

Script 6

Group-level publication plots

Peter Kamerman

08 August 2020

Contents

Analysis notes	1
Definitions of missingness	1
Definition of data inconsistencies	2
Import data	2
Quick look	2
Basic clean	3
Quick tabulations	3
Analysis data set for the period 0 to 48 weeks	3
Extract data	3
Proportion with pain	3
Intensity of worst pain	3
Number of pain sites	4
Site of worst pain	4
Plots	4
Proportion with pain	4
Intensity of worst pain	5
Number of pain sites	6
Prevalence of pain sites	7
Site of worst pain	8
Publication plots	9
Session information	10

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, pain_worst,
         site_worst, ends_with('_pain'), any_missing)

df_2 <- 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 19
##   ranid interval_name pain_in_the_las~ pain_worst site_worst head_pain
##   <chr> <ord>         <chr>          <dbl> <chr>      <chr>
## 1 01-0~ 0 weeks      No              0 None      No
## 2 01-0~ 12 weeks     No              0 None      No
## 3 01-0~ 24 weeks     No              0 None      No
## 4 01-0~ 36 weeks     No              0 None      No
## 5 01-0~ 48 weeks     No              0 None      No
## 6 01-0~ 0 weeks      No              0 None      No
## # ... with 13 more variables: cervical_pain <chr>, shoulder_pain <chr>,
## #   arm_pain <chr>, hand_pain <chr>, chest_pain <chr>, abdominal_pain <chr>,
## #   low_back_pain <chr>, buttock_pain <chr>, hip_groin_pain <chr>,
## #   leg_pain <chr>, genital_pain <chr>, foot_pain <chr>, any_missing <chr>
```

```
glimpse(df)
```

```
## Rows: 5,265
## Columns: 19
## $ ranid          <chr> "01-0001", "01-0001", "01-0001", "01-0001", "...
## $ interval_name  <ord> 0 weeks, 12 weeks, 24 weeks, 36 weeks, 48 wee...
## $ 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, ...
## $ site_worst     <chr> "None", "None", "None", "None", "None", "None", "None..."
```

```
## $ head_pain      <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ cervical_pain  <chr> "No", "No", "No", "No", "No", "No", "No", "Ye...
## $ shoulder_pain  <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ arm_pain        <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ hand_pain       <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ chest_pain      <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ abdominal_pain <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ low_back_pain   <chr> "No", "No", "No", "No", "No", "No", "No", "Ye...
## $ buttock_pain    <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ hip_groin_pain  <chr> "No", "No", "No", "No", "No", "No", "Yes", "N...
## $ leg_pain        <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ genital_pain    <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ foot_pain       <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ any_missing     <chr> "No", "No", "No", "No", "No", "No", "No", "No..."
```

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
```

Extract data

Proportion with pain

```
df_proportion <- df %>%
  select(ranid, interval_name, pain_in_the_last_week, pain_worst) %>%
  mutate(pain = ifelse(pain_in_the_last_week == 'Yes' & pain_worst > 0,
    yes = 'Yes',
    no = 'No')) %>%
  select(-pain_in_the_last_week, -pain_worst)
```

Intensity of worst pain

```
df_intensity <- df %>%
  select(ranid, interval_name, pain_in_the_last_week, pain_worst) %>%
```

```
filter(pain_in_the_last_week == 'Yes' & pain_worst > 0) %>%
select(-pain_in_the_last_week)
```

Number of pain sites

```
df_sites <- df %>%
  select(ranid, interval_name, pain_in_the_last_week, pain_worst, ends_with('_pain')) %>%
  filter(pain_in_the_last_week == 'Yes' & pain_worst > 0) %>%
  select(-pain_in_the_last_week, -pain_worst)
```

Site of worst pain

```
df_worst_site <- df %>%
  select(ranid, interval_name, pain_in_the_last_week, pain_worst, site_worst) %>%
  filter(pain_in_the_last_week == 'Yes' & pain_worst > 0) %>%
  select(-pain_in_the_last_week, -pain_worst)
```

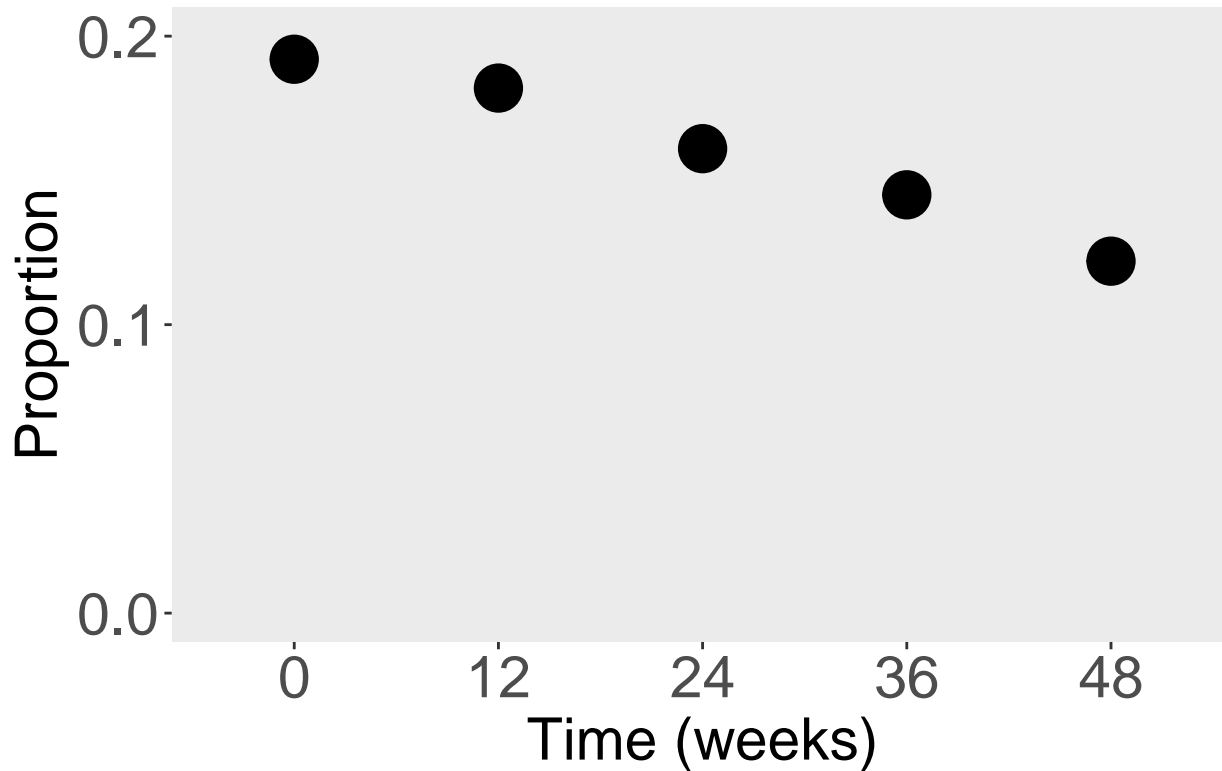
Plots

Proportion with pain

```
# Process data
df_proportion %<>%
  group_by(interval_name, pain) %>%
  summarise(count = n()) %>%
  group_by(interval_name) %>%
  mutate(total = sum(count)) %>%
  mutate(proportion = round(count / total, 3)) %>%
  filter(pain == 'Yes') %>%
  ungroup()

# Plot
p_proportion <- df_proportion %>%
  ggplot(data = .) +
  aes(x = interval_name,
      y = proportion) +
  geom_point(size = 8) +
  labs(title = 'A: Proportion with pain',
       y = 'Proportion',
       x = 'Time (weeks)') +
  scale_y_continuous(limits = c(0, 0.2),
                    breaks = c(0, 0.1, 0.2)) +
  scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
  theme(plot.title = element_text(size = 22),
        axis.title = element_text(size = 22),
        axis.text = element_text(size = 22),
        panel.grid = element_blank()); p_proportion
```

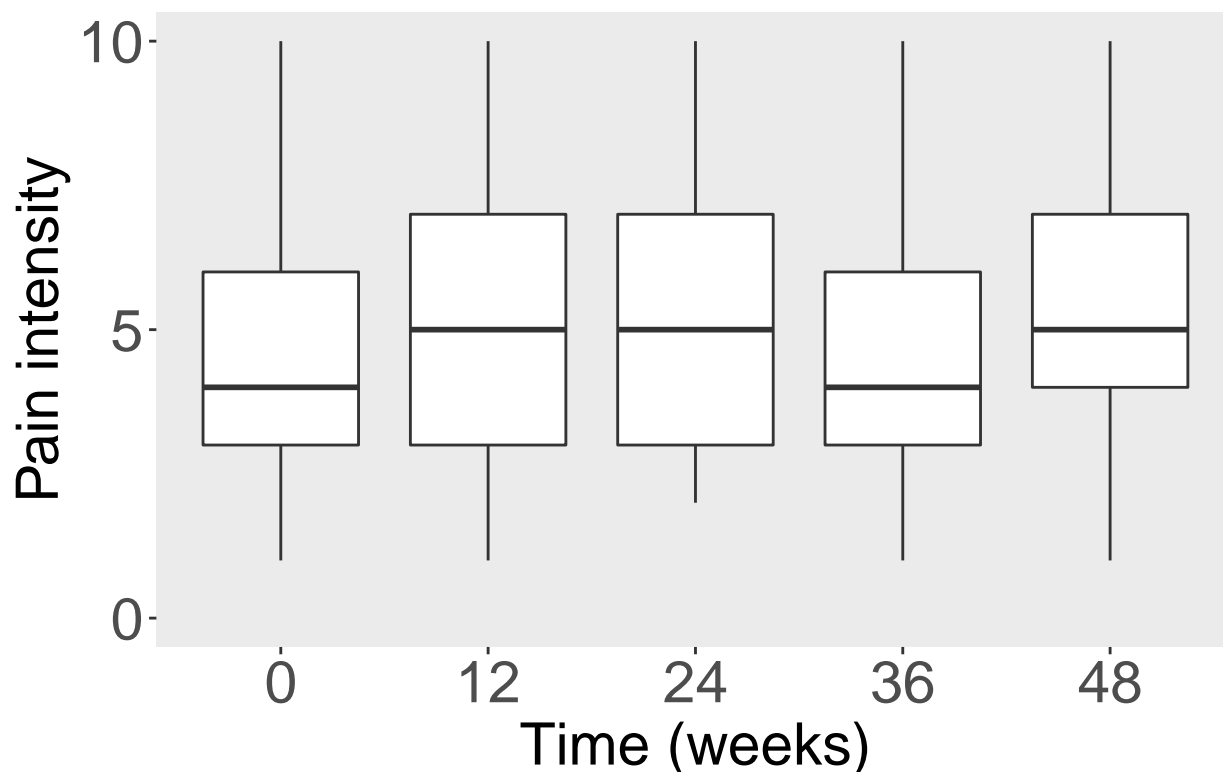
A: Proportion with pain



Intensity of worst pain

```
# Plot data
p_intensity <- df_intensity %>%
  ggplot(data = .) +
  aes(x = interval_name,
      y = pain_worst) +
  geom_boxplot() +
  labs(title = 'B: Intensity of the worst pain',
       y = 'Pain intensity',
       x = 'Time (weeks)') +
  scale_y_continuous(limits = c(0, 10),
                    breaks = c(0, 5, 10)) +
  scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
  theme(plot.title = element_text(size = 22),
        axis.title = element_text(size = 22),
        axis.text = element_text(size = 22),
        panel.grid = element_blank()); p_intensity
```

B: Intensity of the worst pain

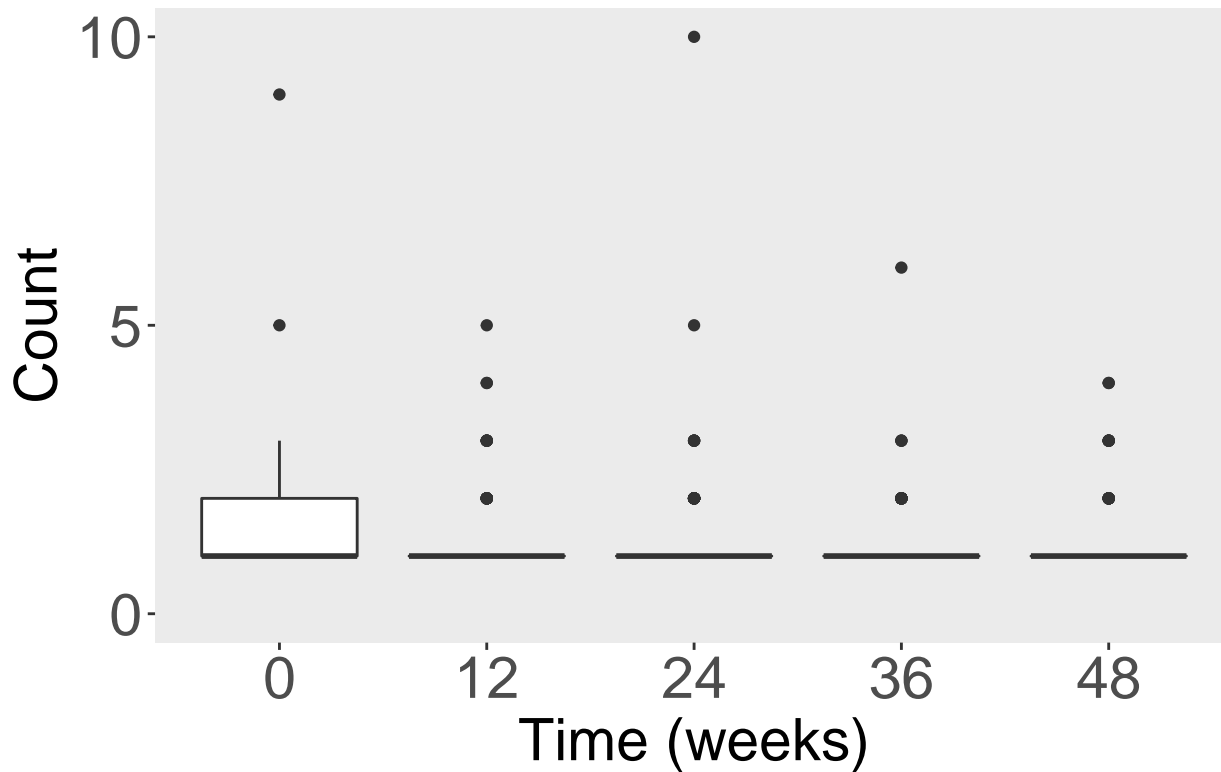


Number of pain sites

```
# Process data
sites <- df_sites %>%
  pivot_longer(cols = ends_with('_pain'),
    names_to = 'sites',
    values_to = 'values') %>%
  mutate(values = ifelse(values == 'Yes',
    yes = TRUE,
    no = FALSE)) %>%
  group_by(ranid, interval_name) %>%
  summarise(sum = sum(values)) %>%
  ungroup()

# Plot
p_sites <- sites %>%
  ggplot(data = .) +
  aes(x = interval_name,
    y = sum) +
  geom_boxplot() +
  labs(title = 'C: Number of pain sites',
    y = 'Count',
    x = 'Time (weeks)') +
  scale_y_continuous(limits = c(0, 10),
    breaks = c(0, 5, 10)) +
  scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
  theme(axis.title = element_text(size = 22),
    plot.title = element_text(size = 22),
    axis.text = element_text(size = 22),
    panel.grid = element_blank()); p_sites
```

C: Number of pain sites



Prevalence of pain sites

```
# Process data
prevalence <- df_sites %>%
  pivot_longer(col = ends_with('_pain'),
               names_to = 'key',
               values_to = 'values') %>%
  group_by(interval_name, key, values) %>%
  summarise(count = n()) %>%
  group_by(interval_name, key) %>%
  mutate(total = sum(count),
         prop = count / total) %>%
  filter(values != 'No') %>%
  filter(prop >= 0.10) %>%
  ungroup() %>%
  mutate(key = factor(key,
                      levels = c('leg_pain', 'chest_pain', 'foot_pain',
                                'low_back_pain', 'head_pain', 'abdominal_pain'),
                      labels = c('Leg', 'Chest', 'Foot',
                                'Low back', 'Head', 'Abdomen'),
                      ordered = TRUE))

# Plot
p_prevalence <- ggplot(data = prevalence) +
  aes(x = interval_name,
      y = key,
      fill = key) +
  geom_tile(colour = '#FFFFFF', size = 1) +
  labs(title = 'D: Common pain sites',
```

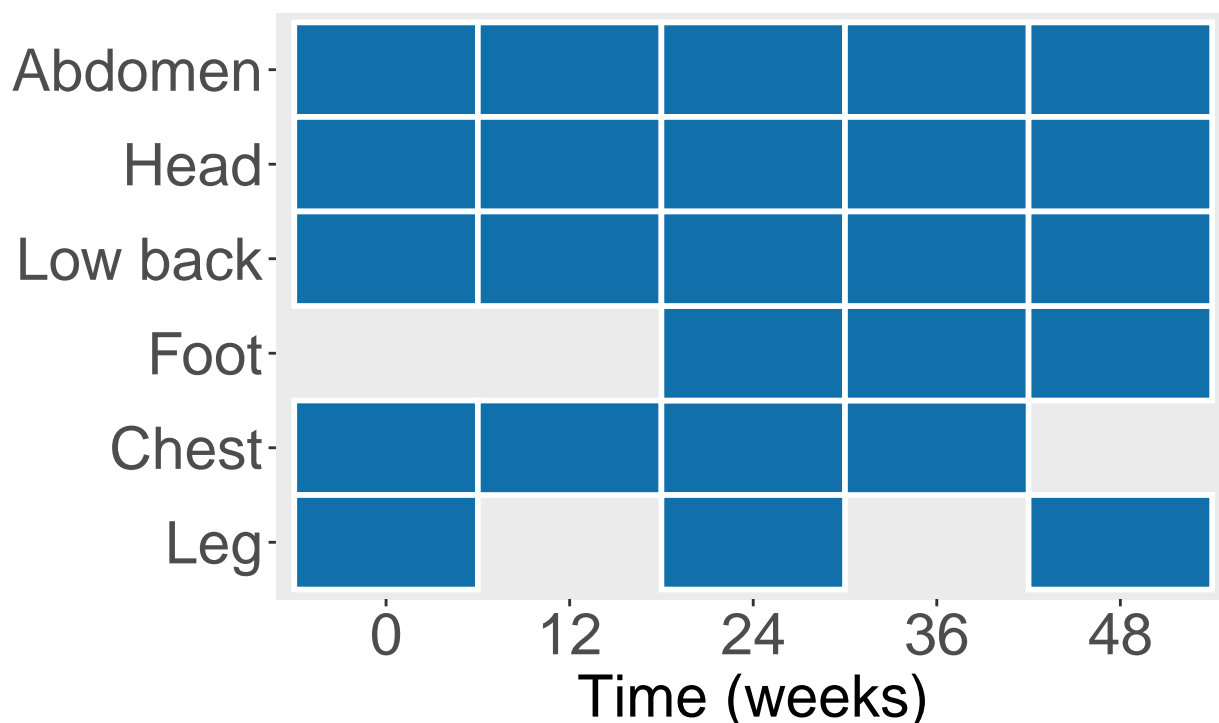
```

    subtitle = expression('(prevalence ' >= '10%)'),
    x = 'Time (weeks)' +
scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
scale_fill_manual(values = rep(c('#1170aa'), 6)) +
theme(legend.position = 'none',
      plot.title = element_text(size = 22),
      plot.subtitle = element_text(size = rel(1)),
      legend.title = element_blank(),
      axis.title.y = element_blank(),
      axis.title.x = element_text(size = 22),
      axis.text = element_text(size = 22),
      panel.grid = element_blank()); p_prevalence

```

D: Common pain sites

(prevalence $\geq 10\%$)



Site of worst pain

```

# Process data
df_worst_site %<>%
  group_by(interval_name, site_worst) %>%
  summarise(count = n()) %>%
  group_by(interval_name) %>%
  mutate(total = sum(count),
         proportion = count / total) %>%
  filter(proportion >= 0.10) %>%
  ungroup() %>%
  mutate(site_worst = factor(site_worst,
                             levels = c('Leg', 'Feet', 'Chest',
                                           'Low back', 'Head', 'Abdomen'),
                             labels = c('Leg', 'Foot', 'Chest',
                                           'Low back', 'Head', 'Abdomen'),
                             ordered = TRUE))

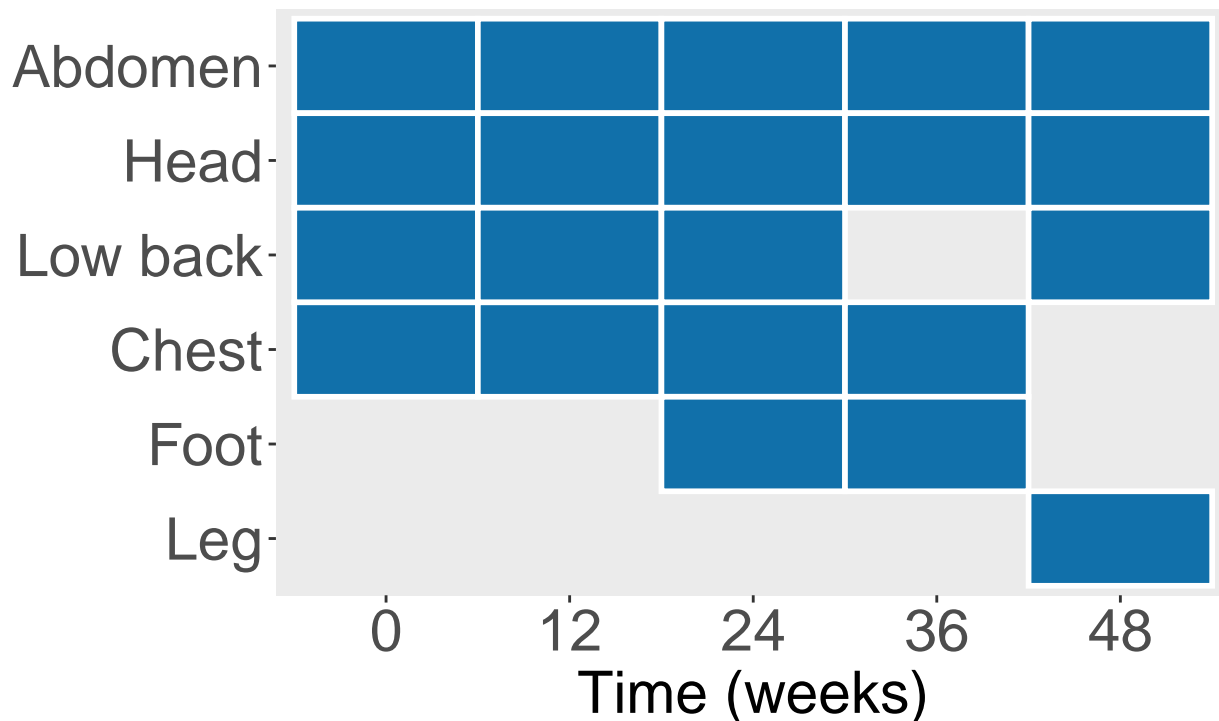
```



```
# Plot
p_worst_sites <- df_worst_site %>%
  ggplot(data = .) +
  aes(x = interval_name,
      y = site_worst,
      fill = site_worst) +
  geom_tile(colour = '#FFFFFF',
            size = 1) +
  labs(title = 'D: Sites of the worst pain',
       subtitle = expression('(prevalence ' >= '10%)'),
       y = 'Proportion',
       x = 'Time (weeks)' +
  scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
  scale_fill_manual(values = rep(c('#1170aa'), 6)) +
  theme(legend.position = 'none',
       plot.title = element_text(size = 22),
       plot.subtitle = element_text(size = rel(1)),
       legend.title = element_blank(),
       axis.title.y = element_blank(),
       axis.title.x = element_text(size = 22),
       axis.text = element_text(size = 22),
       panel.grid = element_blank()); p_worst_sites
```

D: Sites of the worst pain

(prevalence $\geq 10\%$)



Publication plots

```
# With p_prevalence
pp_plot2 <- p_proportion + p_intensity + p_sites + p_worst_sites +
```

```
plot_layout(ncol = 2)

ggsave(x = pp_plot2,
       filename = 'figures/figure-2.png',
       height = 10, width = 11)
```

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 vctr_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  http_1.4.1
## [49] rstudioapi_0.11 R6_2.4.1       nlme_3.1-148   compiler_4.0.2
```