

Supplement 2

Data completeness

Peter Kamerman

Last updated: 31 January 2019

Contents

Import data	1
Tabular summary of completeness	2
Clean data	3
Graphical summaries of completeness	4
Study site	4
Study site and intervention group	5
Continuous data collection	10
Prepare data	10
Study site	11
Study site and intervention group	12
Summary	13
Manuscript plot	13
Session information	21

A descriptive analysis of the completeness of data across the 48 weeks of the trial.

Because the BPI was used to collect the primary outcome measure for the study (pain intensity), we assumed the BPI would be the most representative measure completion rates.

Import data

```
# Read in bpi data
bpi <- read_rds('./data-cleaned/bpi.rds')

# Read in site and group info
demo <- read_rds('./data-cleaned/demographics.rds') %>%
  select(ID, Study_site, Group)

# Join the two datasets
bpi %<>%
  left_join(demo)

# Remove demo
rm(demo)
```

Tabular summary of completeness

This summary ignores study site and intervention group stratification.

```
bpi %>%  
  # Remove unneeded columns  
  select(-contains('_rx'), -ID, -Study_site, -Group) %>%  
  # Skim to a df  
  skim_to_wide() %>%  
  # Choose required columns  
  select(variable, missing, complete, n) %>%  
  # Process df to make it more informative  
  separate(col = variable,  
            into = c('variable', 'time'),  
            sep = '\\\\.') %>%  
  mutate(subscale = case_when(  
    variable == 'Pain_present' ~ 'Pain_present',  
    variable == 'Worst_pain' |  
      variable == 'Least_pain' |  
      variable == 'Pain_now' |  
      variable == 'Average_pain' ~ 'Pain_intensity',  
    variable == 'Mood' |  
      variable == 'Sleep' |  
      variable == 'Activities_of_daily_living' |  
      variable == 'Enjoyment_of_life' |  
      variable == 'Relationship_with_others' |  
      variable == 'Walking' |  
      variable == 'Work' ~ 'Pain_interference'  
  )) %>%  
  mutate(subscale = factor(subscale,  
                           levels = c('Pain_present',  
                                       'Pain_intensity',  
                                       'Pain_interference'),  
                           ordered = TRUE),  
         time = factor(time,  
                       levels = c('BL', 'Wk4', 'Wk8',  
                                  'Wk12', 'Wk24', 'Wk48'),  
                       ordered = TRUE)) %>%  
  select(variable, time, subscale, missing, complete, n) %>%  
  arrange(time, subscale) %>%  
  # Filter out pain interference  
  filter(subscale != 'Pain_interference') %>%  
  # Tabulate  
  kable(., caption = 'Tabular summary of data completeness across BPI items')
```

Table 1: Tabular summary of data completeness across BPI items

variable	time	subscale	missing	complete	n
Pain_present	BL	Pain_present	16	144	160
Average_pain	BL	Pain_intensity	51	109	160
Least_pain	BL	Pain_intensity	16	144	160

variable	time	subscale	missing	complete	n
Pain_now	BL	Pain_intensity	16	144	160
Worst_pain	BL	Pain_intensity	16	144	160
Pain_present	Wk4	Pain_present	57	103	160
Average_pain	Wk4	Pain_intensity	89	71	160
Least_pain	Wk4	Pain_intensity	58	102	160
Pain_now	Wk4	Pain_intensity	57	103	160
Worst_pain	Wk4	Pain_intensity	57	103	160
Pain_present	Wk8	Pain_present	58	102	160
Average_pain	Wk8	Pain_intensity	91	69	160
Least_pain	Wk8	Pain_intensity	58	102	160
Pain_now	Wk8	Pain_intensity	58	102	160
Worst_pain	Wk8	Pain_intensity	59	101	160
Pain_present	Wk12	Pain_present	79	81	160
Average_pain	Wk12	Pain_intensity	94	66	160
Least_pain	Wk12	Pain_intensity	79	81	160
Pain_now	Wk12	Pain_intensity	79	81	160
Worst_pain	Wk12	Pain_intensity	79	81	160
Pain_present	Wk24	Pain_present	73	87	160
Average_pain	Wk24	Pain_intensity	104	56	160
Least_pain	Wk24	Pain_intensity	73	87	160
Pain_now	Wk24	Pain_intensity	73	87	160
Worst_pain	Wk24	Pain_intensity	73	87	160
Pain_present	Wk48	Pain_present	82	78	160
Average_pain	Wk48	Pain_intensity	114	46	160
Least_pain	Wk48	Pain_intensity	82	78	160
Pain_now	Wk48	Pain_intensity	82	78	160
Worst_pain	Wk48	Pain_intensity	82	78	160

Although there are small variations in the number of missing data across BPI items, the first item on the BDI assesses whether the participant has pain at the time of completing the questionnaire (**Pain_present**), and it will be used as a proxy of missing data across all other items.

Note that **Average_pain** values show greater incompleteness because average pain data were not recorded at the *R1* study site (n = 47).

Clean data

Retain **Pain_present** data only.

```
# Gather into long format and process time/question column
bpi %<>%
  select(ID, Study_site, Group, starts_with('Pain_present')) %>%
  gather(key = question,
         value = answer,
         -ID, - Study_site, - Group) %>%
  # Separate pain_question into constituent parts
  separate(col = question,
           into = c('question', 'time'),
           sep = '\\.')
```

```

# Convert time points to integer
ungroup() %>%
mutate(time = str_replace(string = time,
                           pattern = 'Wk',
                           replacement = ''),
       time = str_replace(string = time,
                           pattern = 'BL',
                           replacement = '0'),
       time = as.integer(time))

```

Graphical summaries of completeness

These summaries are stratified by study site, and study site and intervention group.

Study site

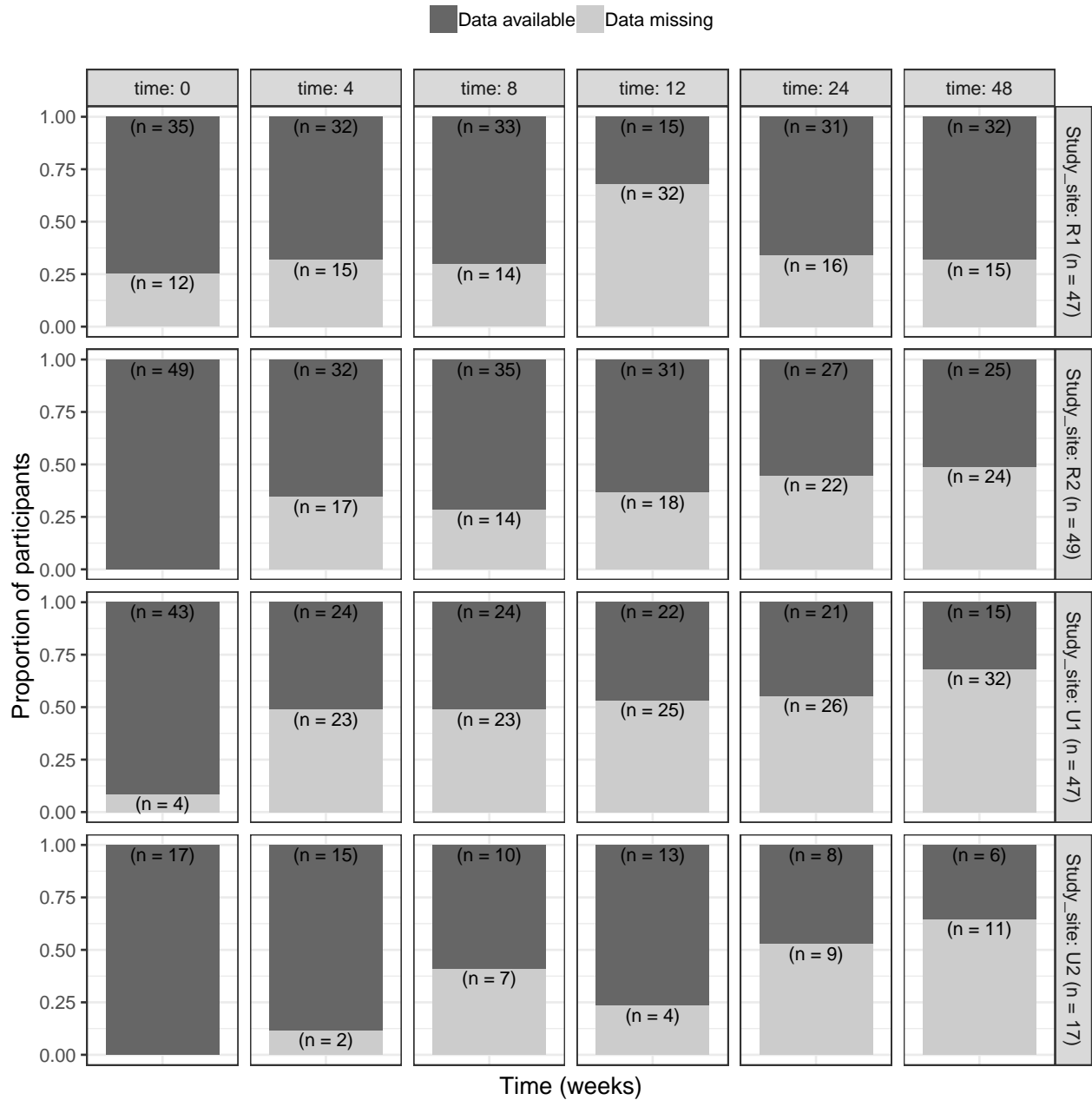
```

bpi %>%
  # Code whether data in bdi_rating is missing or not
  mutate(coding = ifelse(is.na(answer),
                        yes = 'Data missing',
                        no = 'Data available')) %>%
  # Get nominal sample size for each study site
  group_by(Study_site, time) %>%
  mutate(sample_size = n()) %>%
  ungroup() %>%
  mutate(Study_site = paste0(Study_site, ' (n = ', sample_size, ')')) %>%
  # Plot
  ggplot(data = .) +
  aes(x = question) +
  geom_bar(aes(fill = coding),
           position = position_fill()) +
  geom_text(position = position_fill(),
            stat = 'count',
            aes(group = coding,
                label = paste0('(n = ', ..count.., ')')),
            vjust = 1.2) +
  labs(title = 'Completeness of data for the BPI at each study site',
       subtitle = "Nominal sample size at each site is given in the
       'Study_site' facet label",
       x = 'Time (weeks)',
       y = 'Proportion of participants') +
  scale_fill_manual(values = c('#666666', '#CCCCCC')) +
  facet_grid(Study_site ~ time,
             labeller = label_both) +
  theme(legend.position = 'top',
        legend.title = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

```

Completeness of data for the BPI at each study site

Nominal sample size at each site is given in the 'Study_site' facet label



Study site and intervention group

```
complete_group <- bpi %>%
  # Code whether data in bdi_rating is missing or not
  mutate(coding = ifelse(is.na(answer),
    yes = 'Data missing',
    no = 'Data available')) %>%
```

```

# Nest
group_by(Study_site) %>%
nest() %>%
arrange(Study_site) %>%
# Calculate nominal number of participants per study site
mutate(summary_p = map(.x = data,
  ~ filter(.data = .x,
    Group == 'P') %>%
    summarise(count = as.integer(
      sum(!is.na(ID)) / 6))),
  summary_t = map(.x = data,
    ~ filter(.data = .x,
      Group == 'T') %>%
      summarise(count = as.integer(
        sum(!is.na(ID)) / 6)))) %>%

# Plot data
mutate(plot = pmap(.l = list(data, Study_site,
  summary_p, summary_t),
  ~ ggplot(data = ..1) +
    aes(x = question,
      fill = coding) +
    geom_bar(position = position_fill()) +
    geom_text(position = position_fill(),
      stat = 'count',
      aes(label = paste0('(n = ', ..count..., ')')),
      vjust = 1.2) +
    labs(title = str_glue('Site: {..2} - Completeness of data for the BPI for ea
      subtitle = str_glue('Nominal sample size (Group P): {..3}\nNominal samp
      x = 'Time (weeks)',
      y = 'Proportion of participants') +
    scale_fill_manual(values = c('#666666', '#CCCCCC')) +
    facet_grid(Group ~ time,
      labeller = label_both) +
    theme(legend.position = 'top',
      legend.title = element_blank(),
      axis.text.x = element_blank(),
      axis.ticks.x = element_blank()))))

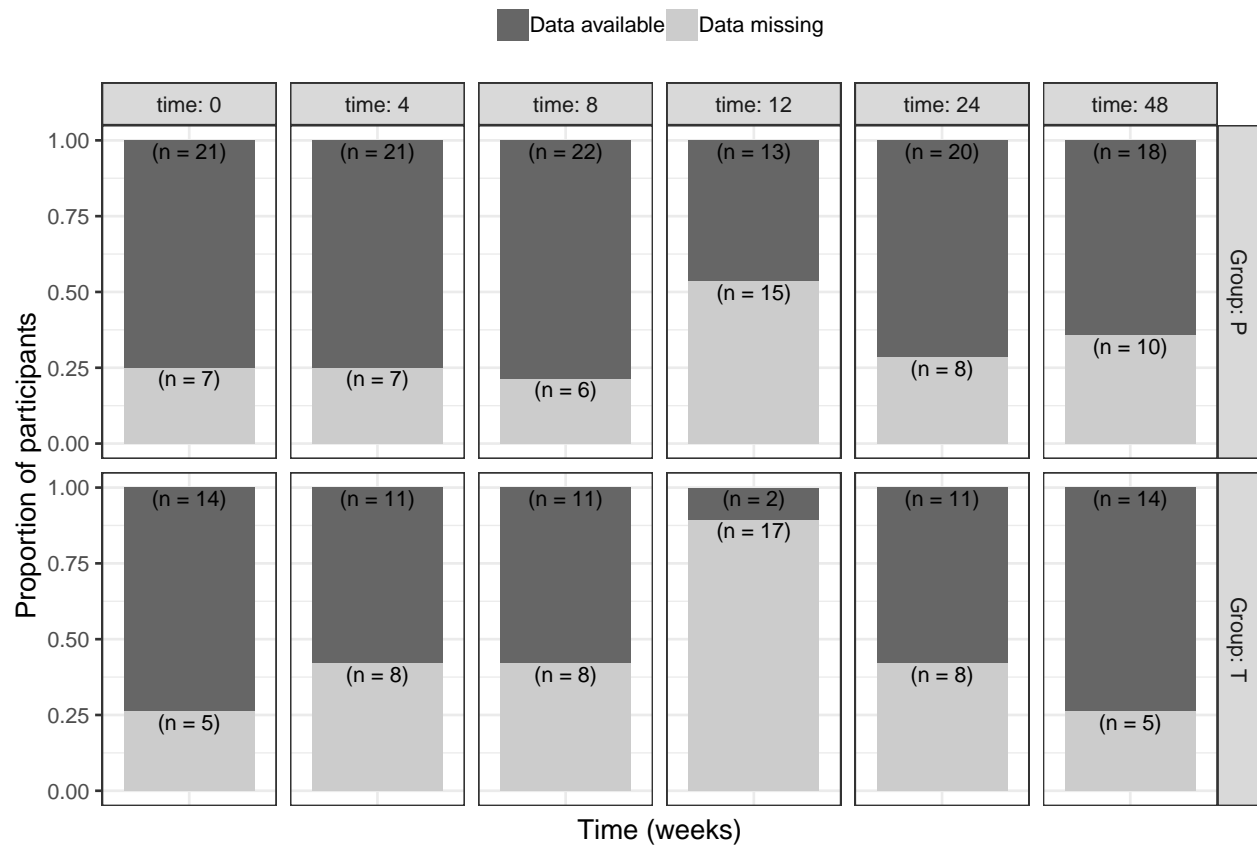
# Print output
walk(.x = complete_group$plot, ~ print(.x))

```

Site: R1 – Completeness of data for the BPI for each intervention group

Nominal sample size (Group P): 28

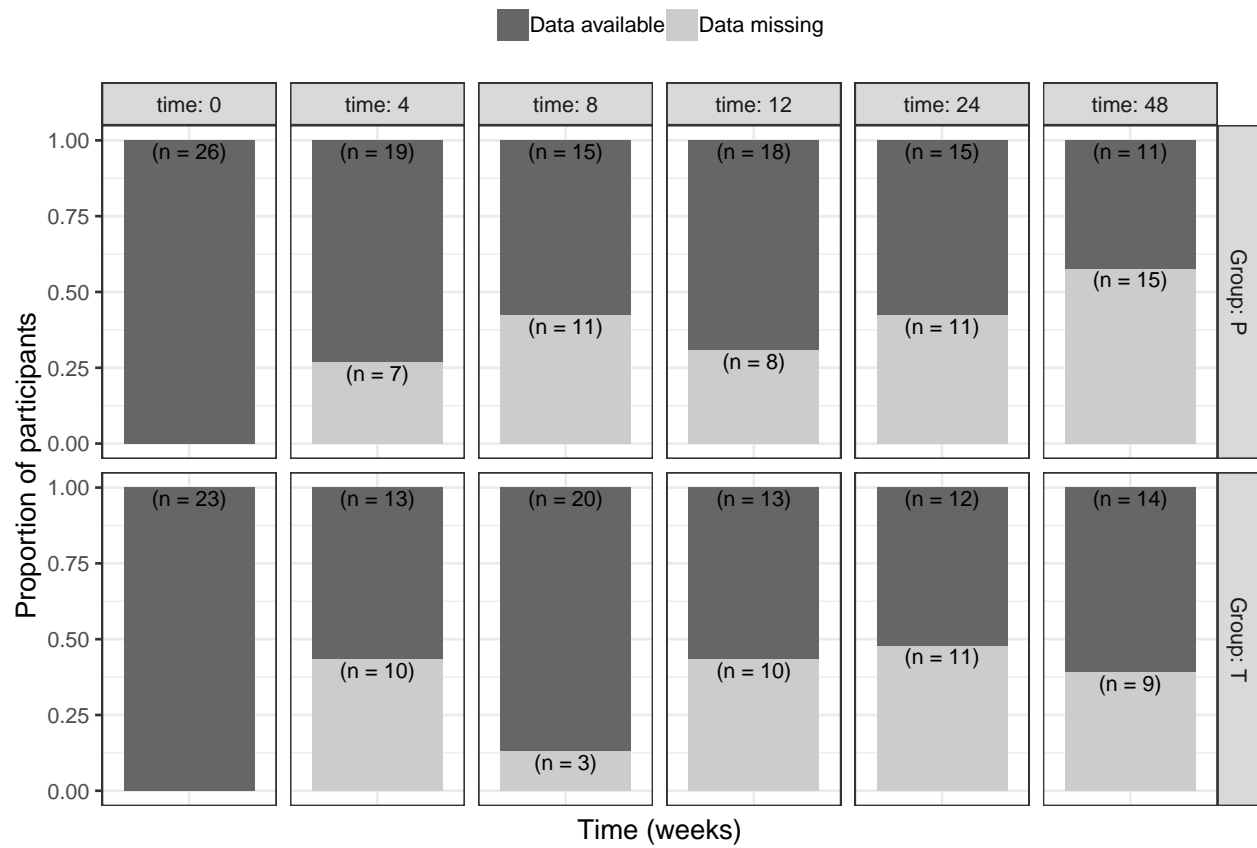
Nominal sample size (Group T): 19



Site: R2 – Completeness of data for the BPI for each intervention group

Nominal sample size (Group P): 26

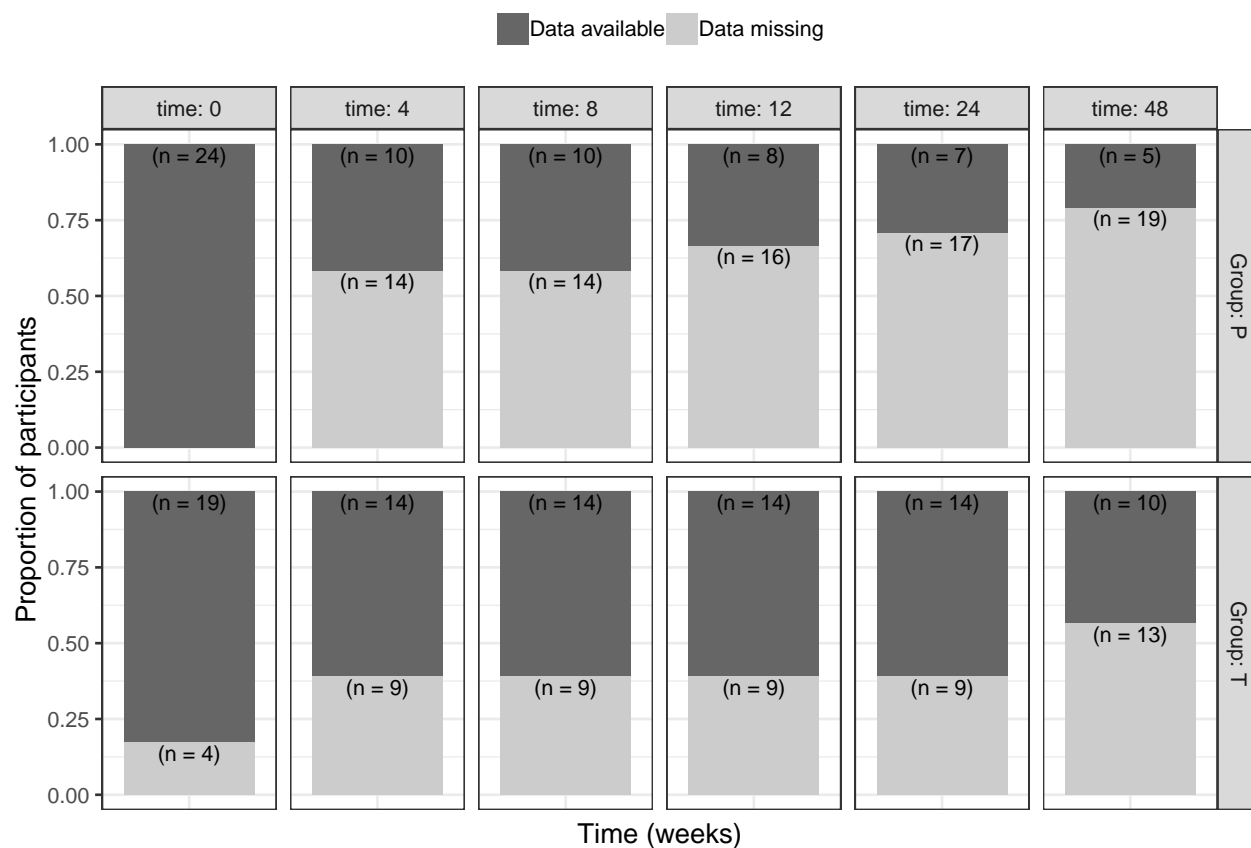
Nominal sample size (Group T): 23



Site: U1 – Completeness of data for the BPI for each intervention group

Nominal sample size (Group P): 24

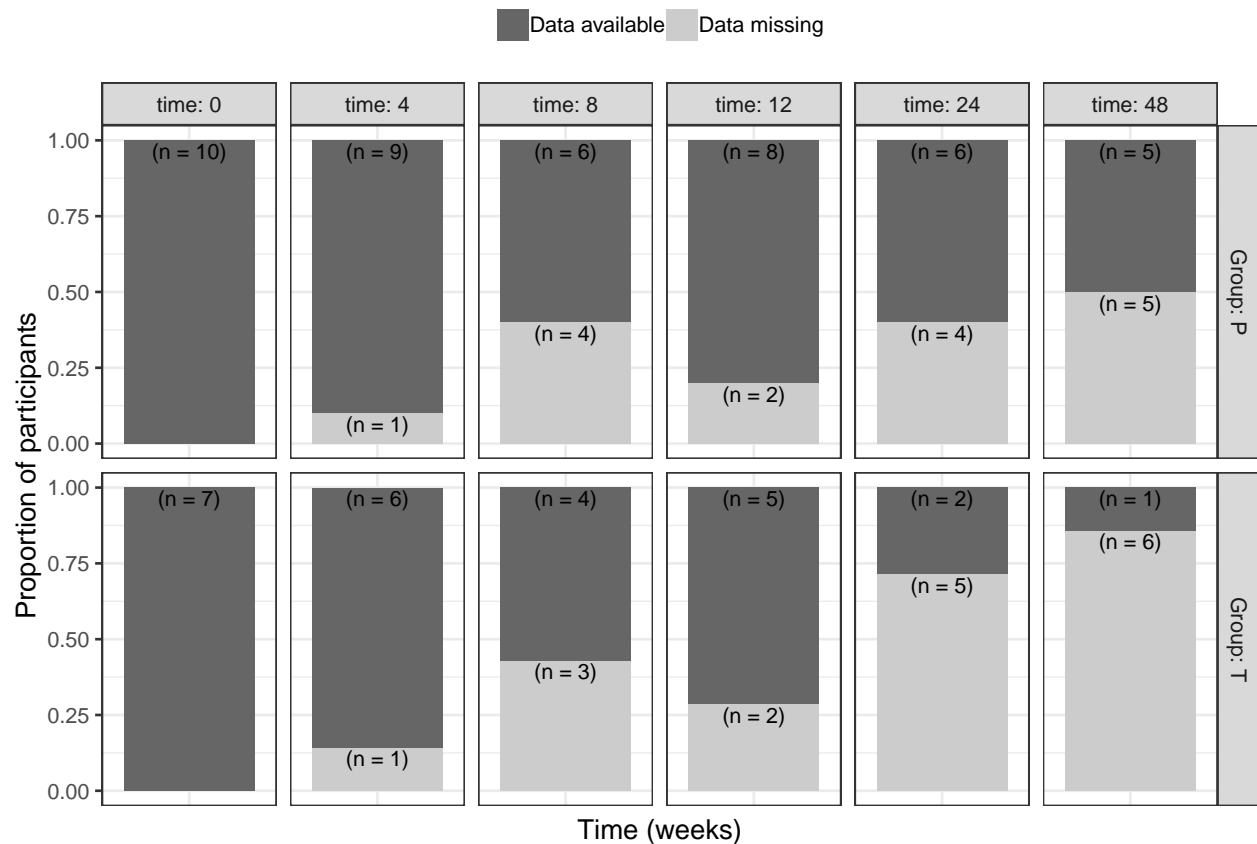
Nominal sample size (Group T): 23



Site: U2 – Completeness of data for the BPI for each intervention group

Nominal sample size (Group P): 10

Nominal sample size (Group T): 7



Continuous data collection

The number of participants with data across successive time points.

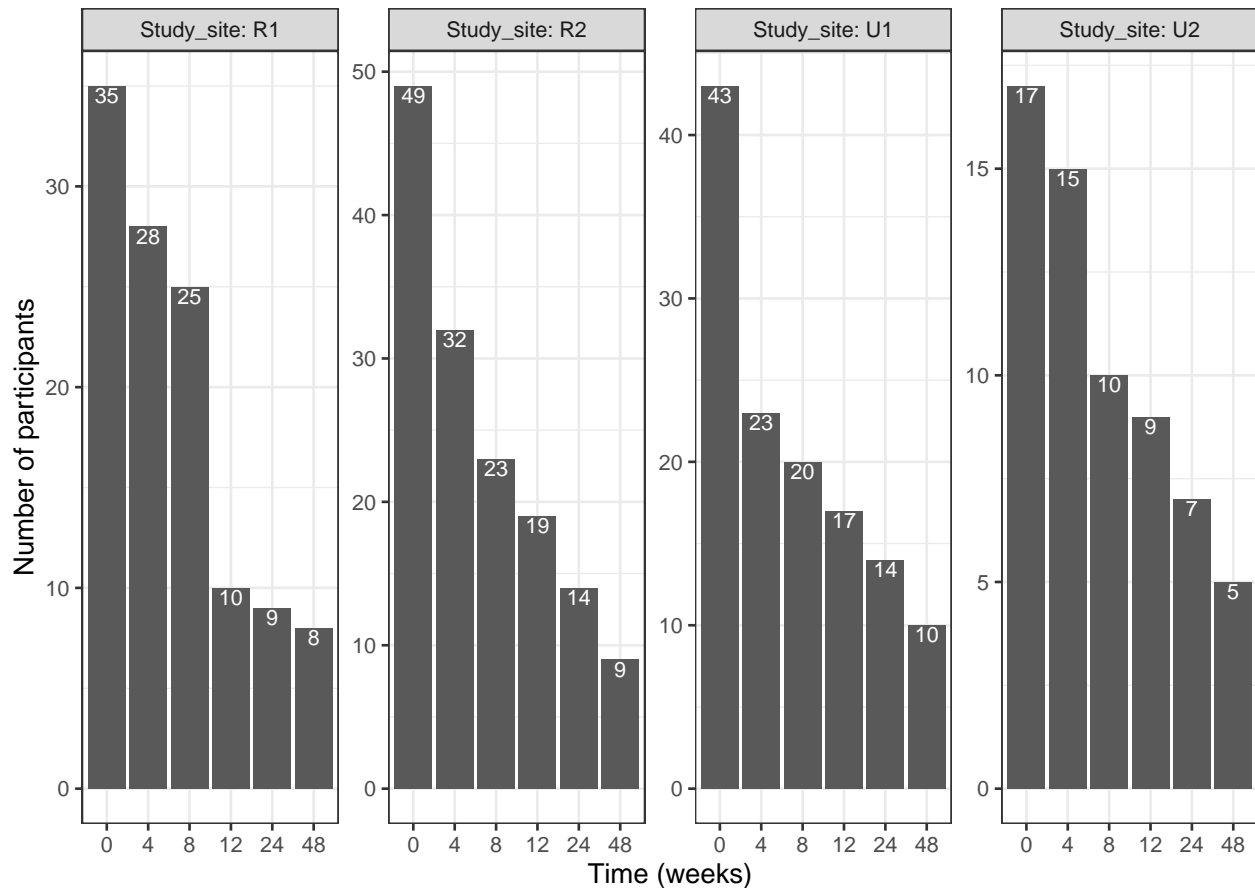
Prepare data

```
bpi_successive <- bpi %>%
  # Recode the answer data to numeric (yes and no are legitimate answers)
  mutate(answer_numeric = case_when(
    answer == 'Yes' ~ '1',
    answer == 'No' ~ '1'
  ),
  answer_numeric = as.integer(answer_numeric)) %>%
  # Generate the counts across time
  group_by(ID) %>%
  mutate(cumulative_data = cumsum(answer_numeric))
```

Study site

```
bpi_successive %>%
  group_by(Study_site, time, ID) %>%
  summarise(count = sum(!is.na(cumulative_data))) %>%
  ungroup() %>%
  # Filter for counts > 0
  filter(count > 0) %>%
  # Recount by Study_site and time
  group_by(Study_site, time) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  # Plot
  ggplot(data = .) +
  aes(x = factor(time),
      y = count) +
  geom_bar(stat = 'identity') +
  geom_text(aes(label = count),
            vjust = 1.1,
            colour = '#FFFFFF') +
  labs(title = 'Number of participants at each study site with data at successive time intervals',
       subtitle = 'Counts shown in each column',
       x = 'Time (weeks)',
       y = 'Number of participants') +
  facet_wrap(~ Study_site,
            ncol = 4,
            labeller = label_both,
            scales = 'free_y')
```

Number of participants at each study site with data at successive time intervals
Counts shown in each column



Study site and intervention group

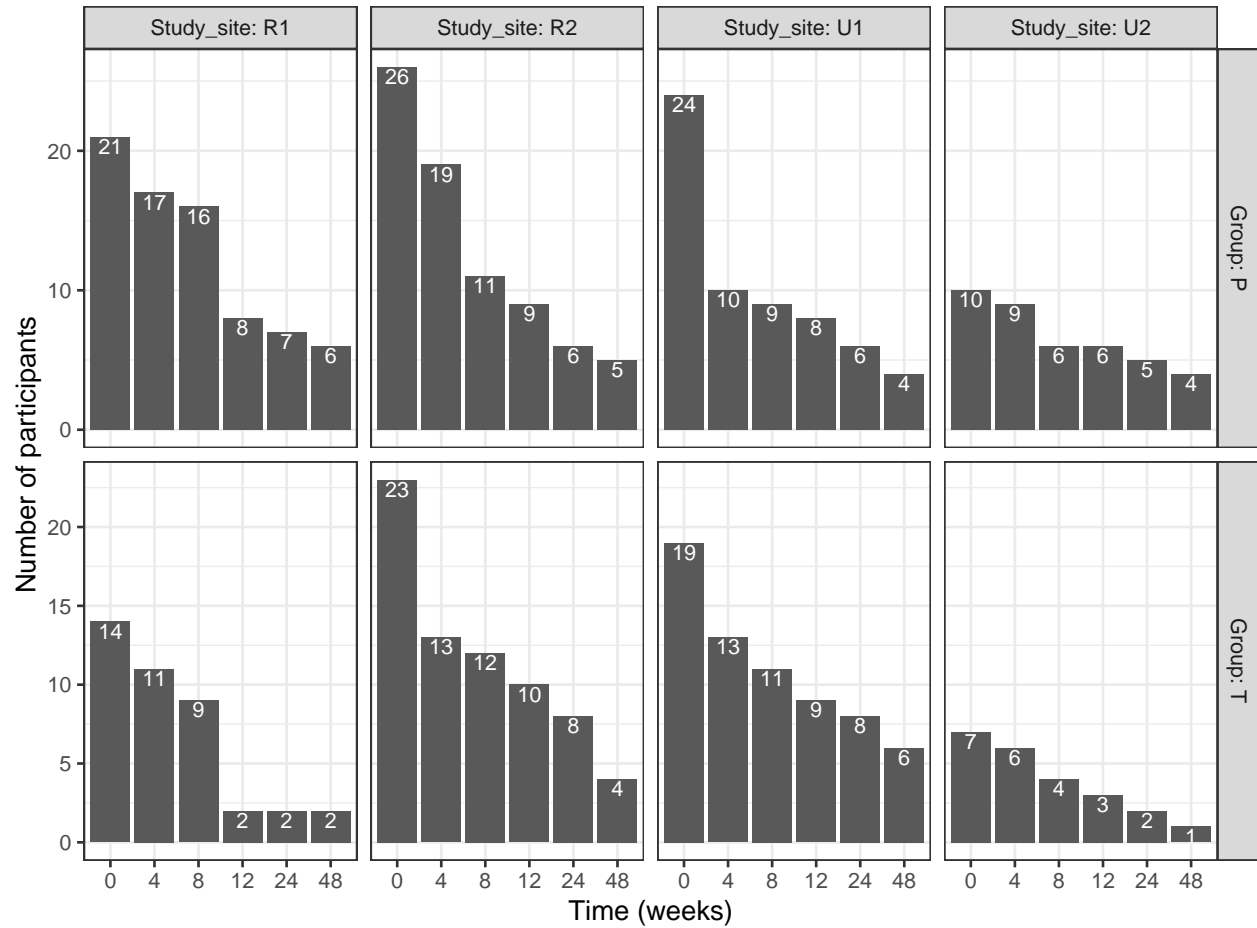
```
bpi_successive %>%
  group_by(Study_site, Group, time, ID) %>%
  summarise(count = sum(!is.na(cumulative_data))) %>%
  ungroup() %>%
  # Filter for counts > 0
  filter(count > 0) %>%
  # Recount by Study_site and time
  group_by(Study_site, Group, time) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  # Plot
  ggplot(data = .) +
  aes(x = factor(time),
      y = count) +
  geom_bar(stat = 'identity') +
  geom_text(aes(label = count),
            colour = '#FFFFFF',
            vjust = 1.1) +
  labs(title = 'Number of participants at each study site with data at successive time intervals',
        x = 'Time (weeks)',
```

```

y = 'Number of participants') +
facet_grid(Group ~ Study_site,
           labeller = label_both,
           scales = 'free_y')

```

Number of participants at each study site with data at successive time intervals



Summary

Other than site *R1*, the other sites have 100% or near 100% (*U1*) records at baseline (time = 0 weeks), thereafter, there is a trend for progressively more incomplete data over time.

The number of participants with continuous data over successive reassessment time-points shows substantial drop-off over the 48 weeks.

Manuscript plot

A figure of loss to follow-up for publication purposes.

The ‘completeness’ plots above catalogued whether data from each participant were missing/available at each time interval without being sensitive to whether participants returned or did not return for reassessment at subsequent time-points. In this analysis, participants were classified as ‘lost to follow-up’ when they had ≥ 2 successive time-points (or week 48 was reached) with missing data, with the time of loss to follow-up being taken as the last time-point for which data were available.

To accomodate erratic attendance at the baseline assessment and subsequent reassessment time-points, we extended our ‘lost to follow-up’ classification to include the following:

- Participants who were recruited, but failed to attend the baseline and week 8 assessments were classified as being lost to follow-up at $-T0$, irrespective of whether they were assessed at other time-points.
- Participants that missed the baseline assessment, but who were reassessed at least at week 8 were classified as lost to follow-up according to the ≥ 2 successive missed reassessment rule, but starting at week 8.

```
# Spread data
bpi_spread <- bpi %>%
  # Code whether data coding data are missing or not
  mutate(coding = ifelse(is.na(answer),
                          yes = '0',
                          no = '1')) %>%
  mutate(coding = as.numeric(coding)) %>%
  select(-answer) %>%
  # Recode time before spreading
  mutate(time = paste0('T', time)) %>%
  # Spread time columns
  spread(key = time,
         value = coding) %>%
  select(ID, T0, T4, T8, T12, T24, T48) %>%
  arrange(T0, T4, T8, T12, T24, T48)

# Get unique combinations
bpi_spread %<>%
  group_by(ID) %>%
  mutate(combos = paste(T0, T4, T8, T12, T24, T48, collapse = ' '))

# Check the combinations
bpi_spread %>%
  .$combos %>%
  unique(.) %>%
  data.frame(combos = .)

##           combos
## 1  0 0 0 0 0 0
## 2  0 0 0 0 0 1
## 3  0 0 0 0 1 0
## 4  0 0 0 0 1 1
## 5  0 1 0 1 1 1
## 6  0 1 1 0 1 1
## 7  0 1 1 1 0 0
## 8  0 1 1 1 1 0
## 9  0 1 1 1 1 1
## 10 1 0 0 0 0 0
## 11 1 0 0 0 0 1
## 12 1 0 0 1 0 0
```

```
## 13 1 0 0 1 1 1
## 14 1 0 1 0 0 0
## 15 1 0 1 0 0 1
## 16 1 0 1 0 1 1
## 17 1 0 1 1 0 1
## 18 1 0 1 1 1 0
## 19 1 0 1 1 1 1
## 20 1 1 0 0 0 0
## 21 1 1 0 0 0 1
## 22 1 1 0 0 1 0
## 23 1 1 0 0 1 1
## 24 1 1 0 1 0 0
## 25 1 1 0 1 0 1
## 26 1 1 0 1 1 0
## 27 1 1 1 0 0 0
## 28 1 1 1 0 0 1
## 29 1 1 1 0 1 0
## 30 1 1 1 0 1 1
## 31 1 1 1 1 0 0
## 32 1 1 1 1 0 1
## 33 1 1 1 1 1 0
## 34 1 1 1 1 1 1
```

```
# Manually specify combos (need to find a way of automating this)
#
# The case_when booleans assign participants with the last visit at which
# they had data recorded (i.e., data missing from future time-points).
#
# Gaps in data have been allowed for, so missing a visit did not result in a
# participant being marked a lost to follow-up at future dates, if they returned
# at some point.
#
# People without week 8 data (T8) and no baseline data (T0) were coded as -T0
# (i.e., recruited and consented, but did not take part in the study).
```

```
bpi_spread %<>%
  mutate(Time_of_loss = case_when(
    T0 == '0' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '0' ~ '-T0',
    T0 == '0' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '1' ~ '-T0',
    T0 == '0' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '1' & T48 == '0' ~ '-T0',
    T0 == '0' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '1' & T48 == '1' ~ '-T0',
    T0 == '0' & T4 == '1' & T8 == '0' & T12 == '1' & T24 == '1' & T48 == '1' ~ '-T0',
    T0 == '0' & T4 == '1' & T8 == '1' & T12 == '0' & T24 == '1' & T48 == '1' ~ 'T48',
    T0 == '0' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '0' & T48 == '0' ~ 'T12',
    T0 == '0' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '0' ~ 'T24',
    T0 == '0' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '1' ~ 'T48',
    T0 == '1' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '0' ~ 'T0',
    T0 == '1' & T4 == '0' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '1' ~ 'T0',
    T0 == '1' & T4 == '0' & T8 == '0' & T12 == '1' & T24 == '0' & T48 == '0' ~ 'T0',
    T0 == '1' & T4 == '0' & T8 == '0' & T12 == '1' & T24 == '1' & T48 == '1' ~ 'T0',
    T0 == '1' & T4 == '0' & T8 == '1' & T12 == '0' & T24 == '0' & T48 == '0' ~ 'T8',
    T0 == '1' & T4 == '0' & T8 == '1' & T12 == '0' & T24 == '0' & T48 == '1' ~ 'T48',
    T0 == '1' & T4 == '0' & T8 == '1' & T12 == '0' & T24 == '1' & T48 == '1' ~ 'T48',
    T0 == '1' & T4 == '0' & T8 == '1' & T12 == '1' & T24 == '0' & T48 == '1' ~ 'T48',
```

```

T0 == '1' & T4 == '0' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '0' ~ 'T24',
T0 == '1' & T4 == '0' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '1' ~ 'T48',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '0' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '0' & T24 == '0' & T48 == '1' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '0' & T24 == '1' & T48 == '0' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '0' & T24 == '1' & T48 == '1' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '1' & T24 == '0' & T48 == '0' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '1' & T24 == '0' & T48 == '1' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '0' & T12 == '1' & T24 == '1' & T48 == '0' ~ 'T4',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '0' & T24 == '0' & T48 == '0' ~ 'T8',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '0' & T24 == '0' & T48 == '1' ~ 'T48',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '0' & T24 == '1' & T48 == '0' ~ 'T24',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '0' & T24 == '1' & T48 == '1' ~ 'T48',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '0' & T48 == '0' ~ 'T12',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '0' & T48 == '1' ~ 'T48',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '0' ~ 'T24',
T0 == '1' & T4 == '1' & T8 == '1' & T12 == '1' & T24 == '1' & T48 == '1' ~ 'T48'
))

# Tabulate
bpi_spread %>%
  group_by(combos, Time_of_loss) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  mutate(Time_of_loss = factor(Time_of_loss,
                                levels = c('-T0', 'T0', 'T4', 'T8',
                                              'T12', 'T24', 'T48'),
                                ordered = TRUE)) %>%
  arrange(Time_of_loss, desc(count)) %>%
  mutate(combos = str_replace_all(combos,
                                    pattern = '0',
                                    replacement = '_'),
         combos = str_replace_all(combos,
                                    pattern = '1',
                                    replacement = 'A')) %>%
  kable(., caption = 'Loss to follow-up classification by attendance sequence',
        col.names = c('Attendance sequence',
                       'Lost to follow-up classification',
                       'Number of participants'))

```

Table 2: Loss to follow-up classification by attendance sequence

Attendance sequence	Lost to follow-up classification	Number of participants
-----	-T0	8
-----A	-T0	1
-----A	-T0	1
-----AA	-T0	1
-A-AA	-T0	1
A-----	T0	22
A--A--	T0	2
A---A	T0	1
A--AA	T0	1
AA----	T4	5
AA--A	T4	4

Attendance sequence	Lost to follow-up classification	Number of participants
A A _ A _ _	T4	4
A A _ A A _	T4	4
A A _ _ A _	T4	1
A A _ _ A A	T4	1
A A _ A _ A	T4	1
A A A _ _ _	T8	5
A _ A _ _ _	T8	2
A A A A _ _	T12	7
_ A A A _ _	T12	1
A A A A A _	T24	12
A _ A A A _	T24	4
A A A _ A _	T24	3
_ A A A A _	T24	1
A A A A A A	T48	32
A A A _ A A	T48	14
A _ A _ A A	T48	5
A _ A A A A	T48	4
A A A A _ A	T48	4
A _ A _ _ A	T48	3
A _ A A _ A	T48	2
_ A A _ A A	T48	1
_ A A A A A	T48	1
A A A _ _ A	T48	1

```

# Drop columns
bpi_time <- bpi_spread %>%
  select(ID, Time_of_loss) %>%
  mutate(counter = '1',
         Time_of_loss = factor(Time_of_loss,
                               levels = c('-T0', 'T0', 'T4', 'T8',
                                           'T12', 'T24', 'T48'),
                               ordered = TRUE))

# Generate plot data
bpi_summary <- bpi_time %>%
  # How many people are have data at a given time interval
  group_by(Time_of_loss) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  # Calculate the cumulative loss
  mutate(cumulative = cumsum(count)) %>%
  # Get the reverse number (how many people add to the data at each time interval)
  mutate(rev_cumulative = nrow(bpi_time) - cumulative) %>%
  # Get the lag 1 rev_cumulative value
  mutate(lag_cumulative = lag(rev_cumulative))

# Add lag_cumulative value for -T0
bpi_summary[1, 5] <- nrow(bpi_time)

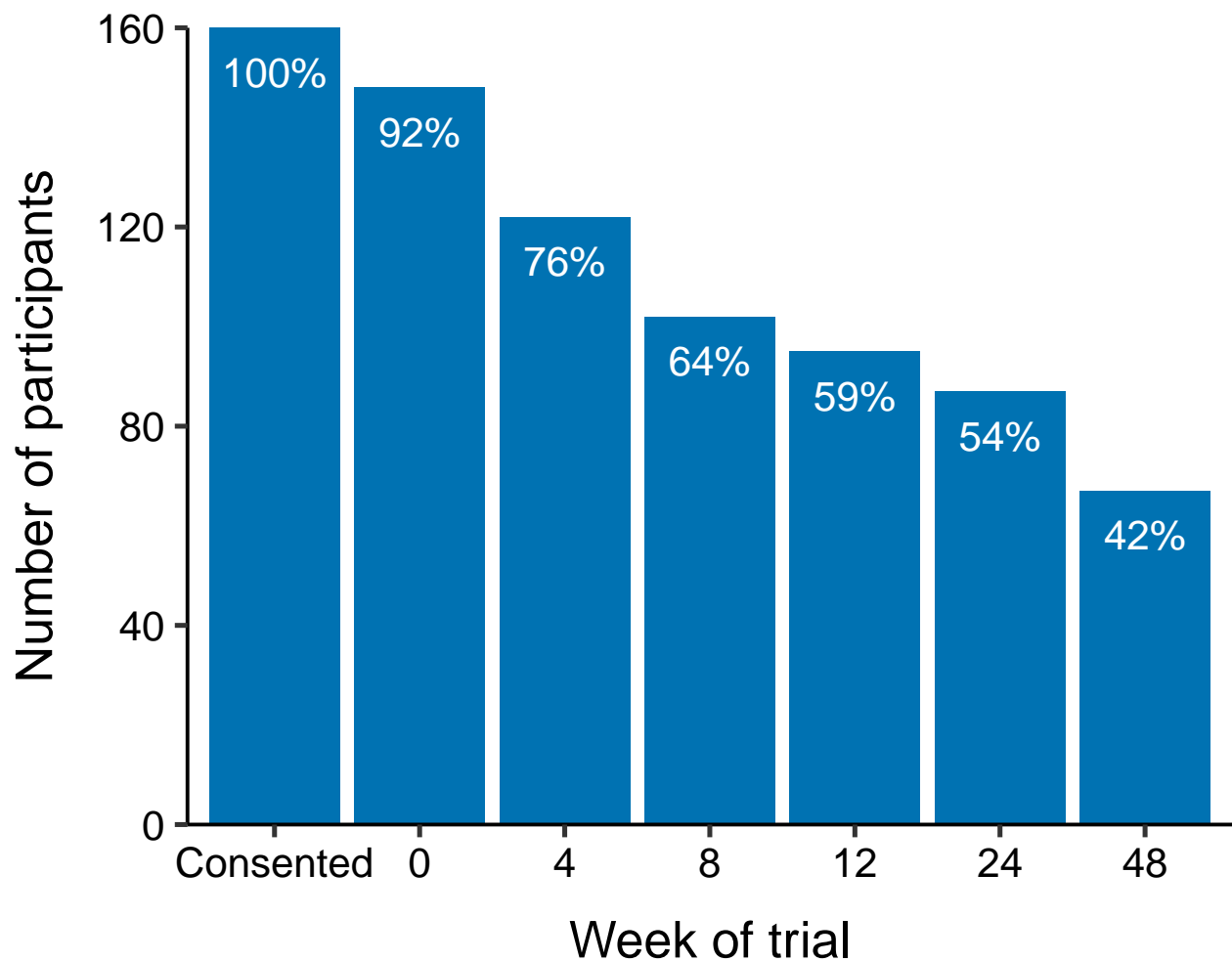
# Colour (dark)
p1 <- ggplot(data = bpi_summary) +
  aes(x = Time_of_loss,

```

```

    y = lag_cumulative) +
  geom_bar(stat = 'identity',
    fill = '#0072B2') +
  geom_text(aes(label =
    str_glue('{round(100 * (lag_cumulative / nrow(bpi_time)))}%'),
    colour = '#FFFFFF',
    size = 7.5,
    vjust = 2) +
  scale_x_discrete(labels = c('Consented', '0', '4', '8', '12', '24', '48')) +
  scale_y_continuous(limits = c(0, 160),
    breaks = c(0, 40, 80, 120, 160),
    expand = c(0, 0)) +
  labs(x = 'Week of trial',
    y = 'Number of participants') +
  theme_bw(base_size = 26) +
  theme(legend.position = 'none',
    panel.border = element_blank(),
    panel.grid = element_blank(),
    axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)),
    axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0)),
    axis.text = element_text(colour = '#000000'),
    axis.line = element_line(size = 0.9)); p1

```

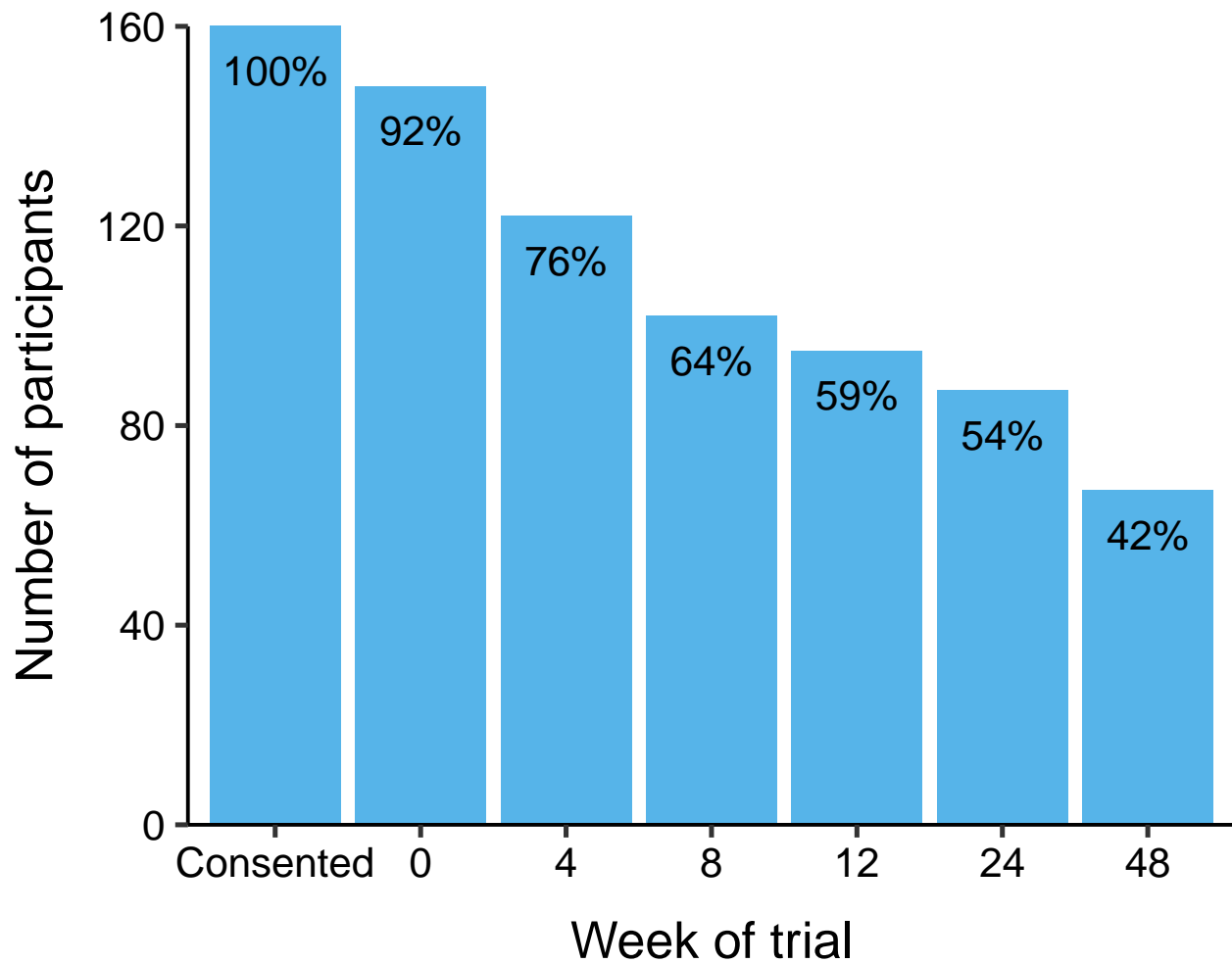


```

ggsave(filename = 'figures/figure-1_colourA.pdf',
        plot = p1,
        height = 8,
        width = 10,
        units = 'in')

# Colour (light)
p2 <- ggplot(data = bpi_summary) +
  aes(x = Time_of_loss,
      y = lag_cumulative) +
  geom_bar(stat = 'identity',
          fill = '#56B4E9') +
  geom_text(aes(label =
                str_glue('{round(100 * (lag_cumulative / nrow(bpi_time)))}%'),
                colour = '#000000',
                size = 7.5,
                vjust = 2) +
  scale_x_discrete(labels = c('Consented', '0', '4', '8', '12', '24', '48')) +
  scale_y_continuous(limits = c(0, 160),
                    breaks = c(0, 40, 80, 120, 160),
                    expand = c(0, 0)) +
  labs(x = 'Week of trial',
       y = 'Number of participants') +
  theme_bw(base_size = 26) +
  theme(legend.position = 'none',
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)),
        axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0)),
        axis.text = element_text(colour = '#000000'),
        axis.line = element_line(size = 0.9)); p2

```



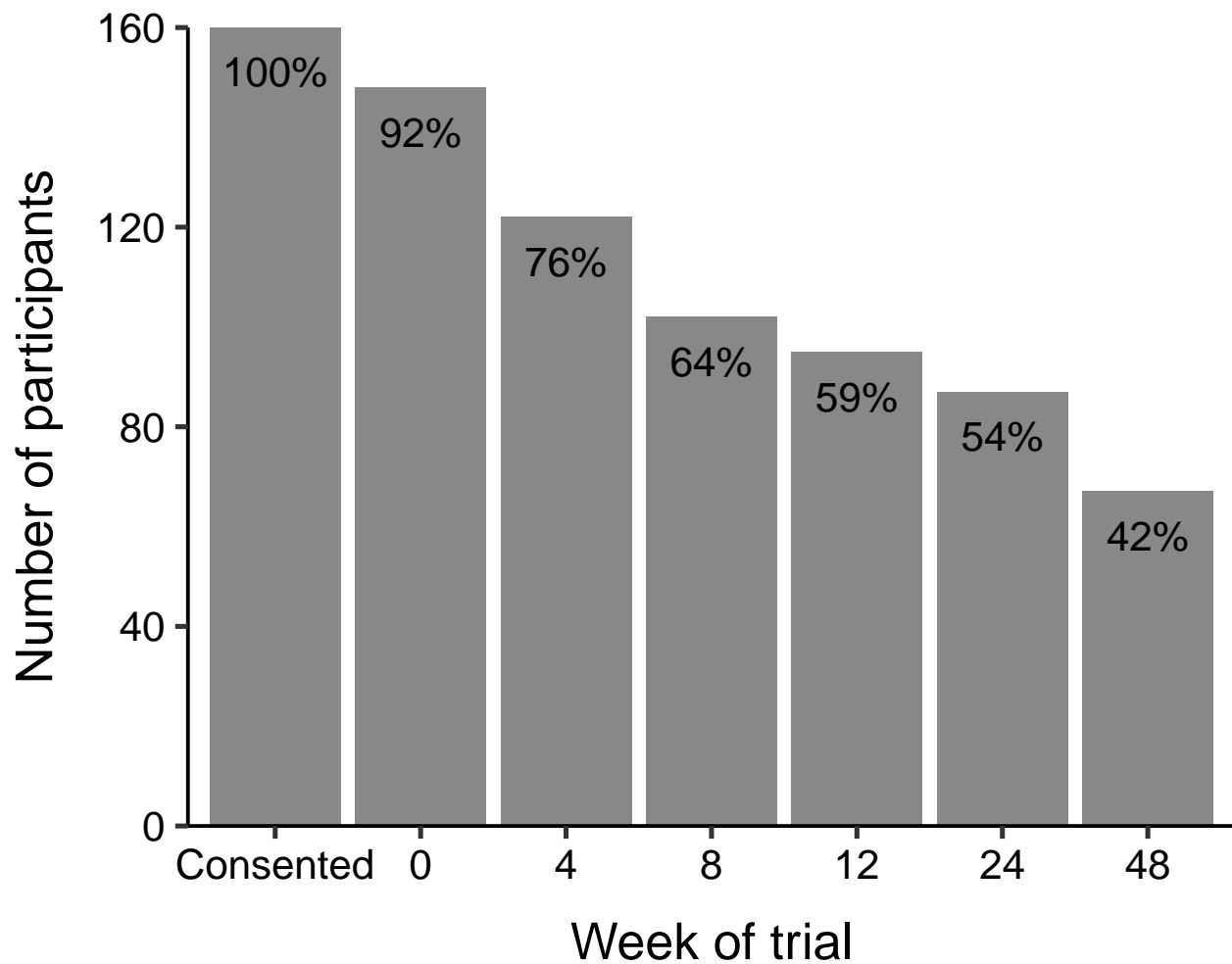
```
ggsave(filename = 'figures/figure-1_colourB.pdf',
        plot = p2,
        height = 8,
        width = 10,
        units = 'in')

# Greyscale
p3 <- ggplot(data = bpi_summary) +
  aes(x = Time_of_loss,
      y = lag_cumulative) +
  geom_bar(stat = 'identity',
          fill = '#888888') +
  geom_text(aes(label =
                str_glue('{round(100 * (lag_cumulative / nrow(bpi_time)))}%'),
                colour = '#000000',
                size = 7.5,
                vjust = 2) +
  scale_x_discrete(labels = c('Consented', '0', '4', '8', '12', '24', '48')) +
  scale_y_continuous(limits = c(0, 160),
                    breaks = c(0, 40, 80, 120, 160),
                    expand = c(0, 0)) +
  labs(x = 'Week of trial',
       y = 'Number of participants') +
```

```

theme_bw(base_size = 26) +
theme(legend.position = 'none',
      panel.border = element_blank(),
      panel.grid = element_blank(),
      axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)),
      axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0)),
      axis.text = element_text(colour = '#000000'),
      axis.line = element_line(size = 0.9)); p3

```



```

ggsave(filename = 'figures/figure-1_greyscale.pdf',
        plot = p3,
        height = 8,
        width = 10,
        units = 'in')

```

Session information

```

## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)

```

```

## Running under: macOS Mojave 10.14.2
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] bindrcpp_0.2.2  knitr_1.21      skimr_1.0.4     magrittr_1.5
## [5] forcats_0.3.0   stringr_1.3.1   dplyr_0.7.8     purrr_0.3.0
## [9] readr_1.3.1     tidyr_0.8.2     tibble_2.0.1    ggplot2_3.1.0
## [13] tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] tidyselect_0.2.5 xfun_0.4        reshape2_1.4.3  haven_2.0.0
## [5] lattice_0.20-38  colorspace_1.4-0 generics_0.0.2  htmltools_0.3.6
## [9] yaml_2.2.0       rlang_0.3.1     pillar_1.3.1    glue_1.3.0
## [13] withr_2.1.2.9000 modelr_0.1.2     readxl_1.2.0    bindr_0.1.1
## [17] plyr_1.8.4       munsell_0.5.0   gtable_0.2.0    cellranger_1.1.0
## [21] rvest_0.3.2      evaluate_0.12   labeling_0.3     highr_0.7
## [25] broom_0.5.1      Rcpp_1.0.0      scales_1.0.0    backports_1.1.3
## [29] jsonlite_1.6     hms_0.4.2       digest_0.6.18   stringi_1.2.4
## [33] grid_3.5.2       cli_1.0.1       tools_3.5.2     lazyeval_0.2.1
## [37] crayon_1.3.4     pkgconfig_2.0.2 xml2_1.2.0       lubridate_1.7.4
## [41] assertthat_0.2.0 rmarkdown_1.11  httr_1.4.0      rstudioapi_0.9.0
## [45] R6_2.3.0         nlme_3.1-137    compiler_3.5.2

```