Supplement 2

Statistics at each time period

Peter Kamerman

01 April 2020

Contents

Objective

To describe the pain characteristics (analysis cohort and stratified by data missingness), demographic characteristics (analysis cohort only), and disease status (analysis cohort only) at each time point.

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

```r
df <- read_rds('data-cleaned/data-ADVANCE.rds') %>%
    select(ranid, interval_name, group, pain_in_the_last_week, pain_worst,
           cd4_cells.ul, viral_load_cp.ml,
           general_health, any_missing, interval_numeric)
```

First look

```r
head(df)
```

```
## # A tibble: 6 x 10
##   ranid interval_name group pain_in_the_las~ pain_worst cd4_cells.ul
##   <chr> <ord>         <chr> <chr>                 <dbl>        <dbl>
## 1 01-0~ 0 weeks       DTG ~ No                        0          642
## 2 01-0~ 12 weeks      DTG ~ No                        0           NA
## 3 01-0~ 24 weeks      DTG ~ No                        0          525
## 4 01-0~ 36 weeks      DTG ~ No                        0           NA
## 5 01-0~ 48 weeks      DTG ~ No                        0          668
## 6 01-0~ 0 weeks       DTG ~ No                        0          241
## # ... with 4 more variables: viral_load_cp.ml <dbl>, general_health <dbl>,
## #   any_missing <chr>, interval_numeric <dbl>
```

```r
glimpse(df)
```

```
## Rows: 5,265
## Columns: 10
## $ 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, ...
## $ cd4_cells.ul           <dbl> 642, NA, 525, NA, 668, 241, NA, 364, NA, 495,...
## $ viral_load_cp.ml       <dbl> 641, 50, 50, 50, 50, 3851, 50, 50, 50, 50, 90...
## $ general_health         <dbl> 4, 4, 5, 5, 4, 3, 5, 3, 3, 3, 4, 5, 5, 5, 5, ...
## $ 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

```r
# Clean and process data
df_pain <- df %>%
    select(ranid, any_missing) %>%
    distinct()

pain_nested <- df %>%
    select(ranid, interval_name, pain_worst, any_missing) %>%
    group_by(interval_name) %>%
    nest()

df %<>%
    filter(any_missing == 'No') %>%
    select(-any_missing)
```

Quick tabulation

Analysis data set for the period 0 to 48 weeks

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

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

Analysis

Number of participants with/without complete pain data

```r
df_pain %>%
    group_by(any_missing) %>%
    summarise(count = n()) %>%
    mutate(n = sum(count),
           proportion = round(count / n, 3)) %>%
    kable(caption = 'Number of participants with/without missing pain data')
```

Table 1: Number of participants with/without missing pain data

any_missing	count	n	proportion
No	787	1053	0.747

any_missing	count	n	proportion
Yes	266	1053	0.253

Pain frequency at each week, by data missingness

Number of participants with pain at each interval

```
pain_nested <- pain_nested  %>%
    mutate(numbers = map2(.x = data,
                          .y = interval_name,
                         ~ .x %>%
                             mutate(pain_status = ifelse(pain_worst > 0,
                                                      yes = 'pain',
                                                      no = 'no pain')) %>%
                             group_by(any_missing, pain_status) %>%
                             summarise(count = n()) %>%
                             mutate(pain_status = ifelse(is.na(pain_status),
                                                         yes = 'Missing pain intensity data',
                                                         no = pain_status)) %>%
                             group_by(any_missing) %>%
                             mutate(total = sum(count),
                                    proportion = round(count / total, 2)) %>%
                             kable(caption = str_glue('{str_to_sentence(.y)}: Number of participants with/wit

walk(.x = pain_nested$numbers, ~print(.x))
```

Table 2: 0 weeks: Number of participants with/without missing pain data and who had pain > 0

any_missing	pain_status	count	total	proportion
No	no pain	636	787	0.81
No	pain	151	787	0.19
Yes	no pain	191	266	0.72
Yes	pain	69	266	0.26
Yes	Missing pain intensity data	6	266	0.02

Table 3: 12 weeks: Number of participants with/without missing pain data and who had pain > 0

any_missing	pain_status	count	total	proportion
No	no pain	644	787	0.82
No	pain	143	787	0.18
Yes	no pain	157	266	0.59
Yes	pain	44	266	0.17
Yes	Missing pain intensity data	65	266	0.24

Table 4: 24 weeks: Number of participants with/without missing pain data and who had pain > 0

any_missing	pain_status	count	total	proportion
No	no pain	660	787	0.84
No	pain	127	787	0.16
Yes	no pain	111	266	0.42

any_missing	pain_status	count	total	proportion
Yes	pain	46	266	0.17
Yes	Missing pain intensity data	109	266	0.41

Table 5: 36 weeks: Number of participants with/without missing pain data and who had pain > 0

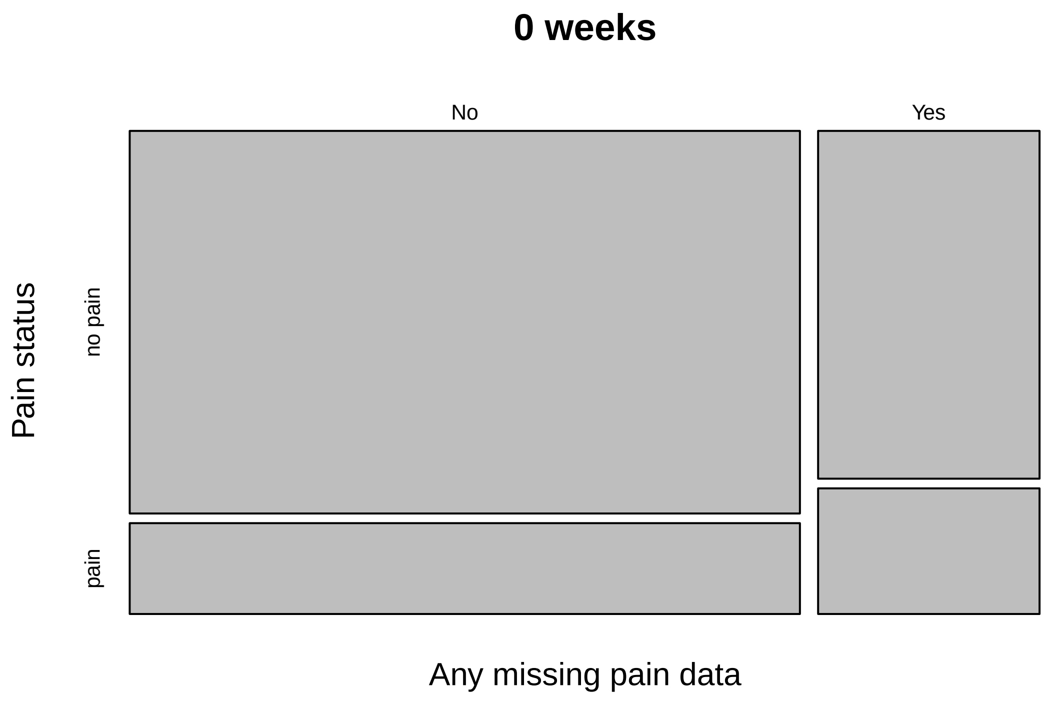
any_missing	pain_status	count	total	proportion
No	no pain	673	787	0.86
No	pain	114	787	0.14
Yes	no pain	68	266	0.26
Yes	pain	22	266	0.08
Yes	Missing pain intensity data	176	266	0.66

Table 6: 48 weeks: Number of participants with/without missing pain data and who had pain > 0

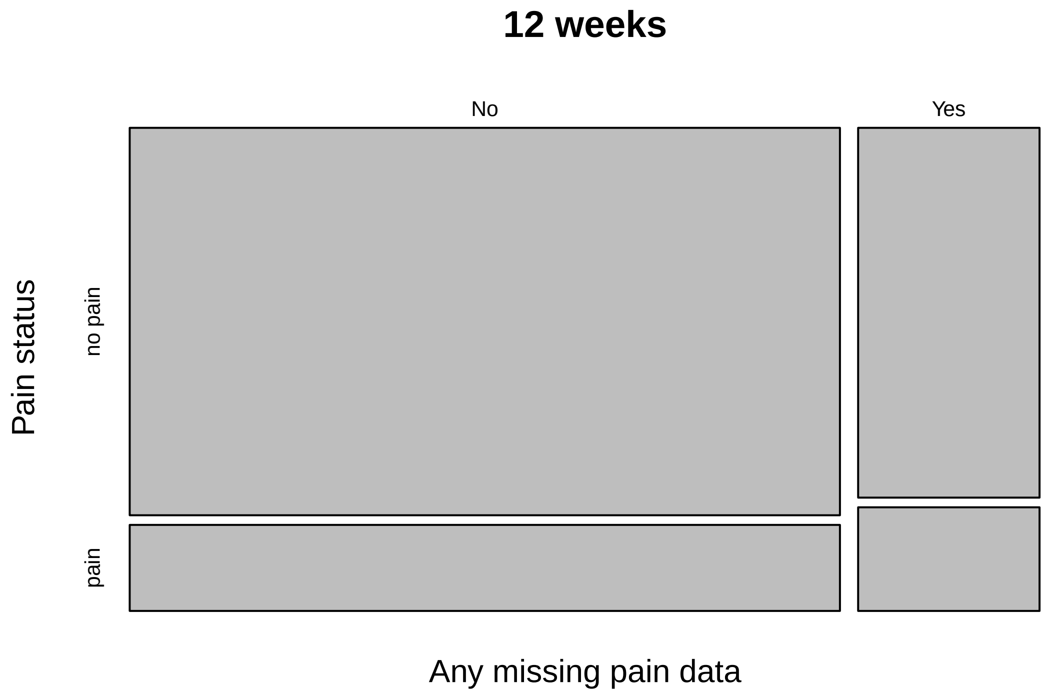
any_missing	pain_status	count	total	proportion
No	no pain	691	787	0.88
No	pain	96	787	0.12
Yes	no pain	95	266	0.36
Yes	pain	20	266	0.08
Yes	Missing pain intensity data	151	266	0.57

Plot of pain frequency at each interval

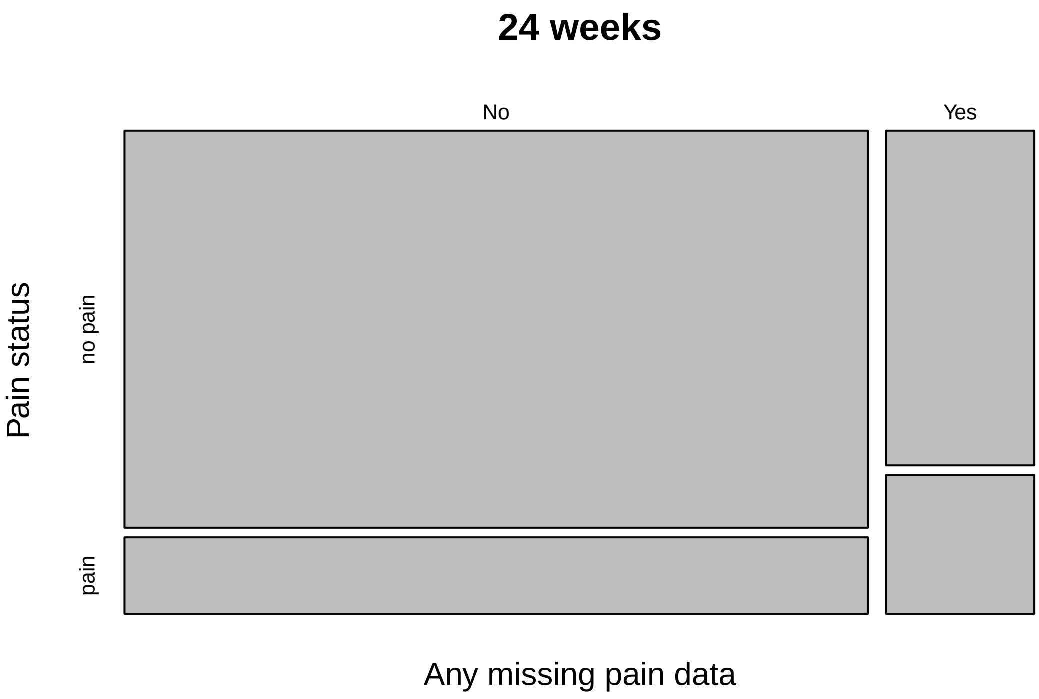
```
pain_nested <- pain_nested %>%
    mutate(xtab = map2(.x = data,
                       .y = interval_name,
                      ~ .x %>%
                         filter(!is.na(pain_worst)) %>%
                         mutate(pain_status = ifelse(pain_worst > 0,
                                                     yes = 'pain',
                                                     no = 'no pain')) %>%
                         xtabs(~ any_missing + pain_status, data = .)))

mosaicplot(pain_nested$xtab[[1]],
            main = '0 weeks',
            xlab = 'Any missing pain data',
            ylab = 'Pain status')
```



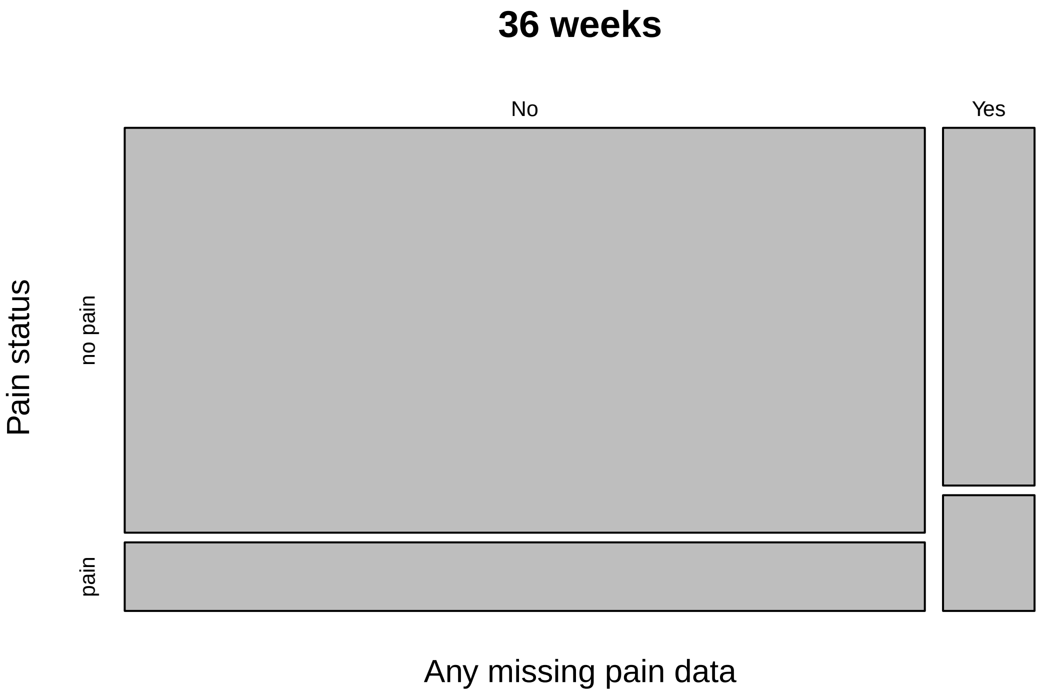
```r
mosaicplot(pain_nested$xtab[[2]],
           main = '12 weeks',
           xlab = 'Any missing pain data',
           ylab = 'Pain status')
```



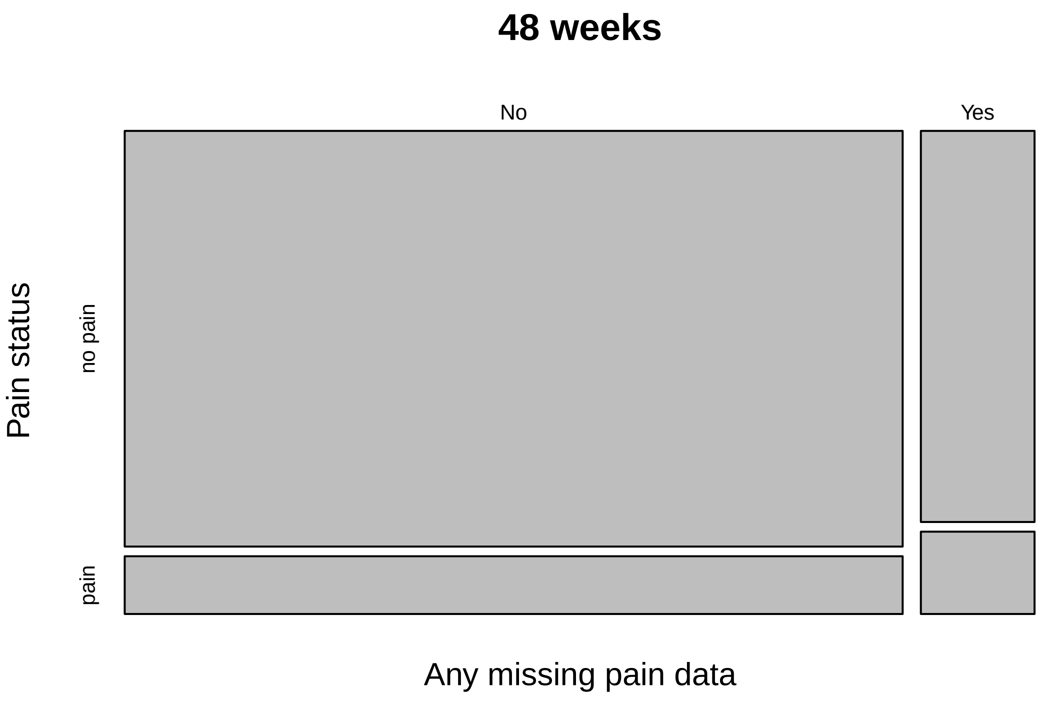
```r
mosaicplot(pain_nested$xtab[[3]],
           main = '24 weeks',
           xlab = 'Any missing pain data',
           ylab = 'Pain status')
```



```r
mosaicplot(pain_nested$xtab[[4]],
           main = '36 weeks',
           xlab = 'Any missing pain data',
           ylab = 'Pain status')
```



```r
mosaicplot(pain_nested$xtab[[5]],
           main = '48 weeks',
           xlab = 'Any missing pain data',
           ylab = 'Pain status')
```



Chi-squared test of pain frequency at each interval

```
pain_nested <- pain_nested %>%
    mutate(chisq = map2(.x = data,
                        .y = interval_name,
                       ~ .x %>%
                           filter(!is.na(pain_worst)) %>%
                           mutate(pain_status = ifelse(pain_worst > 0,
                                                       yes = 'pain',
                                                       no = 'no pain')) %>%
                           xtabs(~ any_missing + pain_status, data = .) %>%
                           chisq.test(.) %>%
                           tidy(.) %>%
                           kable(caption = str_glue('{str_to_sentence(.y)}: missing data by pain frequency'))

walk(.x = pain_nested$chisq, ~print(.x))
```

Table 7: 0 weeks: missing data by pain frequency

statistic	p.value	parameter	method
5.928895	0.0148946	1	Pearson's Chi-squared test with Yates' continuity correction

Table 8: 12 weeks: missing data by pain frequency

statistic	p.value	parameter	method
1.211855	0.270965	1	Pearson's Chi-squared test with Yates' continuity correction

Table 9: 24 weeks: missing data by pain frequency

statistic	p.value	parameter	method
14.2829	0.0001573	1	Pearson's Chi-squared test with Yates' continuity correction

Table 10: 36 weeks: missing data by pain frequency

statistic	p.value	parameter	method
5.377131	0.0204024	1	Pearson's Chi-squared test with Yates' continuity correction

Table 11: 48 weeks: missing data by pain frequency

statistic	p.value	parameter	method
1.973453	0.1600818	1	Pearson's Chi-squared test with Yates' continuity correction

Pain intensity at each week, by data missingness

Pain intensity at each interval

```
pain_nested <- pain_nested %>%
    mutate(intensity = map2(.x = data,
                            .y = interval_name,
                            ~ .x %>%
                                filter(pain_worst > 0) %>%
                                group_by(any_missing) %>%
                                summarise(median = median(pain_worst),
                                          Q25 = quantile(pain_worst, probs = 0.25),
                                          Q75 = quantile(pain_worst, probs = 0.75)) %>%
                                kable(caption = str_glue('{str_to_sentence(.y)}: Mean IQR pain intensity of p

walk(.x = pain_nested$intensity, ~print(.x))
```

Table 12: 0 weeks: Mean IQR pain intensity of participants with/without missing pain data and who had pain > 0

any_missing	median	Q25	Q75
No	4	3	6
Yes	4	3	6

Table 13: 12 weeks: Mean IQR pain intensity of participants with/without missing pain data and who had pain > 0

any_missing	median	Q25	Q75
No	5	3	7
Yes	5	3	8

Table 14: 24 weeks: Mean IQR pain intensity of participants with/without missing pain data and who had pain > 0

any_missing	median	Q25	Q75
No	5	3	7.00
Yes	5	4	6.75

Table 15: 36 weeks: Mean IQR pain intensity of participants with/without missing pain data and who had pain > 0

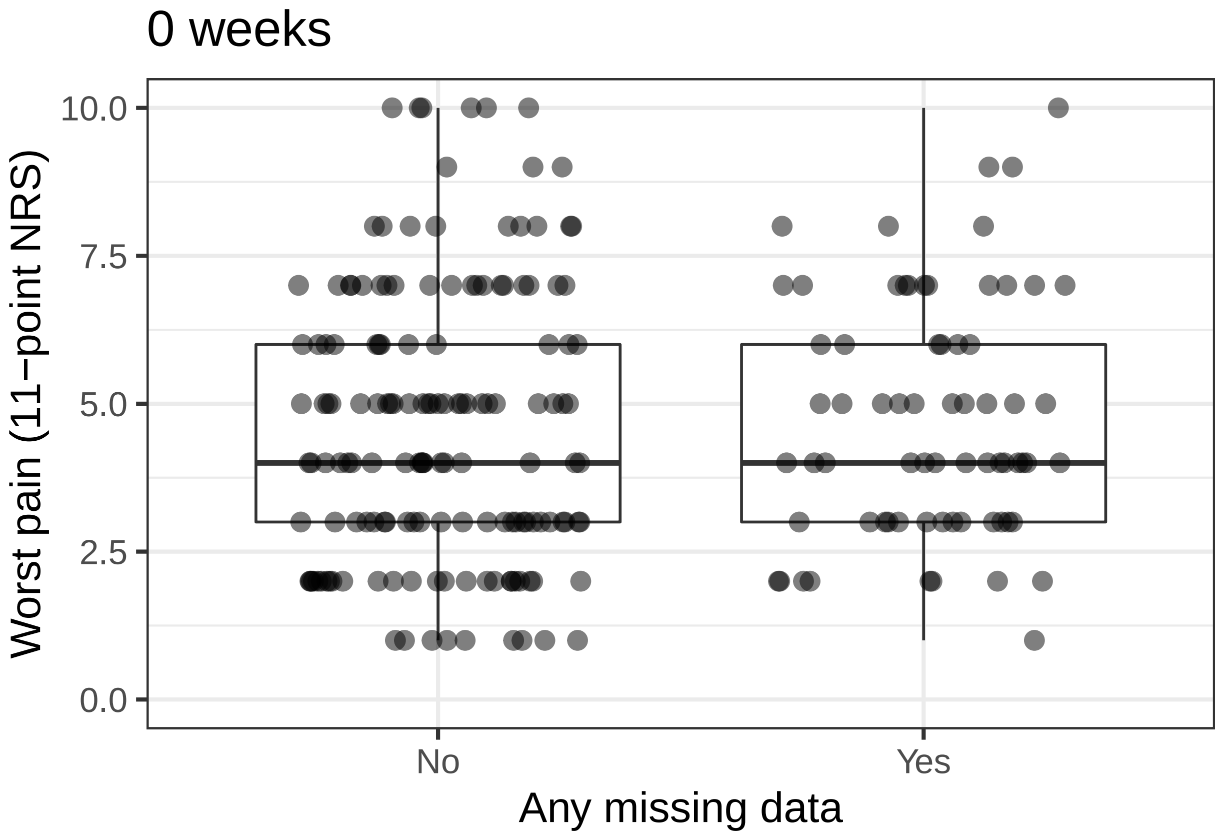
any_missing	median	Q25	Q75
No	4	3	6
Yes	4	4	6

Table 16: 48 weeks: Mean IQR pain intensity of participants with/without missing pain data and who had pain > 0

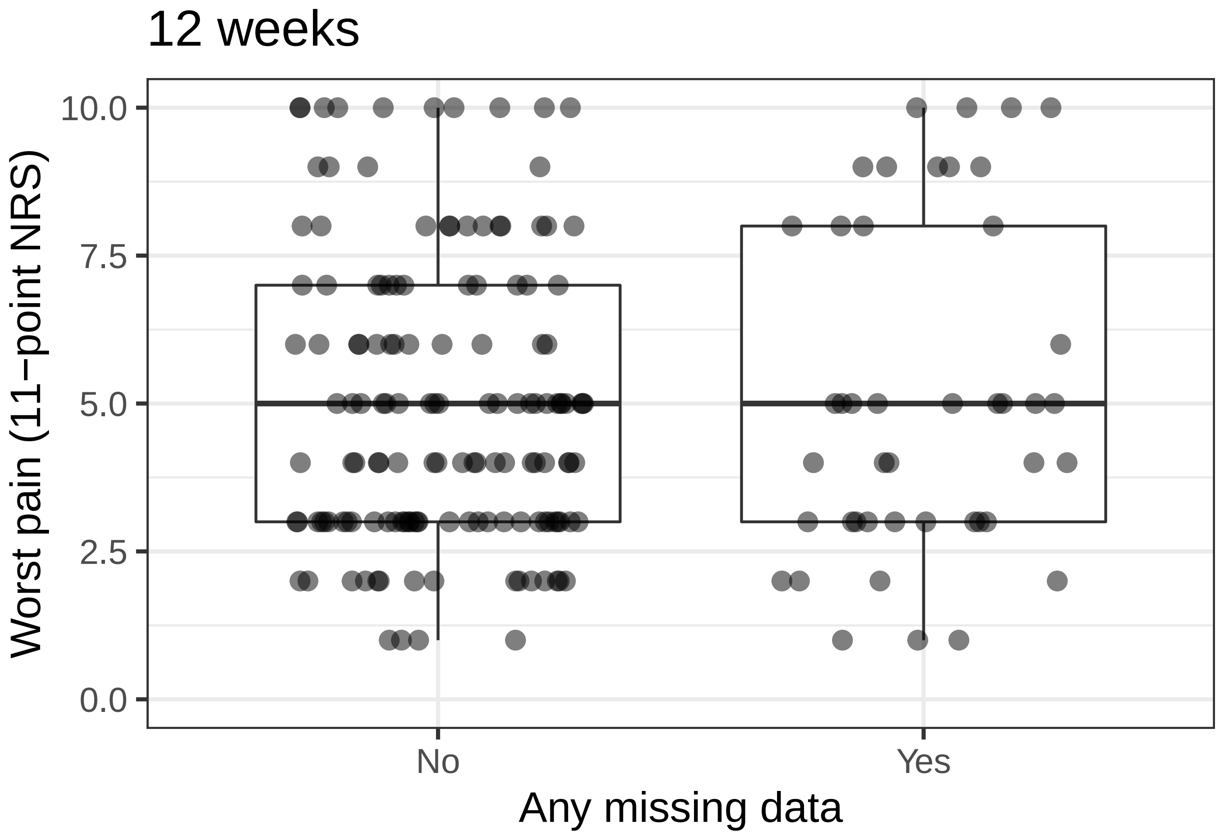
any_missing	median	Q25	Q75
No	5	4.00	7.00
Yes	4	3.75	7.25

Plot of pain intensity at each interval

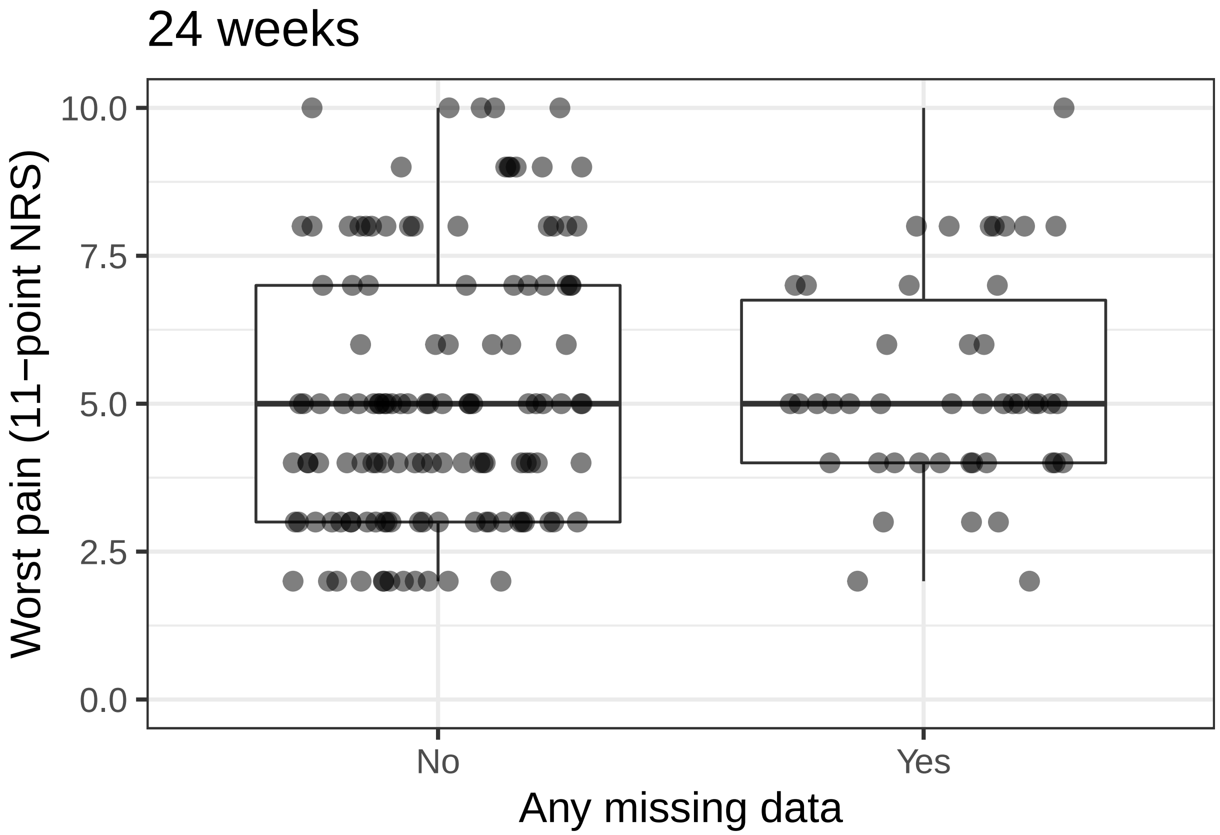
```r
pain_nested <- pain_nested %>%
    mutate(plot = map2(.x = data,
                       .y = interval_name,
                       ~ .x %>%
                           filter(pain_worst > 0) %>%
                           ggplot(data = .) +
                           aes(x = any_missing,
                               y = pain_worst) +
                           geom_boxplot() +
                           geom_point(shape = 21,
                                      size = 3,
                                      fill = '#000000',
                                      alpha = 0.5,
                                      position = position_jitter(height = 0,
                                                                 width = 0.3)) +
                           labs(title = str_glue('{.y}'),
                                x = 'Any missing data',
                                y = 'Worst pain (11-point NRS)') +
                           scale_y_continuous(limits = c(0, 10))))

print(pain_nested$plot[[1]])
```



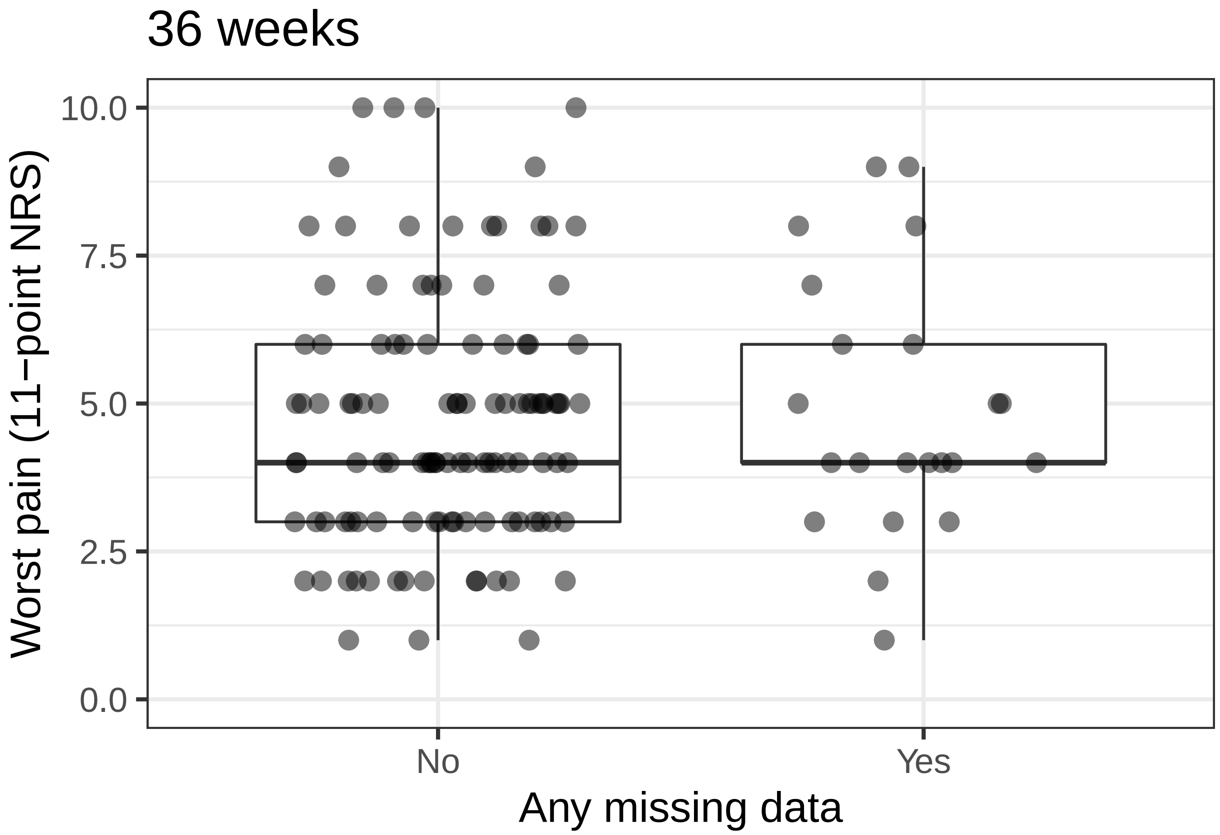
```
print(pain_nested$plot[[2]])
```



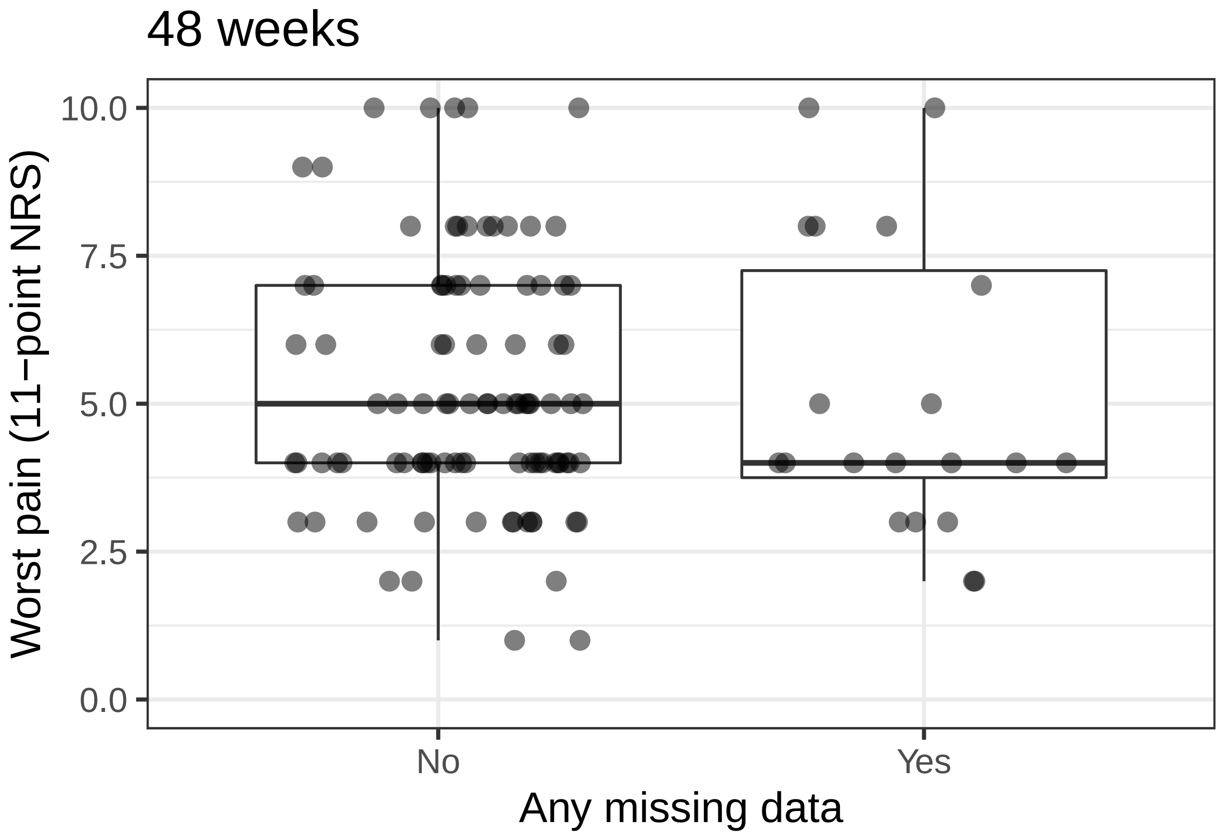
```
print(pain_nested$plot[[3]])
```



```
print(pain_nested$plot[[4]])
```



```
print(pain_nested$plot[[5]])
```



Wilcoxon sum rank test of pain intensity at each interval

```
pain_nested <- pain_nested %>%
    mutate(wilcox = map2(.x = data,
                         .y = interval_name,
                         ~ .x %>%
                             filter(pain_worst > 0) %>%
                             wilcox.test(data = ., pain_worst ~ any_missing) %>%
                             tidy(.) %>%
                             kable(caption = str_glue('{str_to_sentence(.y)}: missing pain data by pain > 0')

walk(.x = pain_nested$wilcox, ~print(.x))
```

Table 17: 0 weeks: missing pain data by pain > 0

statistic	p.value	method	alternative
4937	0.5307091	Wilcoxon rank sum test with continuity correction	two.sided

Table 18: 12 weeks: missing pain data by pain > 0

statistic	p.value	method	alternative
3077	0.8254935	Wilcoxon rank sum test with continuity correction	two.sided

Table 19: 24 weeks: missing pain data by pain > 0

statistic	p.value	method	alternative
2607	0.2747769	Wilcoxon rank sum test with continuity correction	two.sided

Table 20: 36 weeks: missing pain data by pain > 0

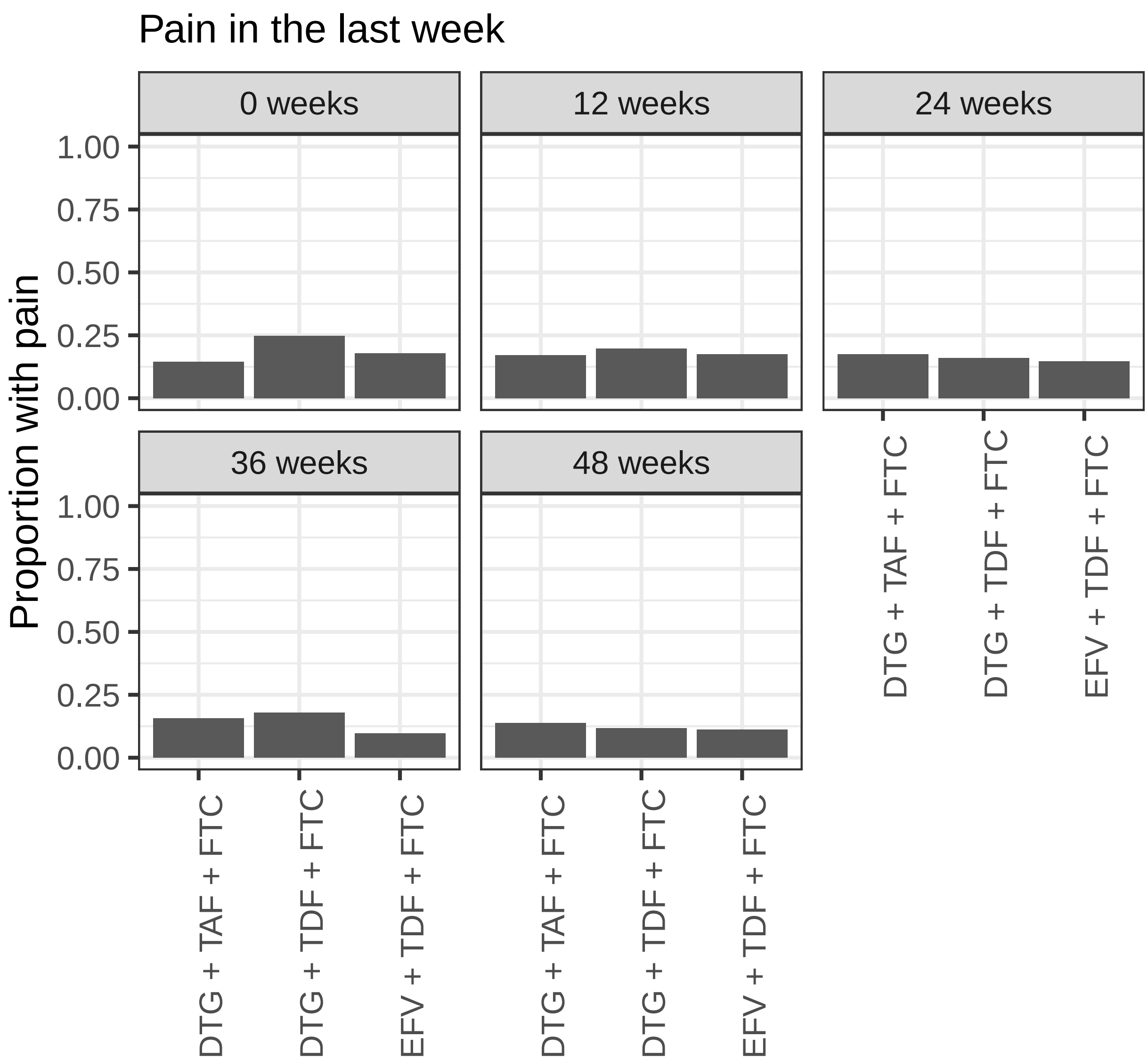
statistic	p.value	method	alternative
1188	0.695158	Wilcoxon rank sum test with continuity correction	two.sided

Table 21: 48 weeks: missing pain data by pain > 0

statistic	p.value	method	alternative
1052.5	0.4941846	Wilcoxon rank sum test with continuity correction	two.sided

Pain frequency by treatment group allocation

```r
# Process data
freq <- df %>%
  select(interval_name, group, pain_in_the_last_week) %>%
  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
freq %>%
  ggplot(data = .) +
  aes(x = group,
      y = proportion) +
  geom_col() +
  labs(subtitle = 'Pain in the last week',
       y = 'Proportion with pain') +
  scale_y_continuous(limits = c(0, 1)) +
  theme(axis.text.x = element_text(angle = 90,
                                    hjust = 0),
        axis.title.x = element_blank()) +
  facet_wrap(~interval_name)
```



```
# Analyse data
xfreq <- df %>%
  group_by(interval_name) %>%
  nest() %>%
  mutate(xfreq = map(.x = data,
                     ~ xtabs(~pain_in_the_last_week + group, data = .x))) %>%
  mutate(chi = map(.x = xfreq,
                   ~ chisq.test(.x) %>%
                     broom::tidy(.)))

walk2(.x = xfreq$chi,
      .y = xfreq$interval_name,
      ~ print(kable(.x, caption = .y)))
```

Table 22: 0 weeks

statistic	p.value	parameter	method
9.564672	0.0083764	2	Pearson's Chi-squared test

Table 23: 12 weeks

statistic	p.value	parameter	method
0.6793423	0.7120044	2	Pearson's Chi-squared test

Table 24: 24 weeks

statistic	p.value	parameter	method
0.7531924	0.6861931	2	Pearson's Chi-squared test

Table 25: 36 weeks

statistic	p.value	parameter	method
7.75227	0.0207308	2	Pearson's Chi-squared test

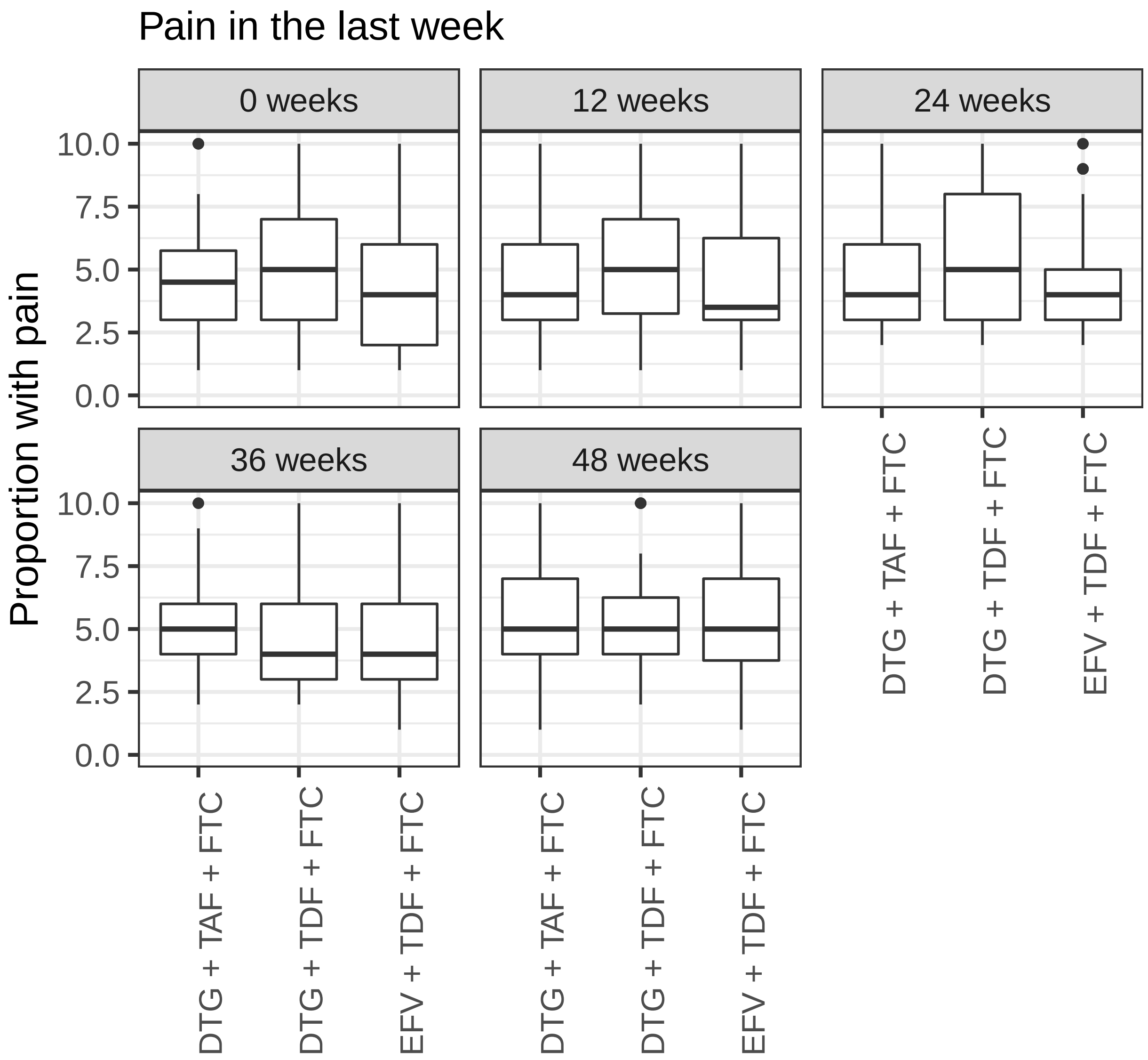
Table 26: 48 weeks

statistic	p.value	parameter	method
0.9057197	0.6358072	2	Pearson's Chi-squared test

No clean pattern in proportion with pain across groups.

Pain intensity by treatment group allocation

```r
# Process data
intensity <- df %>%
  select(interval_name, group, pain_worst) %>%
  filter(pain_worst > 0)

# Plot data
intensity %>%
  ggplot(data = .) +
  aes(x = group,
      y = pain_worst) +
  geom_boxplot() +
  labs(subtitle = 'Pain in the last week',
       y = 'Proportion with pain') +
  scale_y_continuous(limits = c(0, 10)) +
  theme(axis.text.x = element_text(angle = 90,
                                   hjust = 0),
        axis.title.x = element_blank()) +
  facet_wrap(~interval_name)
```



```
# Analyse data
test_intensity <- df %>%
  group_by(interval_name) %>%
  nest() %>%
  mutate(kruskal = map(.x = data,
                       ~ kruskal.test(.x$pain_worst ~ .x$group) %>%
                         broom::tidy(.)))

walk2(.x = test_intensity$kruskal,
      .y = xfreq$interval_name,
      ~ print(kable(.x, caption = .y)))
```

Table 27: 0 weeks

statistic	p.value	parameter	method
10.03363	0.0066256	2	Kruskal-Wallis rank sum test

Table 28: 12 weeks

statistic	p.value	parameter	method
1.131307	0.5679887	2	Kruskal-Wallis rank sum test

Table 29: 24 weeks

statistic	p.value	parameter	method
0.7205333	0.6974903	2	Kruskal-Wallis rank sum test

Table 30: 36 weeks

statistic	p.value	parameter	method
8.073156	0.0176578	2	Kruskal-Wallis rank sum test

Table 31: 48 weeks

statistic	p.value	parameter	method
0.9037845	0.6364227	2	Kruskal-Wallis rank sum test

No clean pattern in intensity of worst pain across groups.

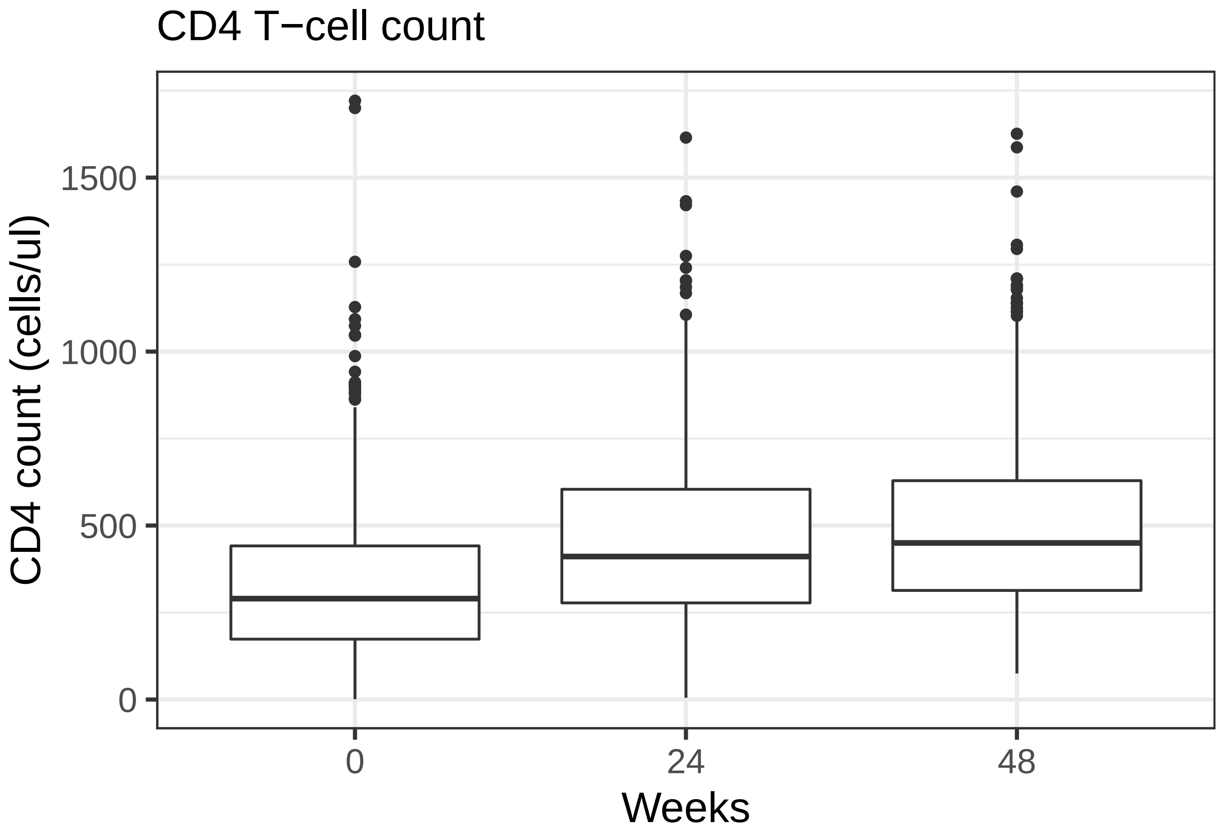
CD4 T-cell count

```r
# Tabulate data
df %>%
  filter(interval_numeric %in% c(0, 24, 48)) %>%
  select(interval_name, cd4_cells.ul) %>%
  group_by(interval_name) %>%
  skim() %>%
  yank('numeric') %>%
  select(-skim_variable, -hist, -complete_rate) %>%
  kable(caption = '7-number summary of CD4 T-cell count (cells/ul)')
```

Table 32: 7-number summary of CD4 T-cell count (cells/ul)

interval_name	n_missing	mean	sd	p0	p25	p50	p75	p100
0 weeks	0	333.2516	224.0549	1	173.50	290	441.50	1721
24 weeks	11	452.7590	237.4241	5	277.75	411	604.25	1615
48 weeks	16	489.8755	246.6119	75	313.50	450	629.00	1626

```r
# Plot data
df %>%
  filter(interval_numeric %in% c(0, 24, 48)) %>%
  ggplot(data = .) +
  aes(x = factor(interval_numeric),
      y = cd4_cells.ul) +
  geom_boxplot() +
  labs(subtitle = 'CD4 T-cell count',
       y = 'CD4 count (cells/ul)',
       x = 'Weeks')
```



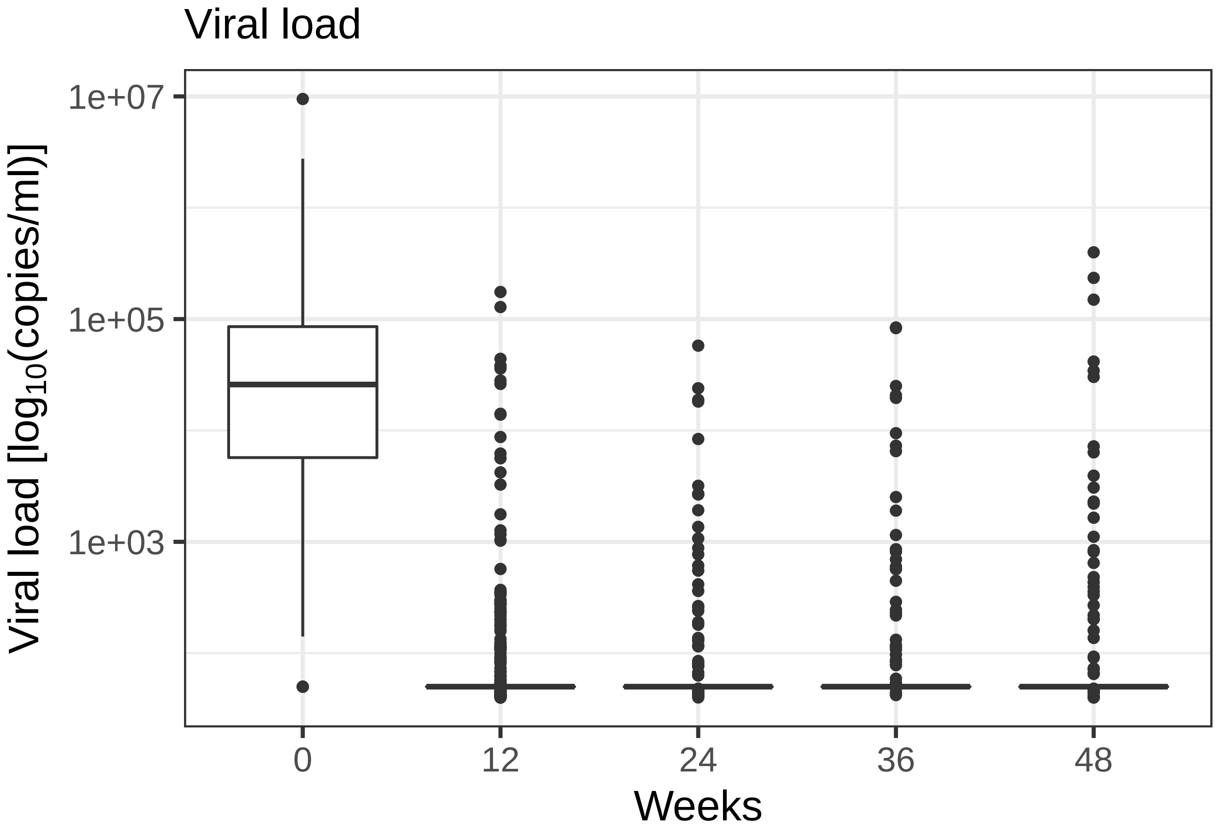
Viral load

```
# Tabulate data
df %>%
  select(interval_name, viral_load_cp.ml) %>%
  group_by(interval_name) %>%
  skim() %>%
  yank('numeric') %>%
  select(-skim_variable, -hist, -complete_rate) %>%
  kable(caption = '7-number summary of viral load (copies/ml)')
```

Table 33: 7-number summary of viral load (copies/ml)

interval_name	n_missing	mean	sd	p0	p25	p50	p75	p100
0 weeks	0	98611.5997	386719.991	50	5704.5	25853	85574	9475772
12 weeks	3	793.2679	8367.886	40	50.0	50	50	175168
24 weeks	2	236.8191	2438.636	40	50.0	50	50	57754
36 weeks	9	391.5308	4466.287	42	50.0	50	50	84167
48 weeks	9	1232.5321	17517.814	40	50.0	50	50	397926

```
# Plot data
df %>%
  ggplot(data = .) +
  aes(x = factor(interval_numeric),
      y = viral_load_cp.ml) +
  geom_boxplot() +
  scale_y_log10() +
  labs(subtitle = 'Viral load',
       y = expression('Viral load [log'[10]*'(copies/ml)]'),
       x = 'Weeks')
```



General health

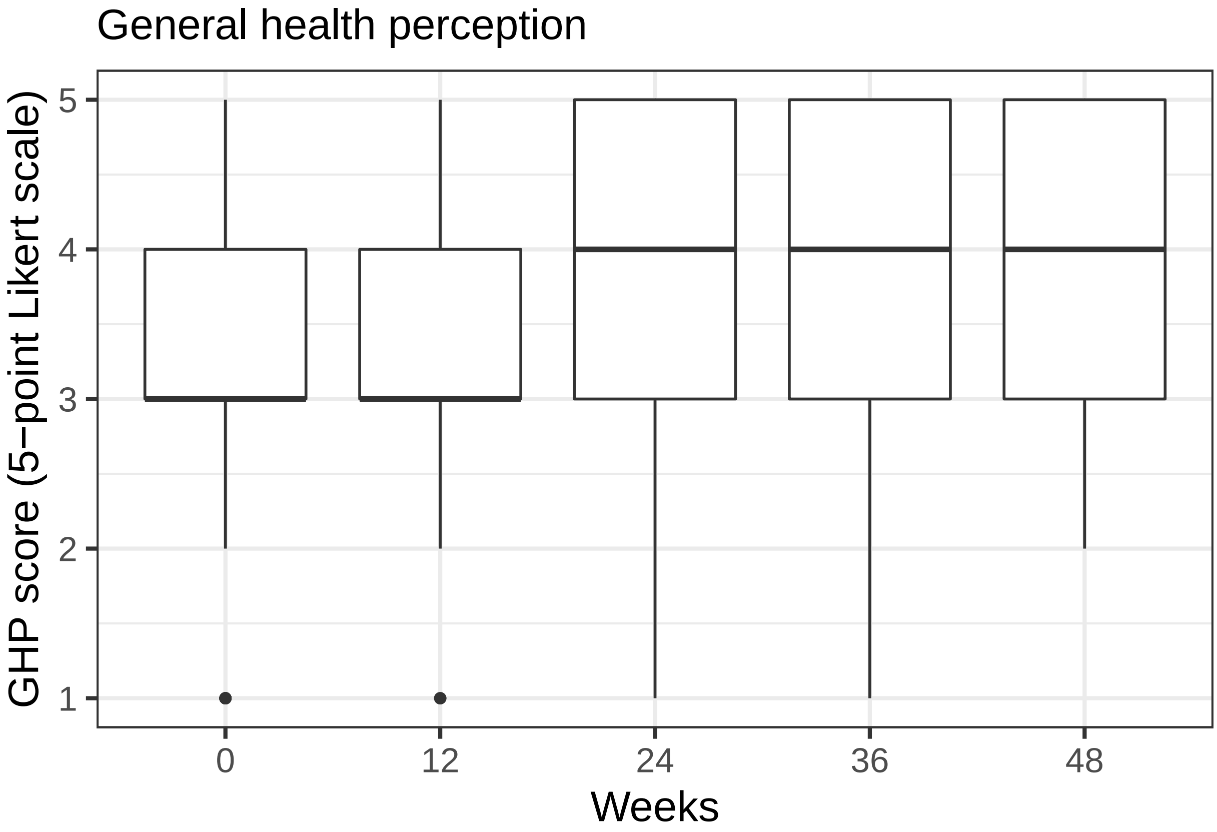
Rating of perceived health status on a 5-point Likert scale (1 = poor, 5 = excellent).

```
# Tabulate data
df %>%
  select(interval_name, general_health) %>%
  group_by(interval_name) %>%
  skim() %>%
  yank('numeric') %>%
  select(-skim_variable, -hist, -complete_rate) %>%
  kable(caption = '7-number summary of the general health score')
```

Table 34: 7-number summary of the general health score

interval_name	n_missing	mean	sd	p0	p25	p50	p75	p100
0 weeks	4	3.449553	0.8152615	1	3	3	4	5
12 weeks	1	3.650127	0.7971166	1	3	3	4	5
24 weeks	0	3.762389	0.8712007	1	3	4	5	5
36 weeks	1	3.807888	0.8858341	1	3	4	5	5
48 weeks	2	3.901911	0.9323051	2	3	4	5	5

```
# Plot data
df %>%
  ggplot(data = .) +
  aes(x = factor(interval_numeric),
      y = general_health) +
  geom_boxplot() +
  labs(subtitle = 'General health perception',
       y = 'GHP score (5-point Likert scale)',
       x = 'Weeks')
```



Session information

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.3
##
## 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] broom_0.5.3        knitr_1.27         skimr_2.0.2        magrittr_1.5
##  [5] forcats_0.4.0      stringr_1.4.0      dplyr_0.8.3        purrr_0.3.3
##  [9] readr_1.3.1        tidyr_1.0.0        tibble_3.0.0       ggplot2_3.3.0.9000
## [13] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
```

```
##  [1] tidyselect_0.2.5 xfun_0.12        repr_1.0.2       haven_2.2.0     
##  [5] lattice_0.20-38  colorspace_1.4-1 vctrs_0.2.4      generics_0.0.2  
##  [9] htmltools_0.4.0  base64enc_0.1-3  yaml_2.2.1       utf8_1.1.4      
## [13] rlang_0.4.5      pillar_1.4.3     withr_2.1.2      glue_1.3.2      
## [17] DBI_1.1.0        dbplyr_1.4.2     modelr_0.1.5     readxl_1.3.1    
## [21] lifecycle_0.2.0  munsell_0.5.0    gtable_0.3.0     cellranger_1.1.0
## [25] rvest_0.3.5      evaluate_0.14    labeling_0.3     fansi_0.4.1     
## [29] highr_0.8        Rcpp_1.0.4       backports_1.1.5  scales_1.1.0    
## [33] jsonlite_1.6.1   farver_2.0.3     fs_1.3.1         hms_0.5.3       
## [37] digest_0.6.25    stringi_1.4.6    grid_3.6.1       cli_2.0.2       
## [41] tools_3.6.1      crayon_1.3.4     pkgconfig_2.0.3  ellipsis_0.3.0  
## [45] xml2_1.2.5       reprex_0.3.0     lubridate_1.7.4  assertthat_0.2.1
## [49] rmarkdown_2.1    httr_1.4.1       rstudioapi_0.11  R6_2.4.1        
## [53] nlme_3.1-143     compiler_3.6.1
```