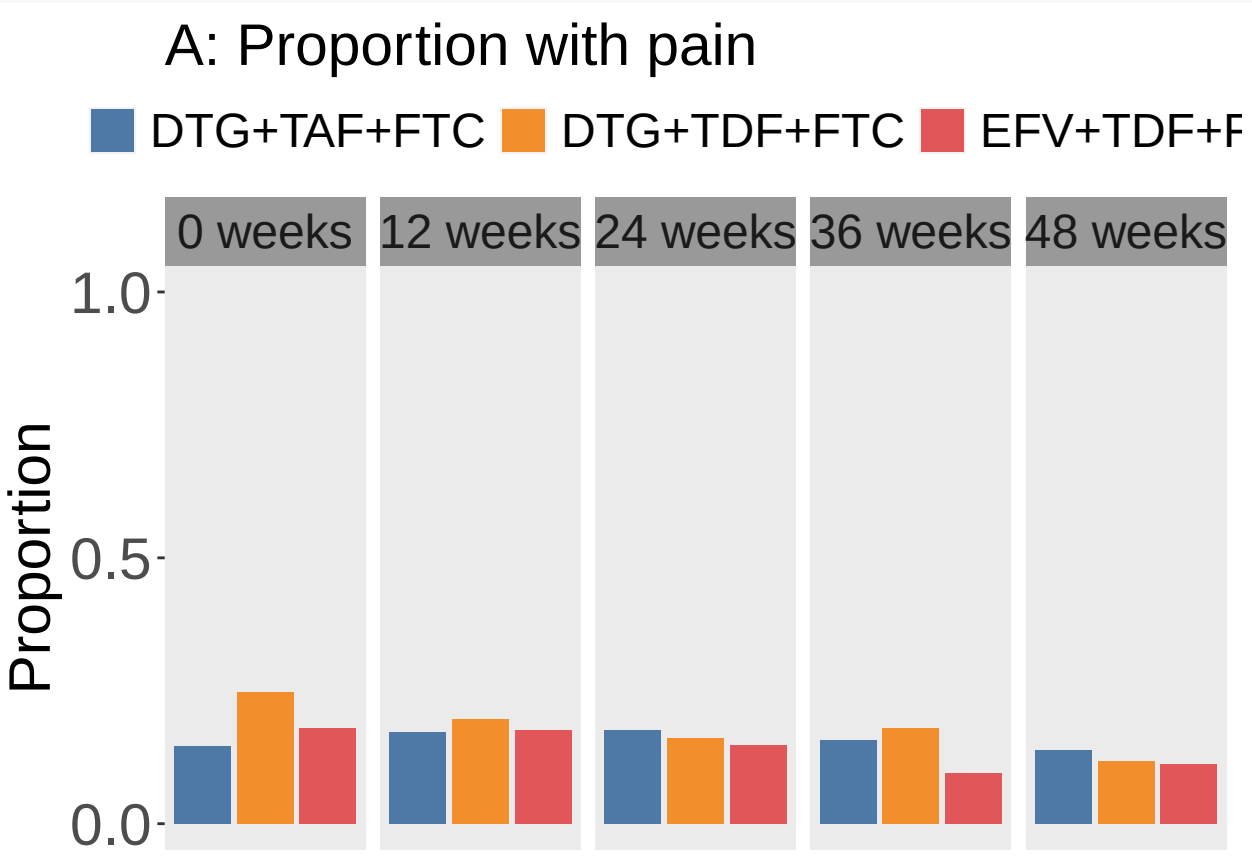
Plots

Analysis

Pain frequency by group

```
# Process data
freq <- df %>%
  select(interval_name, group, pain_in_the_last_week) %>%
  mutate(group = str_remove_all(group, pattern = ' ')) %>%
  mutate(pain_in_the_last_week = pain_in_the_last_week == 'Yes') %>%
  group_by(interval_name, group) %>%
  summarise(count = sum(pain_in_the_last_week == TRUE),
            total = sum(count + sum(pain_in_the_last_week == FALSE)),
            proportion = mean(pain_in_the_last_week)) %>%
  ungroup()

# Plot data
p_proportion <- freq %>%
  ggplot(data = .) +
  aes(x = group,
      y = proportion,
      fill = group) +
  geom_col() +
  labs(title = 'A: Proportion with pain',
       y = 'Proportion') +
  scale_y_continuous(limits = c(0, 1),
                     breaks = c(0, 0.5, 1)) +
  scale_fill_tableau() +
  #scale_fill_manual(values = c('#0a4264', '#1170aa', '#1f9ce9')) +
  theme(axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        plot.title = element_text(size = 22),
        axis.title = element_text(size = 22),
        axis.text.y = element_text(size = 22),
        axis.text.x = element_blank(),
        strip.background = element_rect(fill = '#999999'),
        strip.text = element_text(size = 18),
        panel.grid = element_blank(),
        legend.position = 'top',
        legend.text = element_text(size = 18),
        legend.title = element_blank()) +
  facet_grid(.~interval_name); p_proportion
```

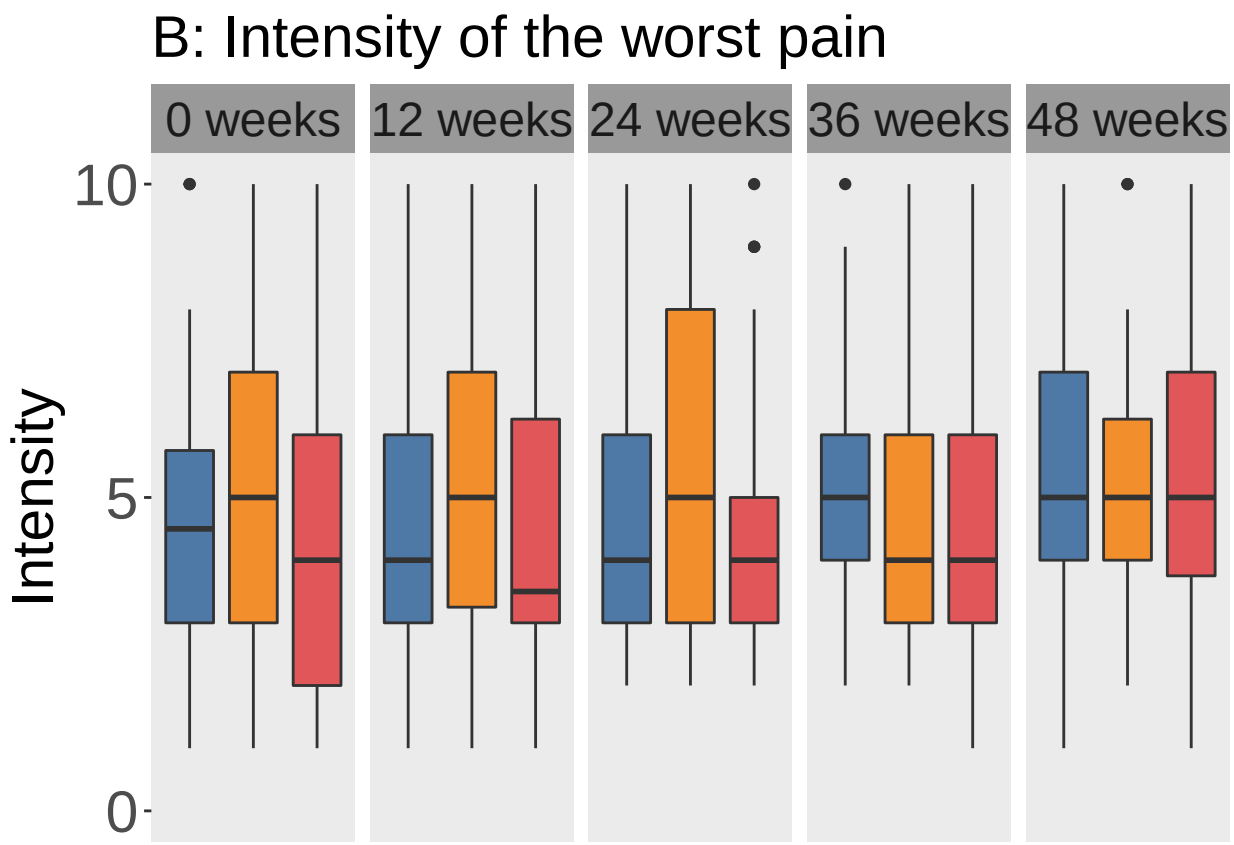


Pain intensity by group

```
# Process data
intensity <- df %>%
  select(interval_name, group, pain_worst) %>%
  mutate(group = str_remove_all(group, pattern = ' ')) %>%
  filter(pain_worst > 0)

# Plot data
p_intensity <- intensity %>%
  ggplot(data = .) +
  aes(x = group,
      y = pain_worst,
      fill = group) +
  geom_boxplot() +
  labs(title = 'B: Intensity of the worst pain',
       y = 'Intensity') +
  scale_y_continuous(limits = c(0, 10),
                     breaks = c(0, 5, 10)) +
  theme(axis.text.x = element_text(angle = 90,
                                   hjust = 0),
        axis.title.x = element_blank()) +
  scale_fill_tableau() +
  #scale_fill_manual(values = c('#0a4264', '#1170aa', '#1f9ce9')) +
  theme(axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        plot.title = element_text(size = 22),
        axis.title = element_text(size = 22),
        axis.text.y = element_text(size = 22),
```

```
        axis.text.x = element_blank(),
        strip.background = element_rect(fill = '#999999'),
        strip.text = element_text(size = 18),
        panel.grid = element_blank(),
        legend.position = 'none') +
  facet_grid(.~interval_name); p_intensity
```



Publication plot

```
pp_plot <- p_proportion + p_intensity + plot_layout(ncol = 1)

ggsave(x = pp_plot,
       filename = 'figures/figure-1.png',
       height = 8, width = 12)
```

Session information

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.5
##
## Matrix products: default
```

```
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/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] knitr_1.29      patchwork_1.0.1 ggthemes_4.2.0  magrittr_1.5
##  [5] forcats_0.5.0   stringr_1.4.0   dplyr_1.0.0     purrr_0.3.4
##  [9] readr_1.3.1     tidyr_1.1.0     tibble_3.0.1    ggplot2_3.3.2
## [13] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0 xfun_0.15        haven_2.3.1      lattice_0.20-41
##  [5] colorspace_1.4-1 vctrs_0.3.1      generics_0.0.2   htmltools_0.5.0
##  [9] yaml_2.2.1       utf8_1.1.4       blob_1.2.1       rlang_0.4.6
## [13] pillar_1.4.4     glue_1.4.1       withr_2.2.0      DBI_1.1.0
## [17] dbplyr_1.4.4     modelr_0.1.8     readxl_1.3.1     lifecycle_0.2.0
## [21] munsell_0.5.0    gtable_0.3.0     cellranger_1.1.0 rvest_0.3.5
## [25] evaluate_0.14    fansi_0.4.1      broom_0.5.6      Rcpp_1.0.4.6
## [29] scales_1.1.1     backports_1.1.8  jsonlite_1.6.1   farver_2.0.3
## [33] fs_1.4.1         hms_0.5.3        digest_0.6.25    stringi_1.4.6
## [37] grid_4.0.2       cli_2.0.2        tools_4.0.2      crayon_1.3.4
## [41] pkgconfig_2.0.3  ellipsis_0.3.1   xml2_1.3.2       reprex_0.3.0
## [45] lubridate_1.7.9  assertthat_0.2.1 rmarkdown_2.3    httr_1.4.1
## [49] rstudioapi_0.11  R6_2.4.1         nlme_3.1-148     compiler_4.0.2
```