Script 6

Individual-level pub;ication plots

Peter Kamerman

22 January 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, pain_in_the_last_week, pain_worst,
           site_worst, ends_with('_pain'), any_missing, interval_numeric)
```

Quick look

```
head(df)
```

```
## # A tibble: 6 x 20
##   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 14 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>, interval_numeric <dbl>
```

```
glimpse(df)
```

```
## Observations: 5,265
## Variables: 20
## $ ranid                 <chr> "01-0001", "01-0001", "01-0001", "01-0001"…
## $ interval_name         <ord> 0 weeks, 12 weeks, 24 weeks, 36 weeks, 48 …
## $ pain_in_the_last_week <chr> "No", "No", "No", "No", "No", "No", "Yes",…
## $ 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", "N…
## $ head_pain             <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ cervical_pain         <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ shoulder_pain         <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ arm_pain              <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ hand_pain             <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ chest_pain            <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ abdominal_pain        <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ low_back_pain         <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ buttock_pain          <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ hip_groin_pain        <chr> "No", "No", "No", "No", "No", "No", "Yes",…
## $ leg_pain              <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ genital_pain          <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ foot_pain             <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ any_missing           <chr> "No", "No", "No", "No", "No", "No", "No", …
## $ interval_numeric      <dbl> 0, 12, 24, 36, 48, 0, 12, 24, 36, 48, 0, 1…
```

Basic clean

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

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
```

Prepare "pain present" filters

```
# Generate the pain sequences
df_sequence <- df %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    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'))

# 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'))

# 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
```

Extract data

Pain intensity and affected sites

```r
## 2 yes's in a row ONLY
df_2yes <- df %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(!ranid %in% vec_filter_3yes) %>%
    filter(ranid %in% vec_filter_2yes) %>%
    mutate(intensity = case_when(
        pain_worst == 0 ~ 'None',
        pain_worst > 0 & pain_worst < 4 ~ 'Mild',
        pain_worst > 3 & pain_worst < 8 ~ 'Moderate',
        pain_worst > 7 & pain_worst <= 10 ~ 'Severe'
    ),
    intensity = factor(intensity,
                       levels = c('None', 'Mild', 'Moderate', 'Severe'),
                       ordered = TRUE)) %>%
    mutate(site_worst = factor(site_worst,
                               levels = c('None', 'Abdomen', 'Arm', 'Buttocks', 'Chest',
                                          'Feet', 'Genitals', 'Hand', 'Head',
                                          'Hip/groin', 'Leg', 'Low back',
                                          'Neck', 'Shoulder'),
                               ordered = TRUE))

## 3 yes's in a row ONLY
df_3yes <- df %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(ranid %in% vec_filter_3yes) %>%
    mutate(intensity = case_when(
        pain_worst == 0 ~ 'None',
        pain_worst > 0 & pain_worst < 4 ~ 'Mild',
        pain_worst > 3 & pain_worst < 8 ~ 'Moderate',
        pain_worst > 7 & pain_worst <= 10 ~ 'Severe'
    ),
    intensity = factor(intensity,
                       levels = c('None', 'Mild', 'Moderate', 'Severe'),
                       ordered = TRUE)) %>%
    mutate(site_worst = factor(site_worst,
                               levels = c('None', 'Abdomen', 'Arm', 'Buttocks', 'Chest',
                                          'Feet', 'Genitals', 'Hand', 'Head',
                                          'Hip/groin', 'Leg', 'Low back',
                                          'Neck', 'Shoulder'),
                               ordered = TRUE))

## At least 4 yes's in a row
df_4yes <- df %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(ranid %in% vec_filter_4yes) %>%
    mutate(intensity = case_when(
        pain_worst == 0 ~ 'None',
        pain_worst > 0 & pain_worst < 4 ~ 'Mild',
        pain_worst > 3 & pain_worst < 8 ~ 'Moderate',
        pain_worst > 7 & pain_worst <= 10 ~ 'Severe'
    ),
    intensity = factor(intensity,
                       levels = c('None', 'Mild', 'Moderate', 'Severe'),
                       ordered = TRUE)) %>%
    mutate(site_worst = factor(site_worst,
                               levels = c('None', 'Abdomen', 'Arm', 'Buttocks', 'Chest',
                                          'Feet', 'Genitals', 'Hand', 'Head',
                                          'Hip/groin', 'Leg', 'Low back',
                                          'Neck', 'Shoulder'),
                               ordered = TRUE))
```

Number of pain sites

```
df_long <- df %>%
    pivot_longer(cols = ends_with('_pain'),
                 names_to = 'site',
                 values_to = 'values')


df_number <- df_long %>%
    group_by(ranid, interval_name) %>%
    nest() %>%
    mutate(logical = map(.x = data,
                         ~ .x %>%
                             mutate(values = ifelse(values == 'Yes',
                                                    yes = TRUE,
                                                    no = FALSE)))) %>%
    mutate(summary = map(.x = logical,
                         ~ .x %>%
                             summarise(total = sum(values, na.rm = TRUE)))) %>%
    select(-data, -logical) %>%
    unnest(summary) %>%
    ungroup()

df_pain <- df %>%
    select(ranid, interval_name, pain_in_the_last_week)

df_number %<>%
    left_join(df_pain)
```

```
## Joining, by = c("ranid", "interval_name")
```

```
# Print summary table
df_number %>%
    group_by(interval_name) %>%
    nest() %>%
    mutate(filtered = map(.x = data,
                          ~ .x %>%
                              filter(total > 0))) %>%
    select(-data) %>%
    unnest(filtered) %>%
    group_by(interval_name) %>%
    summarise(mean = round(mean(total, na.rm = TRUE), 2),
              SD = round(sd(total, na.rm = TRUE), 2),
              median = median(total, na.rm = TRUE),
              Q25 = quantile(total, probs = 0.25, na.rm = TRUE),
              Q75 = quantile(total, probs = 0.75, na.rm = TRUE),
              min = min(total, na.rm = TRUE),
              max = max(total, na.rm = TRUE)) %>%
    knitr::kable(caption = 'Mean number of pain sites in people with pain')
```

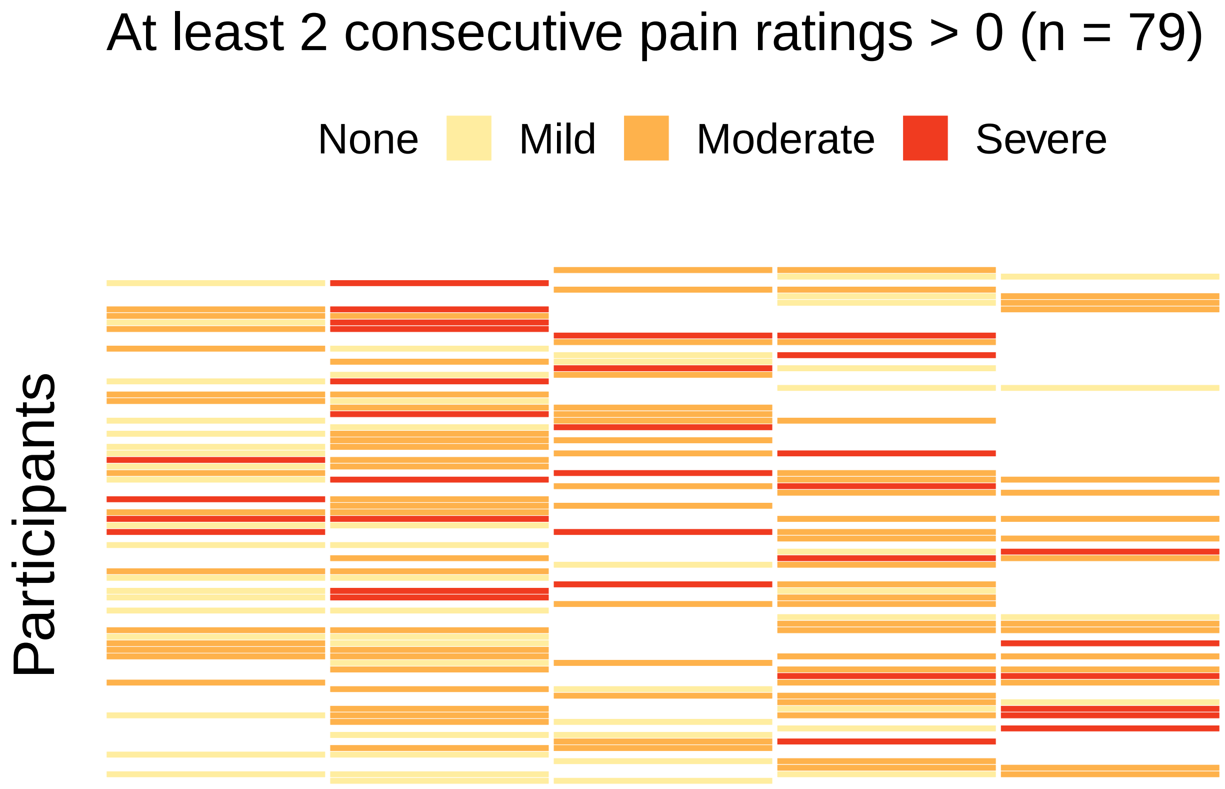
Table 1: Mean number of pain sites in people with pain

interval_name	mean	SD	median	Q25	Q75	min	max
0 weeks	1.38	0.88	1	1	2	1	9
12 weeks	1.25	0.67	1	1	1	1	5
24 weeks	1.35	0.99	1	1	1	1	10
36 weeks	1.24	0.64	1	1	1	1	6
48 weeks	1.30	0.68	1	1	1	1	4

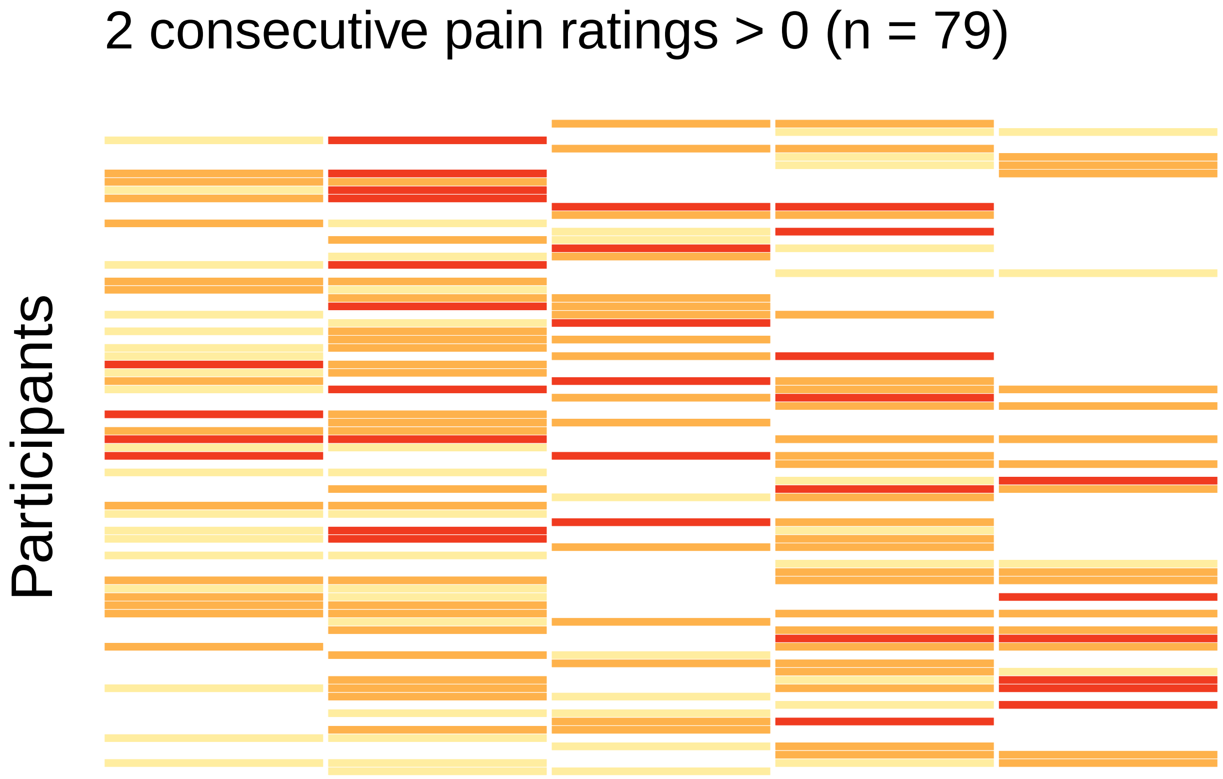
```r
# 2 yes's in a row ONLY
df_ns_2yes <- df_number %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(!ranid %in% vec_filter_3yes) %>%
    filter(ranid %in% vec_filter_2yes) %>%
    mutate(total_text = ifelse(total > 4,
                               yes = '>4',
                               no = as.character(total)),
           total_text = ifelse(total_text == '0',
                               yes = 'None',
                               no = total_text),
           total_text = factor(total_text,
                               levels = c('None', as.character(1:4), '>4'),
                               ordered = TRUE))
# 3 yes's in a row ONLY
df_ns_3yes <- df_number %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(!ranid %in% vec_filter_4yes) %>%
    filter(ranid %in% vec_filter_3yes) %>%
    mutate(total_text = ifelse(total > 4,
                               yes = '>4',
                               no = as.character(total)),
           total_text = ifelse(total_text == '0',
                               yes = 'None',
                               no = total_text),
           total_text = factor(total_text,
                               levels = c('None', as.character(1:4), '>4'),
                               ordered = TRUE))

# At least 4 yes's in a row
df_ns_4yes <- df_number %>%
    mutate(interval_name = fct_drop(interval_name)) %>%
    filter(ranid %in% vec_filter_4yes) %>%
    mutate(total_text = ifelse(total > 4,
                               yes = '>4',
                               no = as.character(total)),
           total_text = ifelse(total_text == '0',
                               yes = 'None',
                               no = total_text),
           total_text = factor(total_text,
                               levels = c('None', as.character(1:4), '>4'),
                               ordered = TRUE))
```

Pain intensity heatmaps

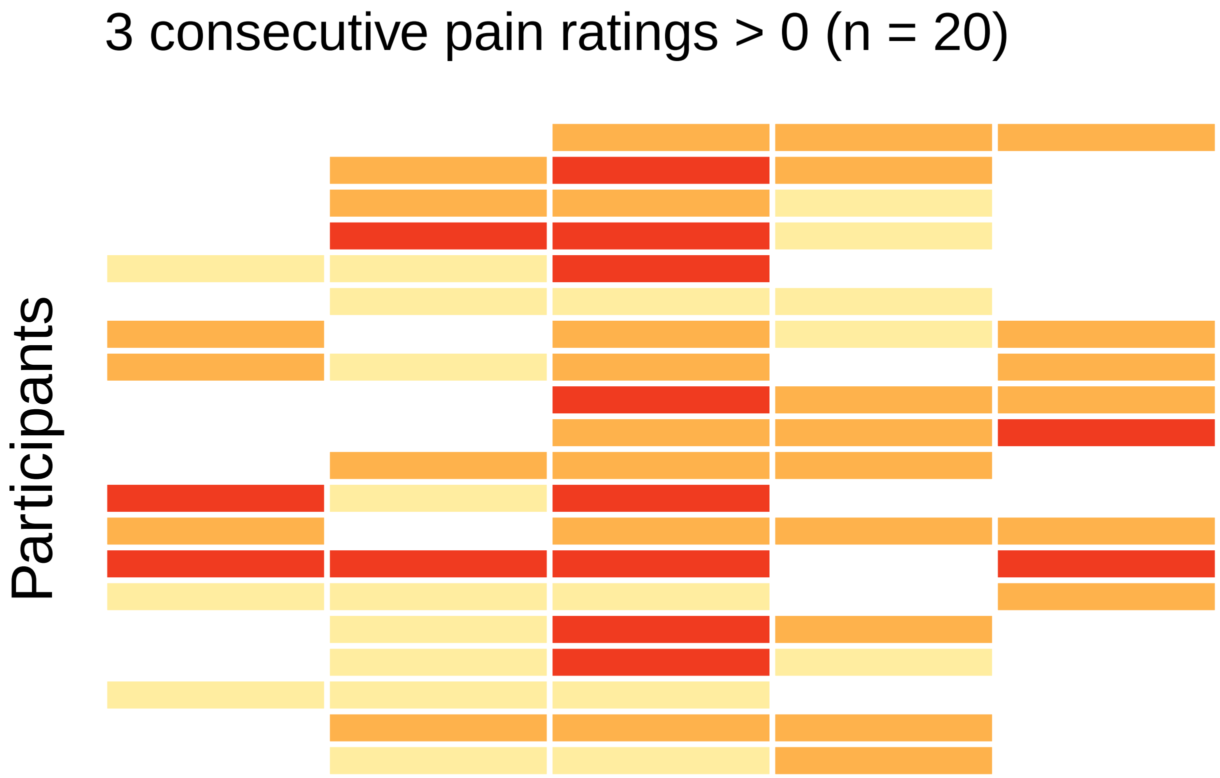
```r
# At least 2 yes's in a row
p_intensity2_legend <- df_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = intensity), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain intensity: ',
                      values = c('#FFFFFF', '#ffeda0','#feb24c','#f03b20'),
                      guide = guide_legend(ncol = 4)) +
    labs(subtitle = str_glue('At least 2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})')
         x = 'Time (weeks)',
         y = 'Participants') +
    scale_x_discrete(expand = c(0, 0)) +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'top',
          legend.title = element_blank(),
          axis.text.y = element_blank(),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          panel.grid = element_blank()); p_intensity2_legend
```



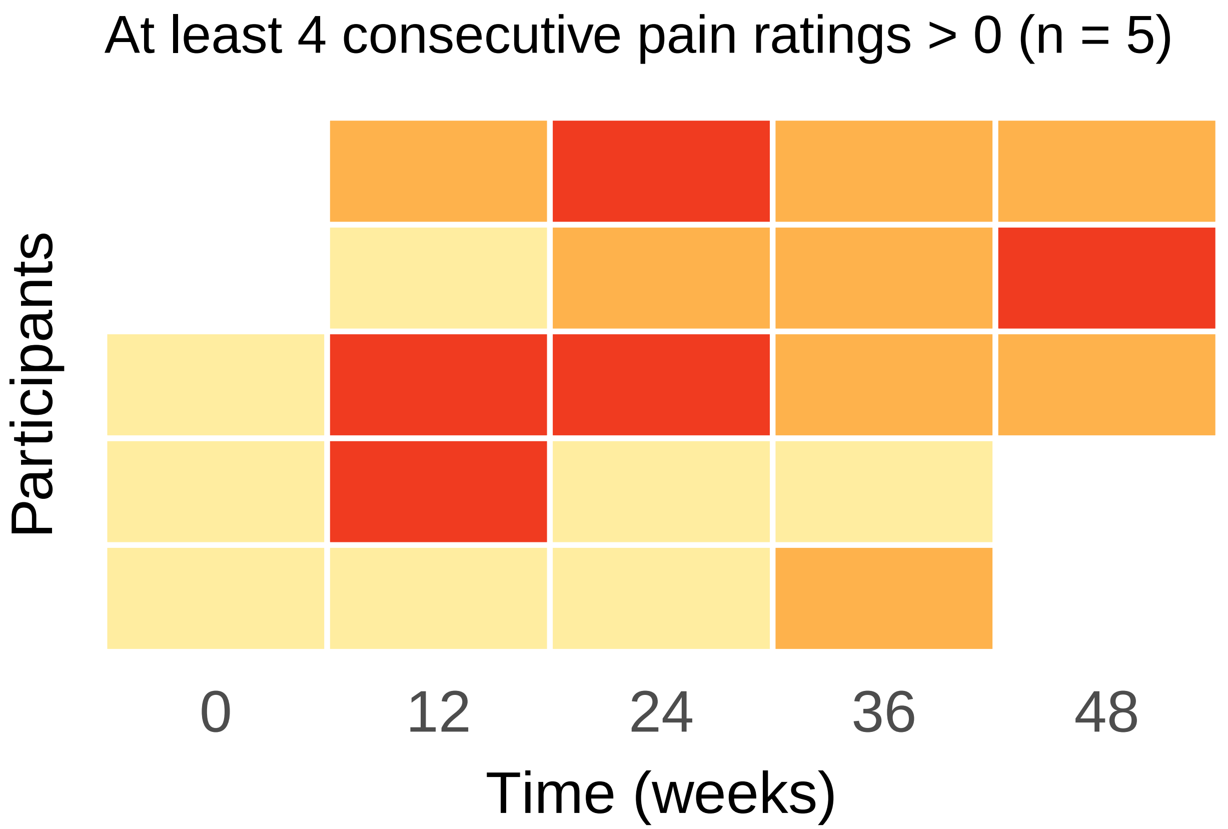
```r
p_intensity2 <- df_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = intensity), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain intensity: ',
                       values = c('#FFFFFF', '#ffeda0','#feb24c','#f03b20'),
                       guide = guide_legend(ncol = 4)) +
    labs(subtitle = str_glue('2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    scale_x_discrete(expand = c(0, 0)) +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.y = element_blank(),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          panel.grid = element_blank()); p_intensity2
```



```
# At least 3 yes's in a row
p_intensity3 <- df_3yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = intensity), colour = '#FFFFFF', size = 1, width = 1) +
    scale_fill_manual(name = 'Pain intensity: ',
                       values = c('#FFFFFF', '#ffeda0','#feb24c','#f03b20'),
                       guide = guide_legend(ncol = 4)) +
    labs(subtitle = str_glue('3 consecutive pain ratings > 0 (n = {length(unique(df_3yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    scale_y_continuous(breaks = 1:length(unique(df_3yes$ranid)),
                       labels = 1:length(unique(df_3yes$ranid))) +
    scale_x_discrete(expand = c(0, 0)) +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          panel.grid = element_blank()); p_intensity3
```



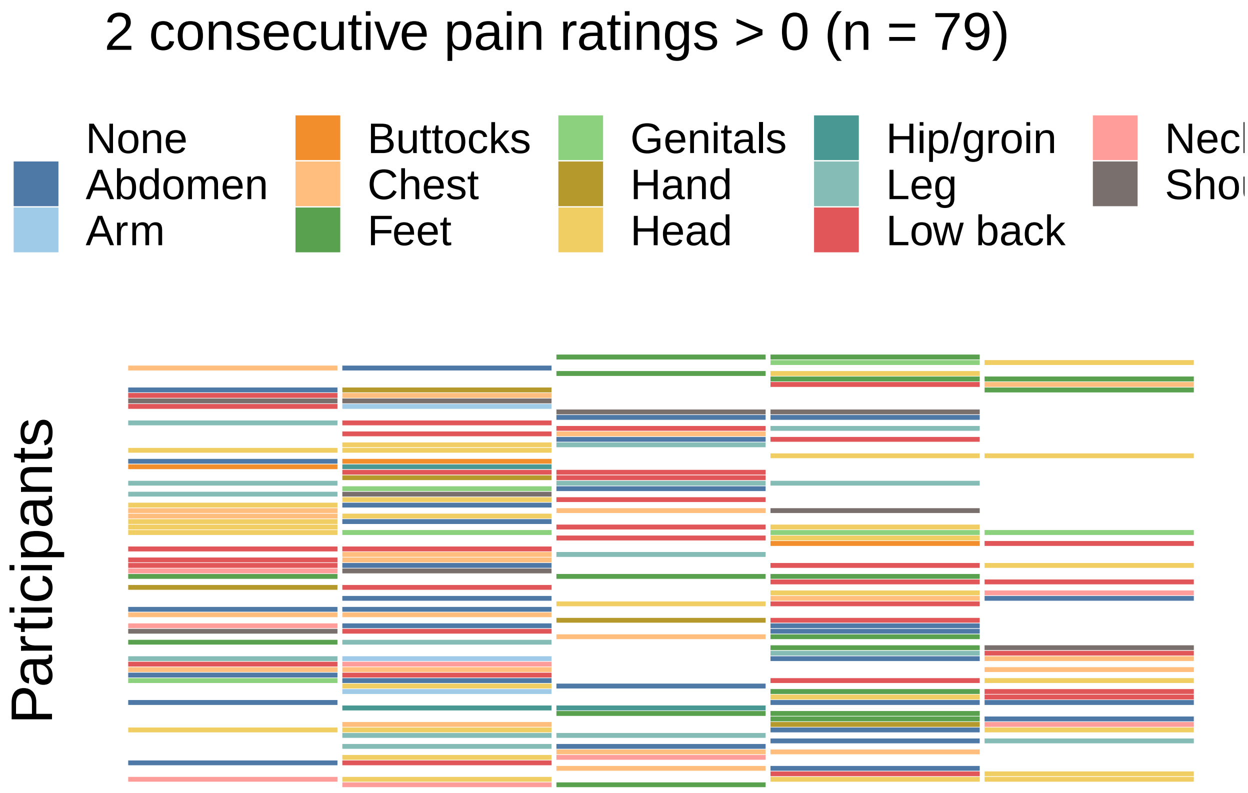
```
# At least 4 yes's in a row
p_intensity4 <- df_4yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = intensity), colour = '#FFFFFF', size = 1, width = 1) +
    scale_fill_manual(name = 'Pain intensity: ',
                      values = c('#FFFFFF', '#ffeda0','#feb24c','#f03b20'),
                      guide = guide_legend(ncol = 4)) +
    labs(subtitle = str_glue('At least 4 consecutive pain ratings > 0 (n = {length(unique(df_4yes$ranid))})')
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    scale_y_continuous(breaks = 1:length(unique(df_4yes$ranid)),
                      labels = 1:length(unique(df_4yes$ranid))) +
    scale_x_discrete(expand = c(0, 0)) +
    theme(legend.position = 'none',
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.title.x = element_text(size = 22),
          axis.text.x = element_text(size = 22, vjust = 2),
          axis.text.y = element_blank(),
          panel.grid = element_blank()); p_intensity4
```



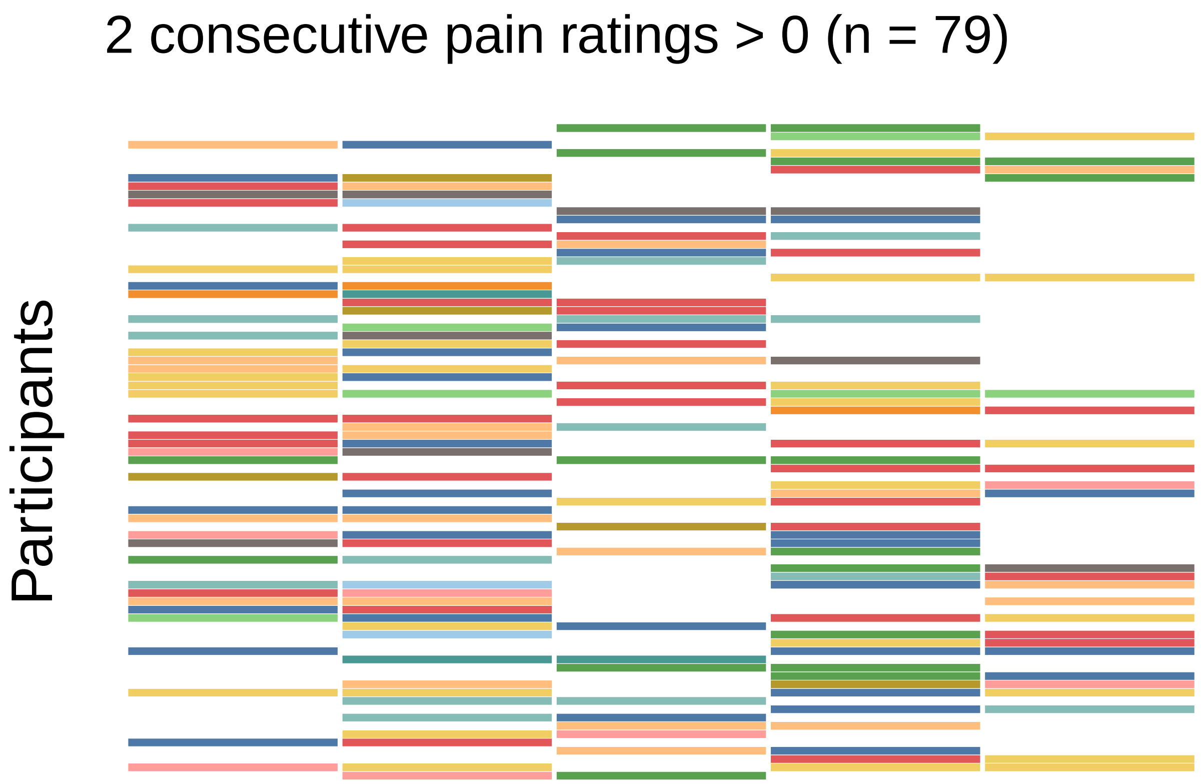
```r
# Patchwork
pw_intensity <- p_intensity2 + p_intensity3 + p_intensity4 +
    plot_layout(ncol = 1, heights = c(2, 2, 0.5)) +
    plot_annotation(title = 'Intensity of pain at its worst',
                    tag_levels = 'A')

# Save
ggsave(filename = 'figures/figure-4a.png', pw_intensity, height = 20, width = 10)
```

Pain sites heatmaps

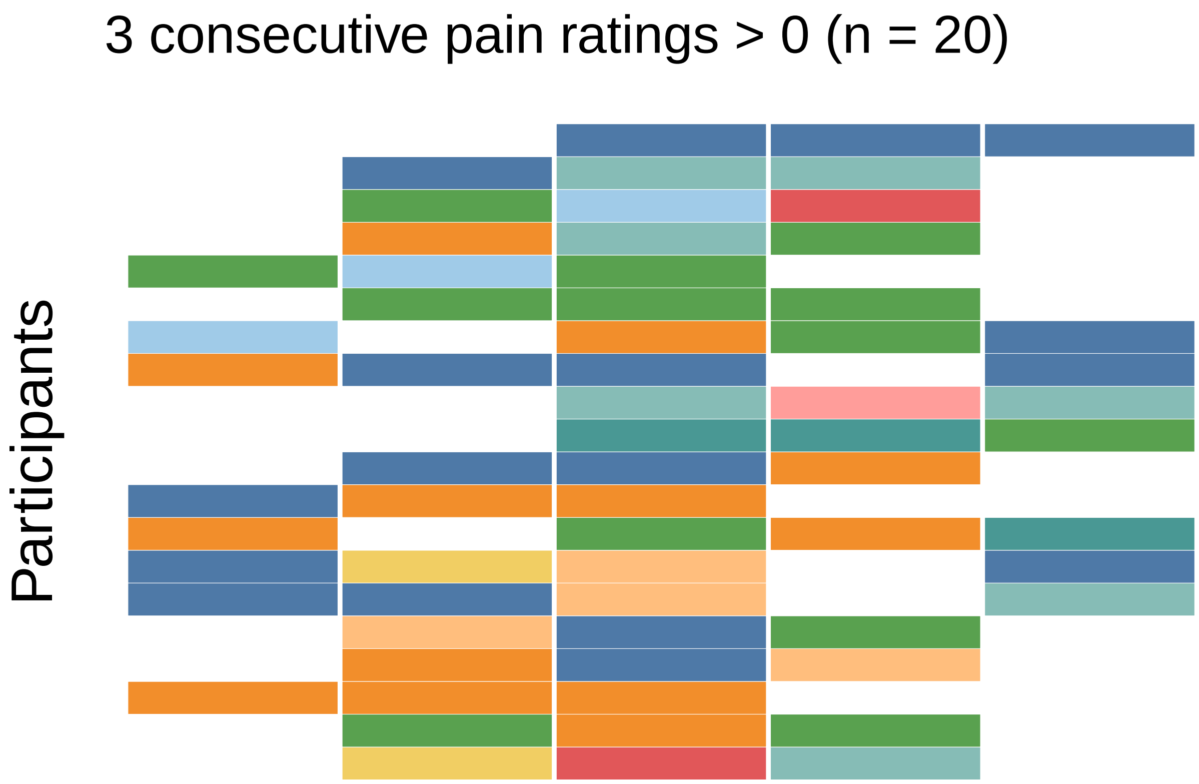
```r
# At least 2 yes's in a row
p_sites2_legend <- df_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = site_worst), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain site: ',
                      values = c('#FFFFFF', '#4E79A7', '#A0CBE8', '#F28E2B',
                                 '#FFBE7D', '#59A14F', '#8CD17D', '#B6992D',
                                 '#F1CE63', '#499894', '#86BCB6', '#E15759',
                                 '#FF9D9A', '#79706E'),
                       guide = guide_legend(ncol = 5)) +
    labs(subtitle = str_glue('2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'top',
          legend.title = element_blank(),
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_sites2_legend
```



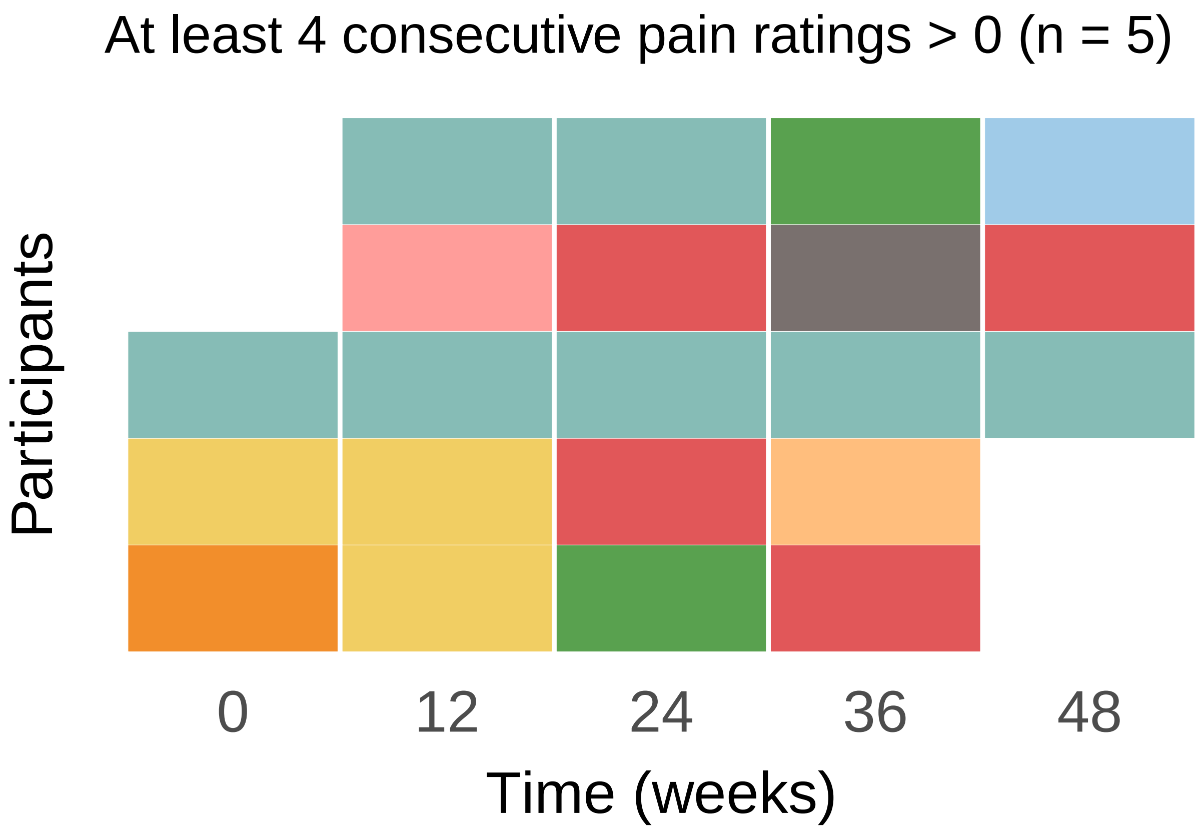
```
p_sites2 <- df_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = site_worst), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain site: ',
                      values = c('#FFFFFF', '#4E79A7', '#A0CBE8', '#F28E2B',
                                 '#FFBE7D', '#59A14F', '#8CD17D', '#B6992D',
                                 '#F1CE63', '#499894', '#86BCB6', '#E15759',
                                 '#FF9D9A', '#79706E'),
                       guide = guide_legend(ncol = 5)) +
    labs(subtitle = str_glue('2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_sites2
```



```
# At least 3 yes's in a row
p_sites3 <- df_3yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = site_worst), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain site: ',
                       values = c('#FFFFFF', '#4E79A7', '#A0CBE8', '#F28E2B',
                                  '#FFBE7D', '#59A14F', '#F1CE63', '#499894',
                                  '#86BCB6', '#E15759', '#FF9D9A', '#79706E'),
                       guide = guide_legend(ncol = 4)) +
    labs(subtitle = str_glue('3 consecutive pain ratings > 0 (n = {length(unique(df_3yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    scale_y_continuous(breaks = 1:length(unique(df_3yes$ranid)),
                       labels = 1:length(unique(df_3yes$ranid))) +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_sites3
```



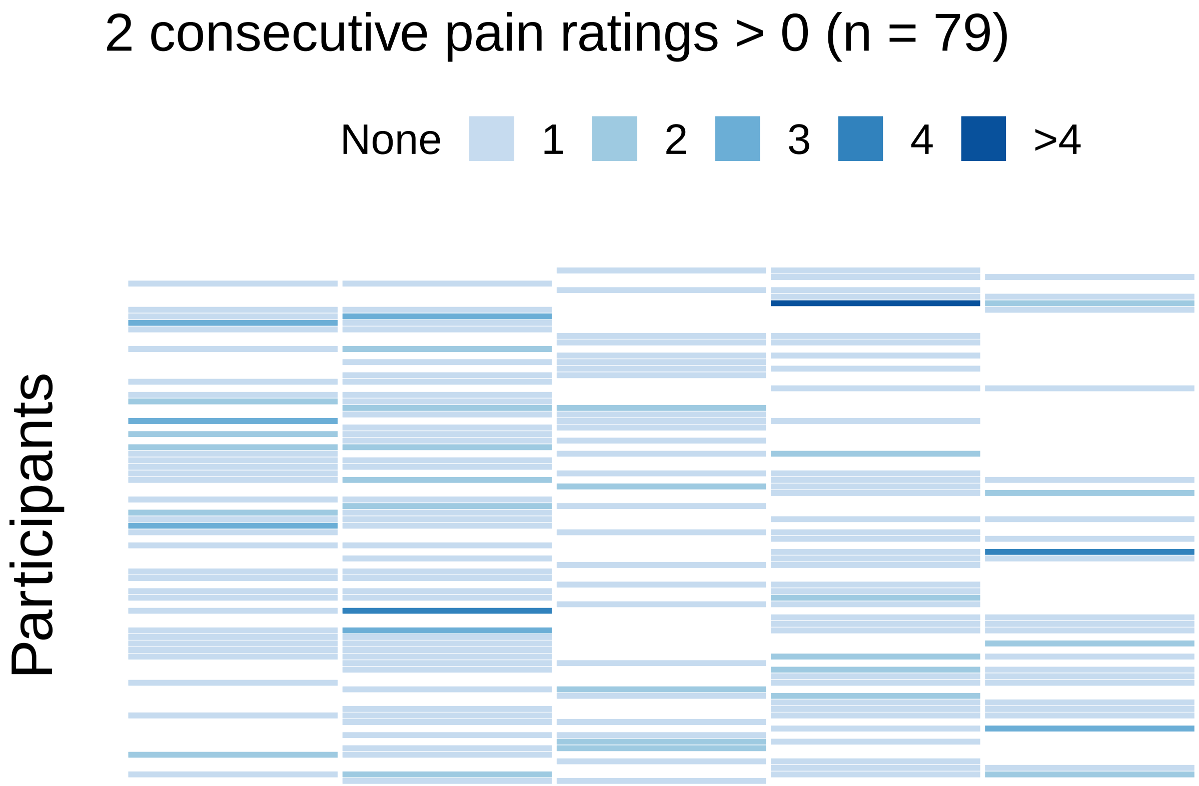
```
# At least 4 yes's in a row
p_sites4 <- df_4yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_numeric),
        y = id) +
    geom_tile(aes(fill = site_worst), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(name = 'Pain site: ',
                      values = c('#FFFFFF', '#A0CBE8', '#F28E2B', '#FFBE7D',
                                 '#59A14F', '#F1CE63', '#86BCB6', '#E15759',
                                 '#FF9D9A', '#79706E'),
                       guide = guide_legend(ncol = 4)) +
    scale_y_continuous(breaks = 1:length(unique(df_4yes$ranid)),
                       labels = 1:length(unique(df_4yes$ranid))) +
    labs(subtitle = str_glue('At least 4 consecutive pain ratings > 0 (n = {length(unique(df_4yes$ranid))})')
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.title.x = element_text(size = 22),
          axis.text.x = element_text(size = 22, vjust = 2),
          axis.text.y = element_blank(),
          panel.grid = element_blank()); p_sites4
```



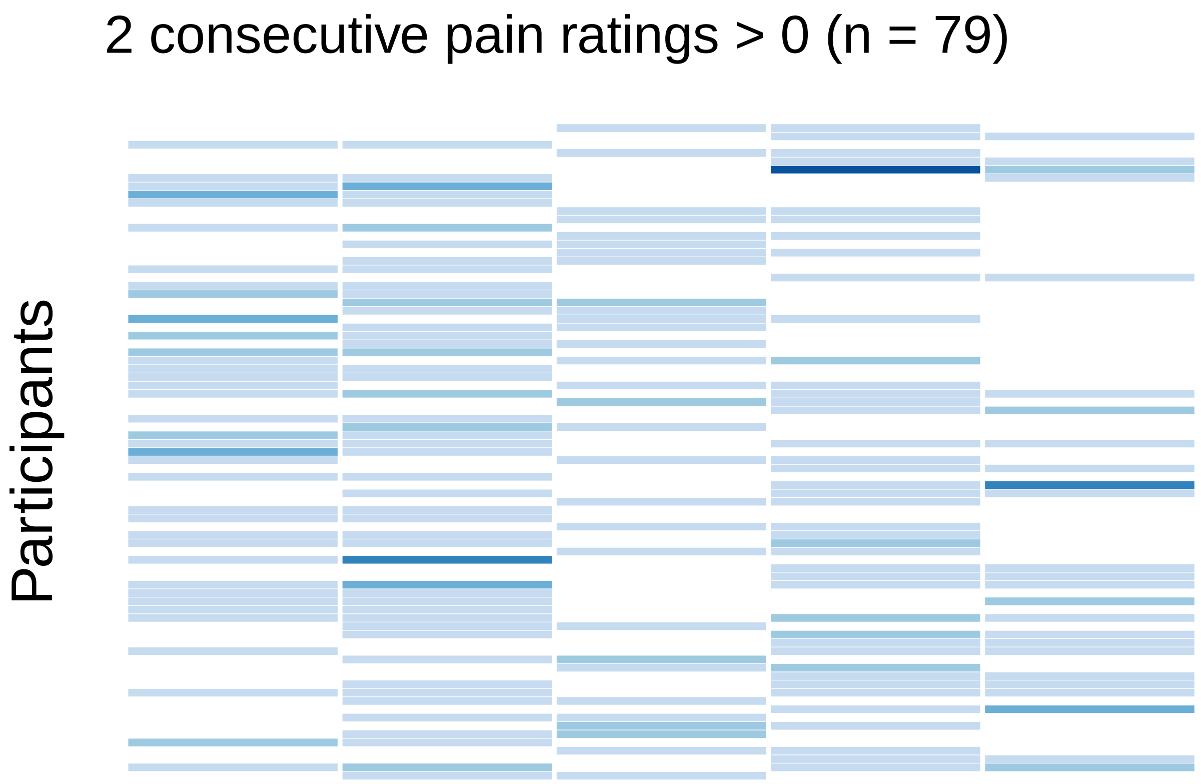
```r
# Patchwork
pw_sites <- p_sites2 + p_sites3 + p_sites4 +
    plot_layout(ncol = 1, heights = c(2, 2, 0.5)) +
    plot_annotation(title = 'Site of worst pain',
                    tag_levels = 'A')

# Save
ggsave(filename = 'figures/figure-4c.png', pw_sites, height = 20, width = 10)
```

Number of pain sites heatmaps

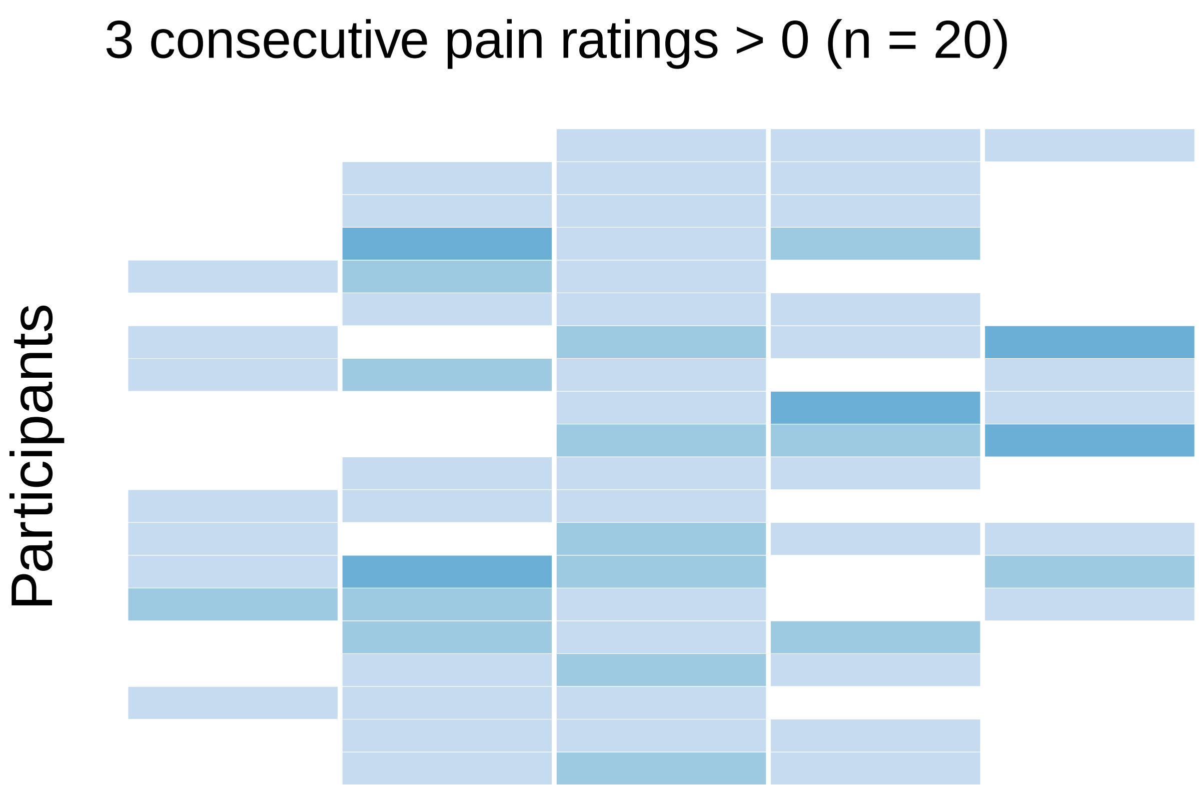
```r
# At least 2 yes's in a row
p_number2_legend <- df_ns_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_name),
        y = id) +
    geom_tile(aes(fill = total_text), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(values = c('#FFFFFF', '#c6dbef','#9ecae1','#6baed6','#3182bd','#08519c'),
                      name = 'Number of pains: ',
                      guide = guide_legend(ncol = 6)) +
    labs(subtitle = str_glue('2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'top',
          legend.title = element_blank(),
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_number2_legend
```



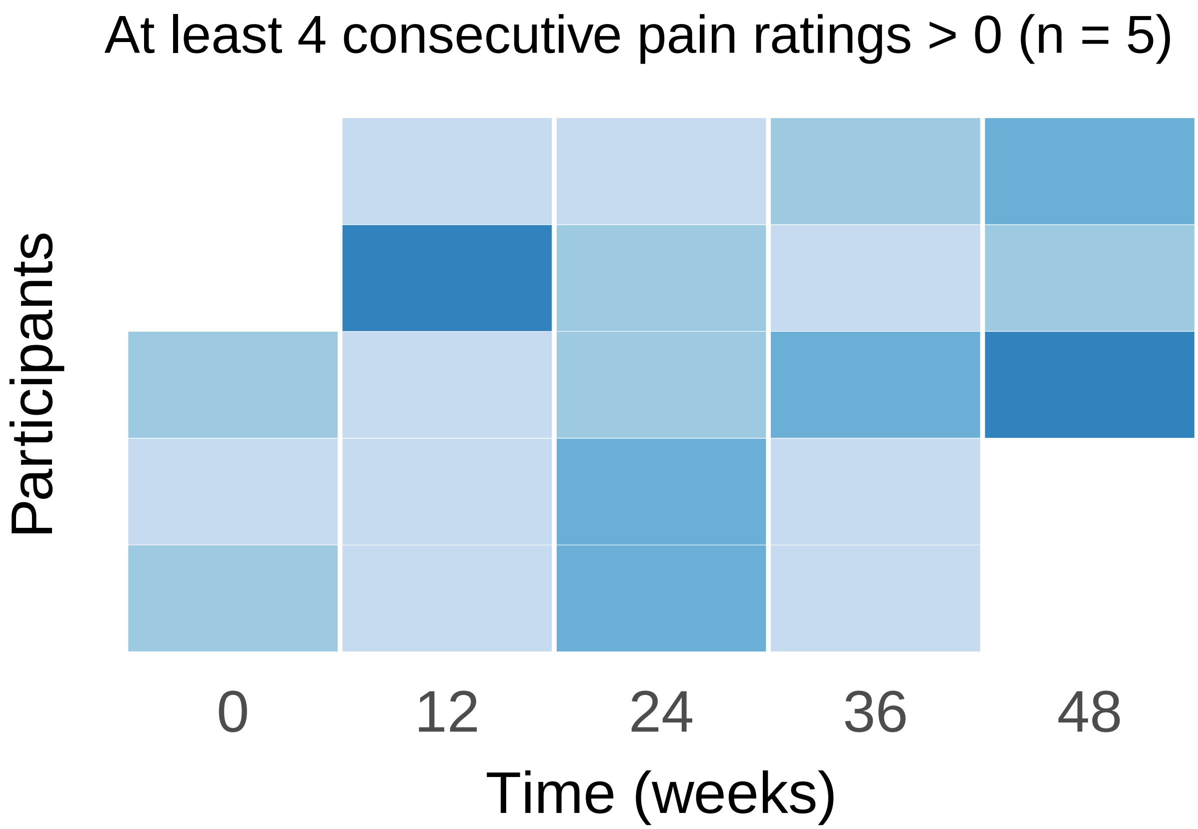
```
p_number2 <- df_ns_2yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_name),
        y = id) +
    geom_tile(aes(fill = total_text), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(values = c('#FFFFFF', '#c6dbef','#9ecae1','#6baed6','#3182bd','#08519c'),
                      name = 'Number of pains: ',
                      guide = guide_legend(ncol = 3)) +
    labs(subtitle = str_glue('2 consecutive pain ratings > 0 (n = {length(unique(df_2yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_number2
```



```
# At least 3 yes's in a row
p_number3 <- df_ns_3yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_name),
        y = id) +
    geom_tile(aes(fill = total_text), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(values = c('#FFFFFF', '#c6dbef','#9ecae1','#6baed6','#3182bd','#08519c'),
                      name = 'Number of pains: ',
                      guide = guide_legend(ncol = 3)) +
    labs(subtitle = str_glue('3 consecutive pain ratings > 0 (n = {length(unique(df_3yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    scale_y_continuous(breaks = 1:length(unique(df_3yes$ranid)),
                       labels = 1:length(unique(df_3yes$ranid))) +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.text.y = element_blank(),
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.text.x = element_blank(),
          axis.title.x = element_blank(),
          panel.grid = element_blank()); p_number3
```



```
# At least 4 yes's in a row
p_number4 <- df_ns_4yes %>%
    group_by(interval_name) %>%
    mutate(id = row_number()) %>%
    ggplot(.) +
    aes(x = factor(interval_name),
        y = id) +
    geom_tile(aes(fill = total_text), colour = '#FFFFFF', width = 0.98) +
    scale_fill_manual(values = c('#FFFFFF', '#c6dbef','#9ecae1','#6baed6','#3182bd','#08519c'),
                      name = 'Number of pains: ',
                      guide = guide_legend(ncol = 3)) +
    scale_x_discrete(labels = as.character(c(0, 12, 24, 36, 48))) +
    scale_y_continuous(breaks = 1:length(unique(df_4yes$ranid)),
                       labels = c(2, 5, 7, 8, 19)) +
    labs(subtitle = str_glue('At least 4 consecutive pain ratings > 0 (n = {length(unique(df_4yes$ranid))})'),
         x = 'Time (weeks)',
         y = 'Participants') +
    theme_minimal(base_size = 20) +
    theme(legend.position = 'none',
          axis.title.y = element_text(size = 22, vjust = 2),
          axis.title.x = element_text(size = 22),
          axis.text.x = element_text(size = 22, vjust = 2),
          axis.text.y = element_blank(),
          panel.grid = element_blank()); p_number4
```



```
# Patchwork
pw_number <- p_number2 + p_number3 + p_number4 +
    plot_layout(ncol = 1, heights = c(2, 2, 0.5)) +
    plot_annotation(title = 'Number of pain sites',
                    tag_levels = 'A')

# Save
ggsave(filename = 'figures/figure-4b.png', pw_number, height = 20, width = 10)
```

Session information

```
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 magrittr_1.5    forcats_0.4.0   stringr_1.4.0
##  [5] dplyr_0.8.3     purrr_0.3.3     readr_1.3.1     tidyr_1.0.0
##  [9] 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_0.4.2      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_1.0.3
## [29] scales_1.0.0     backports_1.1.5  jsonlite_1.6     hms_0.5.1
## [33] digest_0.6.23    stringi_1.4.3    grid_3.6.1       cli_2.0.0
## [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_2.4.1         nlme_3.1-141     compiler_3.6.1
```