

# Supplement 2

## Descriptive stats at each time interval

Peter Kamerman

13 February 2020

## Contents

<b>Objective</b>	<b>1</b>
<b>Analysis notes</b>	<b>1</b>
Definitions of missingness . . . . .	1
Definition of data inconsistencies . . . . .	2
<b>Import data</b>	<b>2</b>
<b>First look</b>	<b>2</b>
<b>Basic clean</b>	<b>3</b>
<b>Quick tabulation</b>	<b>3</b>
Analysis data set for the period 0 to 48 weeks . . . . .	3
<b>Analysis</b>	<b>3</b>
Pain frequency by group . . . . .	3
Pain intensity by group . . . . .	5
CD4 T-cell count . . . . .	7
Viral load . . . . .	8
General health . . . . .	9
<b>Session information</b>	<b>10</b>

---

## Objective

To describe the demographic characteristics and disease status of the analysis cohort at each study 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

```
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

```
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>
```

```
glimpse(df)
```

```
## Observations: 5,265
## Variables: 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, ...
## $ 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, ...
## $ 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 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
```

---

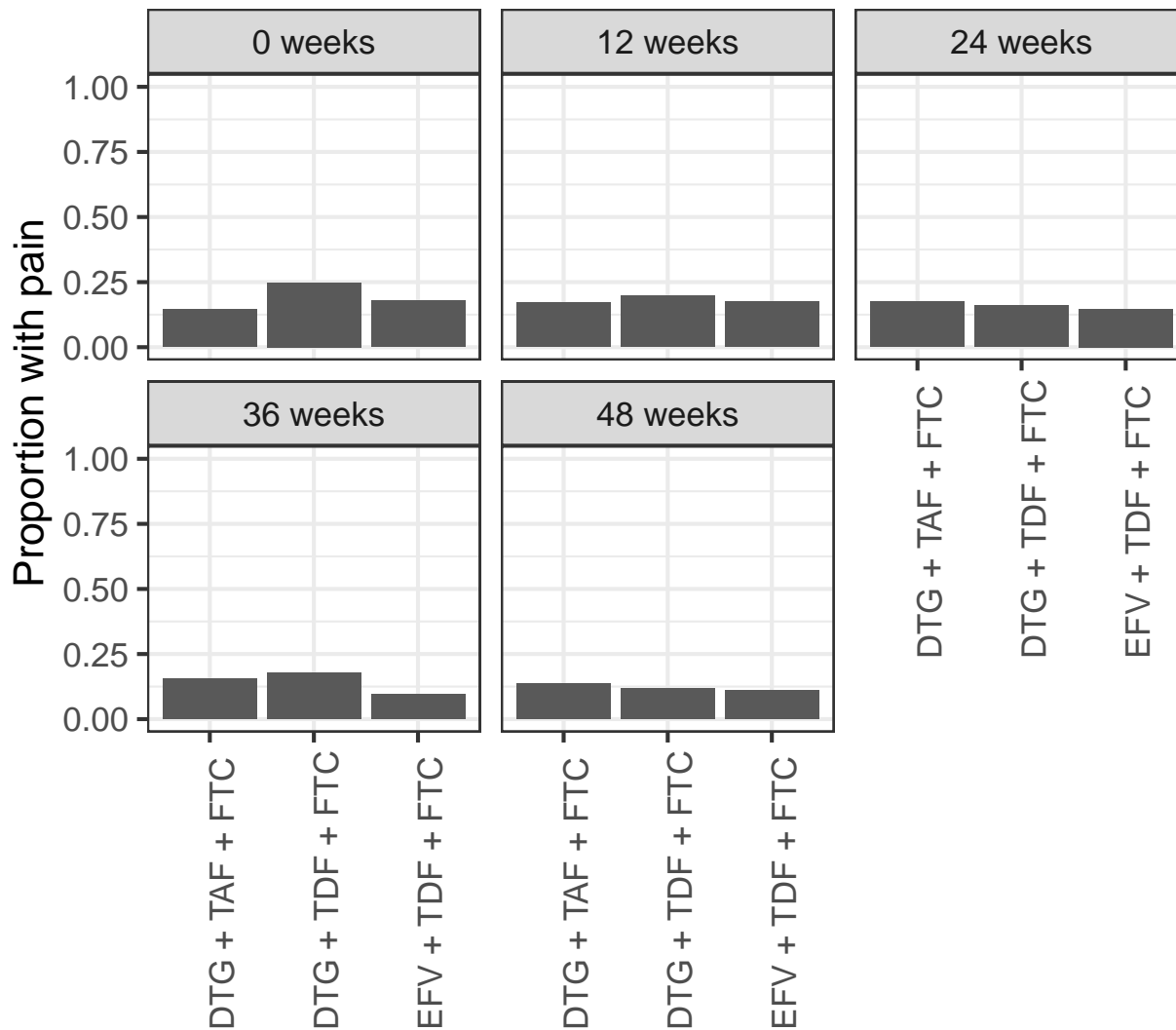
## Analysis

Pain frequency by group

```
# 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)
```

## Pain in the last week



```
# 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: 0 weeks
##
## statistic      p.value    parameter    method
## -----
## 9.564672    0.0083764          2 Pearson's Chi-squared test
##
##
## Table: 12 weeks
```

```
##
## statistic      p.value    parameter  method
## -----
## 0.6793423      0.7120044      2 Pearson's Chi-squared test
##
##
## Table: 24 weeks
##
## statistic      p.value    parameter  method
## -----
## 0.7531924      0.6861931      2 Pearson's Chi-squared test
##
##
## Table: 36 weeks
##
## statistic      p.value    parameter  method
## -----
## 7.75227        0.0207308      2 Pearson's Chi-squared test
##
##
## Table: 48 weeks
##
## statistic      p.value    parameter  method
## -----
## 0.9057197      0.6358072      2 Pearson's Chi-squared test
```

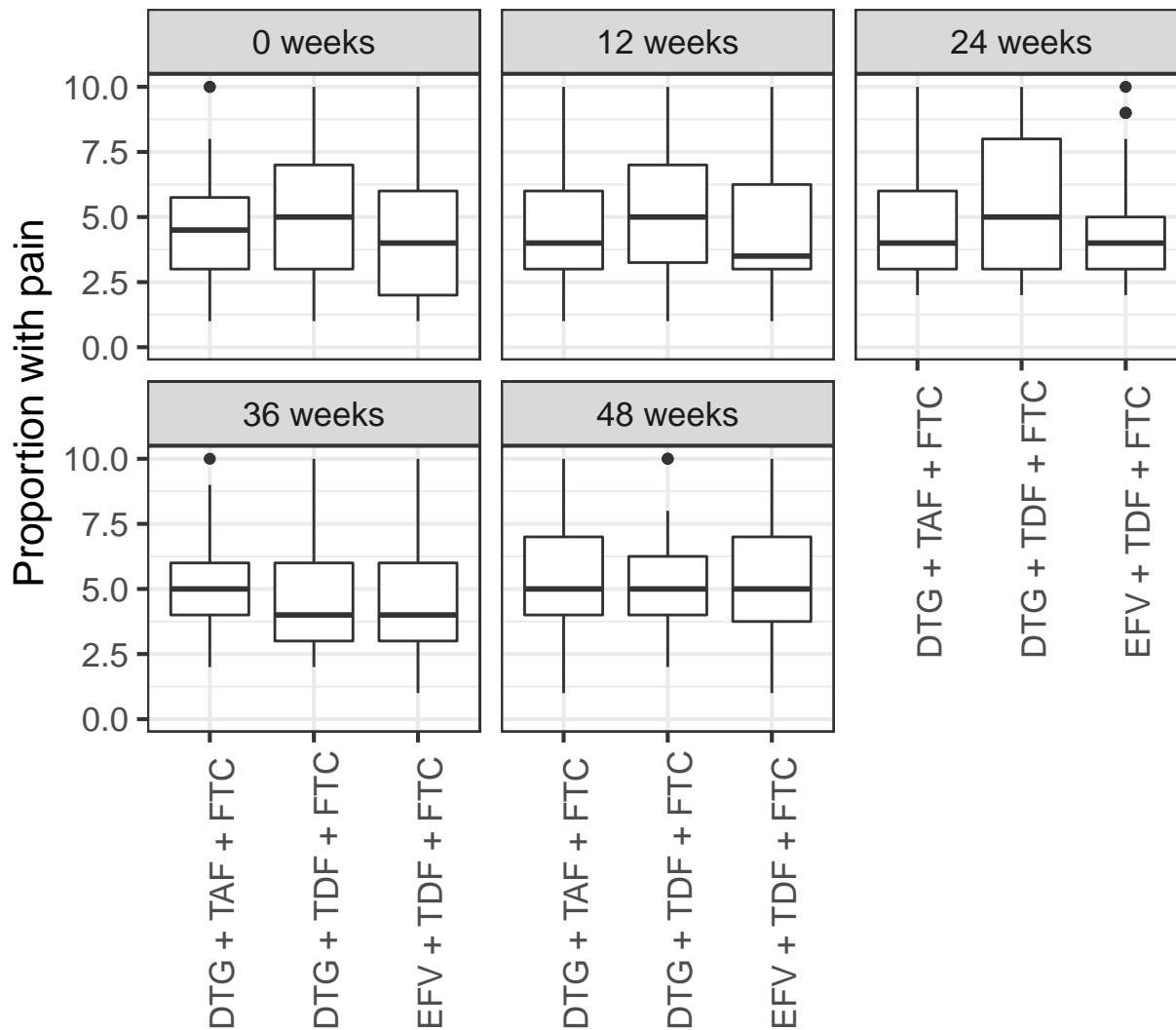
No clean pattern in proportion with pain across groups.

## Pain intensity by group

```
# 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)
```

## Pain in the last week



```
# 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: 0 weeks
##
## statistic      p.value    parameter  method
## -----
## 10.03363      0.0066256          2  Kruskal-Wallis rank sum test
##
##
## Table: 12 weeks
##
## statistic      p.value    parameter  method
```

```
## -----
##      1.131307      0.5679887              2 Kruskal-Wallis rank sum test
##
##
## Table: 24 weeks
##
##      statistic      p.value      parameter      method
## -----
##      0.7205333      0.6974903              2 Kruskal-Wallis rank sum test
##
##
## Table: 36 weeks
##
##      statistic      p.value      parameter      method
## -----
##      8.073156      0.0176578              2 Kruskal-Wallis rank sum test
##
##
## Table: 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

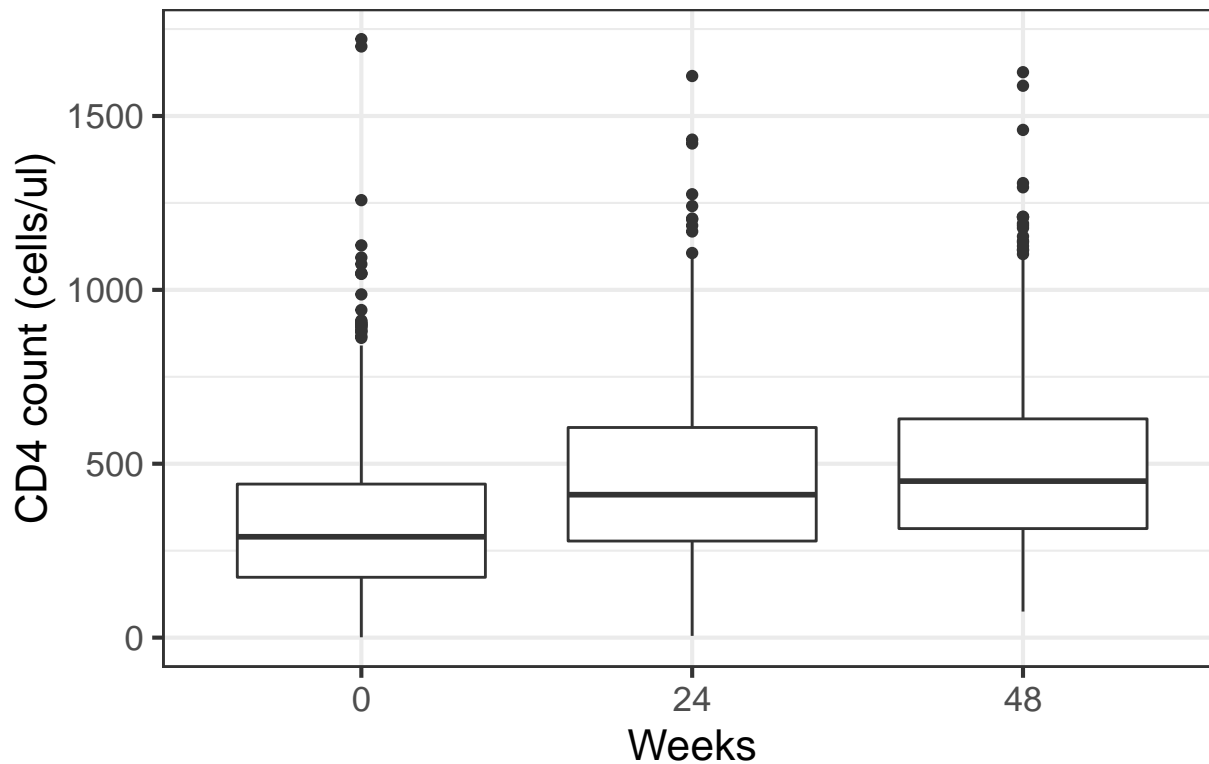
```
# Tabulate data
df %>%
  filter(interval_numeric %in% c(0, 24, 48)) %>%
  select(interval_name, cd4_cells.ul) %>%
  group_by(interval_name) %>%
  skim() %>%
  yank('numeric') %>%
  select(-skim_variable, -hist, -complete_rate) %>%
  kable(caption = '7-number summary of CD4 T-cell count (cells/ul)')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

Table 1: 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

```
# 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')
```

## CD4 T-cell count



## 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)')
```

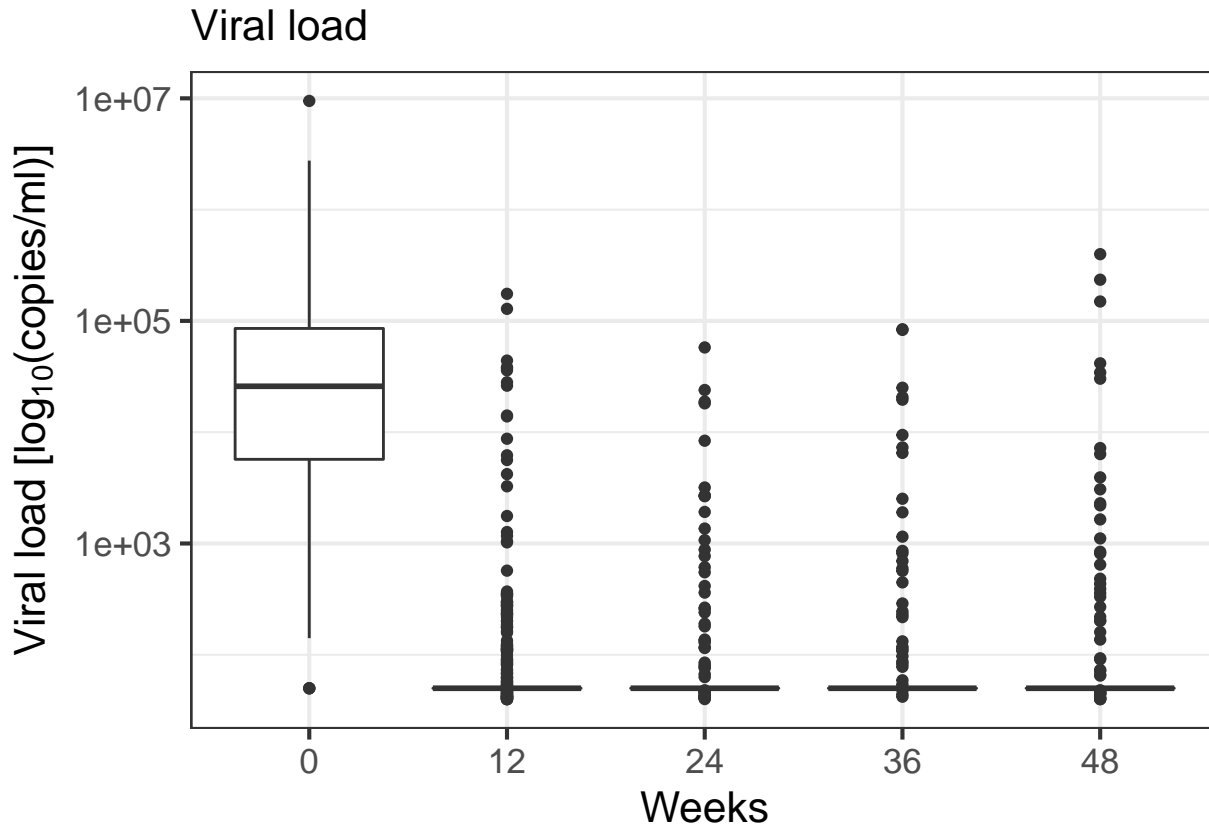
Table 2: 7-number summary of viral load (copies/ml)

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).

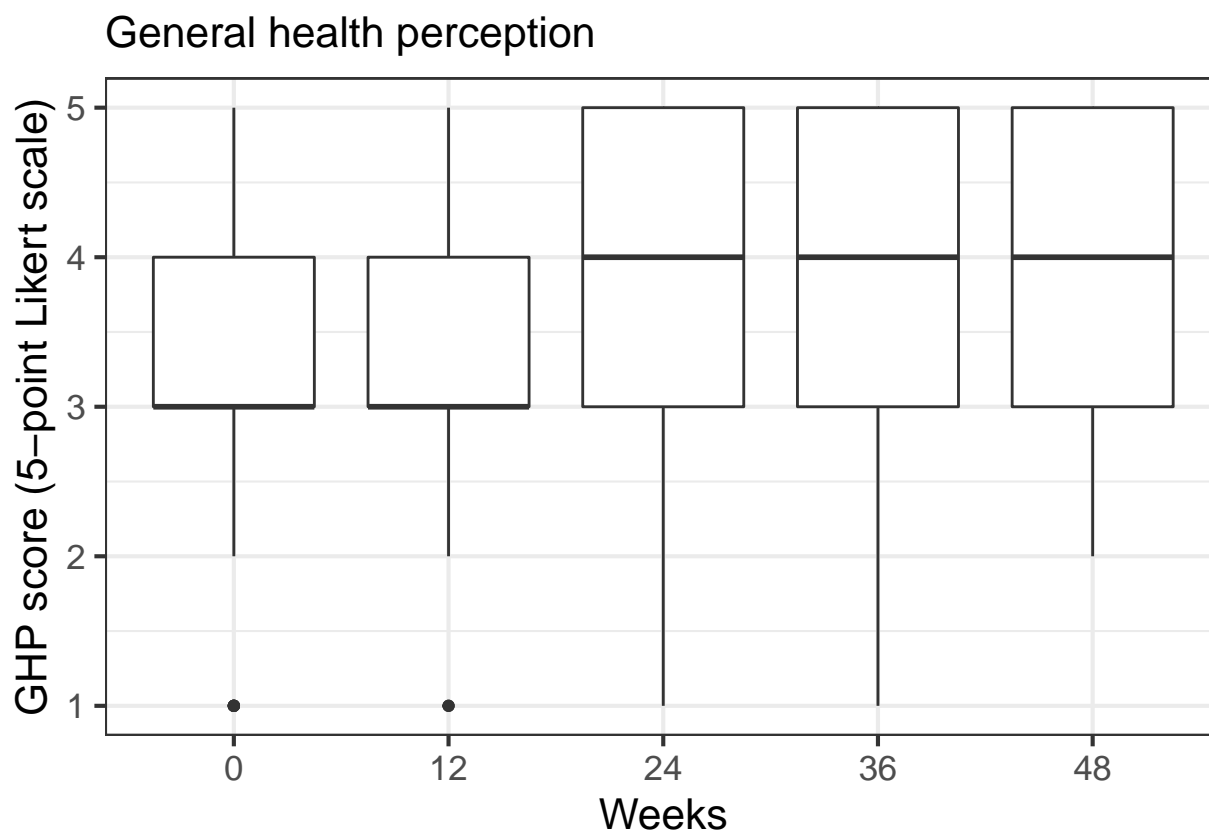
```
# Tabulate data
df %>%
  select(interval_name, general_health) %>%
  group_by(interval_name) %>%
  skim() %>%
  yank('numeric') %>%
  select(-skim_variable, -hist, -complete_rate) %>%
  kable(caption = '7-number summary of the general health score')
```

Table 3: 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 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] knitr_1.27 skimr_2.0.2 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_1.3.0
##
## 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_0.2.1	generics_0.0.2
##	[9]	htmltools_0.4.0	base64enc_0.1-3	yaml_2.2.0	utf8_1.1.4
##	[13]	rlang_0.4.2	pillar_1.4.3	withr_2.1.2	glue_1.3.1
##	[17]	DBI_1.1.0	dbplyr_1.4.2	modelr_0.1.5	readxl_1.3.1
##	[21]	lifecycle_0.1.0	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	broom_0.5.3	Rcpp_1.0.3	scales_1.1.0
##	[33]	backports_1.1.5	jsonlite_1.6	farver_2.0.3	fs_1.3.1
##	[37]	hms_0.5.3	digest_0.6.23	stringi_1.4.5	grid_3.6.1
##	[41]	cli_2.0.1	tools_3.6.1	lazyeval_0.2.2	crayon_1.3.4
##	[45]	pkgconfig_2.0.3	zeallot_0.1.0	xml2_1.2.2	reprex_0.3.0
##	[49]	lubridate_1.7.4	assertthat_0.2.1	rmarkdown_2.1	httr_1.4.1
##	[53]	rstudioapi_0.10	R6_2.4.1	nlme_3.1-143	compiler_3.6.1