

Supplement 1

Summary statistics for baseline variables

Peter Kamerman and Tory Madden

Last updated: 07 May 2019

Contents

Import data	1
Clean data	2
Demographic data	2
Brief Pain Inventory (BPI)	3
Beck's Depression Inventory (BDI)	3
EQ-5D (3L)	3
Self-efficacy Questionnaire 6 (SE6)	4
Analysis	5
Demographic data	5
Continuous variables	5
Factor variables	6
Brief Pain Inventory	7
Continuous variables	7
Beck's Depression Index	8
Continuous variables	8
EQ5D (3L)	9
Continuous variables	9
Self-efficacy Questionnaire 6	10
Continuous variables	10
Session information	11

Tabular summary statistics of baseline variables (time = 0 weeks) for the whole cohort, and stratified by sex and study site.

Import data

```
# Demographic data
demo <- read_rds('data-cleaned/demographics.rds')

# BDI
bdi <- read_rds('data-cleaned/bdi.rds')

# BPI
bpi <- read_rds('data-cleaned/bpi.rds')

# EQ5D
eq5d <- read_rds('data-cleaned/eq5d.rds')

# SE6
se6 <- read_rds('data-cleaned/se6.rds')
```

Clean data

Demographic data

```
demo %<>%
  # Convert sex to factor
  mutate(Sex = factor(Sex)) %>%
  # Rename Years_on_ART
  rename(Years_on_HAART = Years_on_ART) %>%
  # Convert SOS_mnemonic to factor
  mutate(SOS_mnemonic = factor(SOS_mnemonic)) %>%
  # Transfer CD4_nadir data to CD4_recent if missing CD4_recent data
  # (i.e. get the most updated CD4 count available)
  mutate(CD4_most_recent =
    ifelse(is.na(CD4_recent),
      yes = CD4_nadir,
      no = CD4_recent)) %>%
  # Categorise years of schooling into 7 years or less, 8-12 years,
  # and more than 12 years of education, factorize, and then order
  mutate(Education = case_when(
    Years_education <= 7 ~ '0-7 years',
    Years_education > 7 & Years_education <= 12 ~ '8-12 years',
    Years_education > 12 ~ 'More than 12 years'),
    Education = factor(Education,
      levels = c('0-7 years',
        '8-12 years',
        'More than 12 years'),
      ordered = TRUE)) %>%
  # Recode HAART and order
  mutate(HAART = case_when(
    HAART == 'first-line' ~ 'first-line HAART',
    HAART == 'second-line' ~ 'second-line HAART',
    HAART == 'monitoring' ~ 'no HAART'),
    HAART = factor(HAART,
      levels = c('no HAART',
        'first-line HAART',
        'second-line HAART'),
      ordered = TRUE)) %>%
  # Recode and order occupation categories
  mutate(Employment = str_replace_all(Occupation,
    pattern = '^unemployed - .+',
    replacement = 'unemployed'),
    Employment = factor(Employment,
      levels = c('employed', 'unemployed',
        'student/volunteer',
        'unable to work - disability grant')) %>%
  # Select required columns
  select(ID, Study_site, Sex, Age_years, Years_on_HAART,
    CD4_most_recent, HAART, Education, Employment,
    SOS_mnemonic)

# Make a site/sex filter
```

```
sorter <- demo %>%
  select(ID, Study_site, Sex)
```

Brief Pain Inventory (BPI)

```
bpi %<>%
  # Capitalize IDs
  mutate(ID = stringr::str_to_upper(ID)) %>%
  # Select baseline values
  select(ID,
    ends_with('BL')) %>%
  # Select columns
  select(ID, 3:6, 9:15) %>%
  # Calculate Pain Severity Score (PSS) at baseline
  mutate(PSS = rowMeans(.[2:5], na.rm = TRUE),
    # Treat PSS as a discrete scale
    PSS = round(PSS)) %>%
  # Calculate Pain Interference Index (PIS) at baseline
  mutate(PIS = rowMeans(.[6:12], na.rm = TRUE),
    # Treat PIS as a discrete scale
    PIS = round(PIS)) %>%
  #remove unwanted columns
  select(ID, PSS, PIS) %>%
  left_join(sorter)
```

Beck's Depression Inventory (BDI)

```
bdi %<>%
  # Capitalize IDs
  mutate(ID = stringr::str_to_upper(ID)) %>%
  # Make a total score column
  mutate_at(2:ncol(bdi),
    as.numeric) %>%
  mutate(BDI = rowSums(.[2:ncol(bdi)], na.rm = TRUE),
    # Treat BDI as a discrete scale
    BDI = round(BDI)) %>%
  select(ID, BDI) %>%
  left_join(sorter)
```

EQ-5D (3L)

```
eq5d %<>%
  # Capitalize IDs
  mutate(ID = stringr::str_to_upper(ID))

# Calculate eq5d index score
## Create basic term = 1 for all cases in new column
eq5d$index_core <- 1

## Sum all rows for total index score
eq5d %<>%
  mutate(index_sum = rowSums(.[2:6], na.rm = TRUE))

# Create constant term to subtract for domain scores > 1 (i.e. sum > 5)
```

```

eq5d %<>%
  mutate(index_constant = ifelse(index_sum > 5,
                                yes = 0.081,
                                no = 0))

## Create variable for subtraction for each domain
eq5d %<>%
  mutate(Mobility_index = ifelse(Mobility.BL == 2,
                                yes = 0.069,
                                no = ifelse(Mobility.BL == 3,
                                              yes = 0.314,
                                              no = 0))) %>%
  mutate(Self_care_index = ifelse(Self_care.BL == 2,
                                yes = 0.104,
                                no = ifelse(Self_care.BL == 3,
                                              yes = 0.214,
                                              no = 0))) %>%
  mutate(Usual_activities_index = ifelse(Usual_activities.BL == 2,
                                         yes = 0.036,
                                         no = ifelse(Usual_activities.BL == 3,
                                                       yes = 0.094,
                                                       no = 0))) %>%
  mutate(Pain_index = ifelse(Pain.BL == 2,
                             yes = 0.123,
                             no = ifelse(Pain.BL == 3,
                                           yes = 0.386,
                                           no = 0))) %>%
  mutate(Anxiety_depression_index = ifelse(Anxiety_and_depression.BL == 2,
                                           yes = 0.071,
                                           no = ifelse(Anxiety_and_depression.BL == 3,
                                                         yes = 0.236,
                                                         no = 0)))

## Compute the index score using:
## index = index_core - constant_index - Mobility_index...
eq5d %<>%
  mutate(EQ5D_index = index_core - index_constant - Mobility_index
         - Self_care_index - Usual_activities_index - Pain_index
         - Anxiety_depression_index) %>%
  # Convert State_of_health VAS to double
  mutate(EQ5D_VAS = as.numeric(State_of_health.BL))

# Select columns
eq5d %<>% select(ID,
               EQ5D_index,
               EQ5D_VAS) %>%
  left_join(sorter)

```

Self-efficacy Questionnaire 6 (SE6)

```

se6 %<>%
  # Capitalize IDs
  mutate(ID = stringr::str_to_upper(ID)) %>%
  # Calculate SE6 at baseline
  mutate(SE6 = rowMeans(.[2:7], na.rm = TRUE),
         # Treat SE6 as a discrete scale

```

```
SE6 = round(SE6)) %>%
#remove unwanted columns
select(ID,
       SE6) %>%
left_join(sorter)
```

Analysis

Demographic data

Continuous variables

```
demo %>%
  select_if(is.numeric) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrrrrrrr')
```

Table 1: Whole cohort

type	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	Age_years	0	160	160	35.23	5.65	35	32	38	18	58
numeric	CD4_most_recent	8	152	160	406.45	249.51	376	224.75	547	3	1189
numeric	Years_on_HAART	78	82	160	3.56	2.83	3	1	5.06	0.25	13

```
demo %>%
  select(Study_site, Age_years, Years_on_HAART, CD4_most_recent) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By study site',
        align = 'lllrrrrrrrrrr')
```

Table 2: By study site

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	R1	Age_years	0	47	47	35.28	2.99	36	33	38	28	40
numeric	R1	CD4_most_recent	0	47	47	415.43	195.84	397	277.5	538	114	1180
numeric	R1	Years_on_HAART	29	18	47	3.92	2.05	4.17	2.31	5.46	0.67	8.3
numeric	R2	Age_years	0	49	49	32.9	4.63	35	30	36	18	40
numeric	R2	CD4_most_recent	2	47	49	450.53	241.69	407	268.5	562.5	36	1120
numeric	R2	Years_on_HAART	2	47	49	3.97	3.25	3	1	6	0.25	13
numeric	U1	Age_years	0	47	47	39.34	6.27	38	35	43.5	27	58
numeric	U1	CD4_most_recent	6	41	47	302.73	284.63	206	113	368	3	1189
numeric	U1	Years_on_HAART	47	0	47	NaN	NA	NA	NA	NA	Inf	-Inf
numeric	U2	Age_years	0	17	17	30.41	4.77	30	26	34	23	37
numeric	U2	CD4_most_recent	0	17	17	509.94	248.8	471	414	648	119	1097
numeric	U2	Years_on_HAART	0	17	17	2.07	1.69	1	0.67	3.42	0.33	5.25

```
demo %>%
  select(Sex, Age_years, Years_on_HAART, CD4_most_recent) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
```

```
kable(., caption = 'By sex',
      align = 'lllrrrrrrrrr')
```

Table 3: By sex

type	Sex	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	female	Age_years	0	97	97	34.23	5.97	35	30	37	18	58
numeric	female	CD4_most_recent	5	92	97	433.73	273.51	410.5	244.75	570.5	3	1189
numeric	female	Years_on_HAART	33	64	97	3.46	3.03	3	0.96	5	0.25	13
numeric	male	Age_years	0	63	63	36.76	4.76	36	33.5	39	27	50
numeric	male	CD4_most_recent	3	60	63	364.63	202.5	335	210.5	491.75	30	1180
numeric	male	Years_on_HAART	45	18	63	3.92	2.05	4.17	2.31	5.46	0.67	8.3

Factor variables

```
demo %>%
  select_if(is.factor) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrr')
```

Table 4: Whole cohort

type	variable	missing	complete	n	n_unique	top_counts
factor	Education	2	158	160	3	8-1: 112, 0-7: 44, Mor: 2
factor	Employment	3	157	160	4	une: 96, emp: 51, una: 8, stu: 2
factor	HAART	4	156	160	3	fir: 115, sec: 36, no : 5
factor	Sex	0	160	160	2	fem: 97, mal: 63
factor	SOS_mnemonic	47	113	160	2	low: 78, hea: 35

```
demo %>%
  select(Study_site, Education, Employment, HAART, Sex, SOS_mnemonic) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By study site',
        align = 'lllrrrrrr')
```

Table 5: By study site

type	Study_site	variable	missing	complete	n	n_unique	top_counts
factor	R1	Education	0	47	47	2	8-1: 30, 0-7: 17, Mor: 0
factor	R1	Employment	2	45	47	3	emp: 21, une: 20, una: 4, stu: 0
factor	R1	HAART	0	47	47	2	fir: 43, sec: 4, no : 0
factor	R1	Sex	0	47	47	1	mal: 47, fem: 0
factor	R1	SOS_mnemonic	0	47	47	2	hea: 28, low: 19
factor	R2	Education	0	49	49	2	8-1: 30, 0-7: 19, Mor: 0
factor	R2	Employment	0	49	49	3	une: 40, emp: 8, una: 1, stu: 0
factor	R2	HAART	0	49	49	3	fir: 40, sec: 5, no : 4
factor	R2	Sex	0	49	49	1	fem: 49, mal: 0
factor	R2	SOS_mnemonic	0	49	49	1	low: 49, hea: 0
factor	U1	Education	2	45	47	3	8-1: 41, 0-7: 2, Mor: 2
factor	U1	Employment	1	46	47	4	une: 25, emp: 19, stu: 1, una: 1
factor	U1	HAART	4	43	47	3	sec: 24, fir: 18, no : 1
factor	U1	Sex	0	47	47	2	fem: 31, mal: 16

type	Study_site	variable	missing	complete	n	n_unique	top_counts
factor	U1	SOS_mnemonic	47	0	47	0	hea: 0, low: 0
factor	U2	Education	0	17	17	2	8-1: 11, 0-7: 6, Mor: 0
factor	U2	Employment	0	17	17	4	une: 11, emp: 3, una: 2, stu: 1
factor	U2	HAART	0	17	17	2	fir: 14, sec: 3, no : 0
factor	U2	Sex	0	17	17	1	fem: 17, mal: 0
factor	U2	SOS_mnemonic	0	17	17	2	low: 10, hea: 7

```
demo %>%
  select(Sex, Education, Employment, HAART, Sex, SOS_mnemonic) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By sex',
        align = 'lllrrrrr')
```

Table 6: By sex

type	Sex	variable	missing	complete	n	n_unique	top_counts
factor	female	Education	1	96	97	3	8-1: 68, 0-7: 26, Mor: 2
factor	female	Employment	1	96	97	4	une: 65, emp: 26, una: 3, stu: 2
factor	female	HAART	3	94	97	3	fir: 65, sec: 24, no : 5
factor	female	SOS_mnemonic	31	66	97	2	low: 59, hea: 7
factor	male	Education	1	62	63	2	8-1: 44, 0-7: 18, Mor: 0
factor	male	Employment	2	61	63	3	une: 31, emp: 25, una: 5, stu: 0
factor	male	HAART	1	62	63	2	fir: 50, sec: 12, no : 0
factor	male	SOS_mnemonic	16	47	63	2	hea: 28, low: 19

Brief Pain Inventory

- PSS: Pain severity
- PIS: Pain interference

Continuous variables

```
bpi %>%
  select_if(is.numeric) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrrrrrrr')
```

Table 7: Whole cohort

type	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	PIS	17	143	160	5.12	2.58	5	3	7	0	10
numeric	PSS	16	144	160	5.03	2.14	5	4	6	0	10

```
bpi %>%
  select(Study_site, PIS, PSS) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
```

```
kable(., caption = 'By study site',
      align = 'l1lrrrrrrrrrr')
```

Table 8: By study site

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	R1	PIS	12	35	47	4.69	3.19	5	1.5	7	0	9
numeric	R1	PSS	12	35	47	5	3.01	5	3	8	0	10
numeric	R2	PIS	1	48	49	4.79	2.25	5	3	6.25	0	10
numeric	R2	PSS	0	49	49	4.61	1.74	5	4	6	2	10
numeric	U1	PIS	4	43	47	5.12	2.31	5	3.5	7	0	9
numeric	U1	PSS	4	43	47	5	1.6	5	4	6	0	8
numeric	U2	PIS	0	17	17	6.94	2.05	7	6	9	3	10
numeric	U2	PSS	0	17	17	6.41	1.84	6	6	7	3	10

```
bpi %>%
  select(Sex, PIS, PSS) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By sex',
        align = 'l1lrrrrrrrrrr')
```

Table 9: By sex

type	Sex	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	female	PIS	5	92	97	5.32	2.39	5	4	7	0	10
numeric	female	PSS	4	93	97	5.05	1.89	5	4	6	0	10
numeric	male	PIS	12	51	63	4.76	2.88	5	2	7	0	9
numeric	male	PSS	12	51	63	5	2.56	5	4	6	0	10

Beck's Depression Index

Continuous variables

```
bdi %>%
  select_if(is.numeric) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrrrrrrr')
```

Table 10: Whole cohort

type	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	BDI	0	160	160	20.07	13.22	18.5	10	29	0	55

```
bdi %>%
  select(Study_site, BDI) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By study site',
        align = 'l1lrrrrrrrrrr')
```

Table 11: By study site

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	R1	BDI	0	47	47	13.47	12.1	13	0	21.5	0	45
numeric	R2	BDI	0	49	49	25.73	11.12	27	17	33	5	49
numeric	U1	BDI	0	47	47	17.51	12.39	16	8	25.5	0	46
numeric	U2	BDI	0	17	17	29.06	13.15	31	22	35	0	55

```

bdi %>%
  select(Sex, BDI) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By sex',
        align = 'lllrrrrrrrrrr')

```

Table 12: By sex

type	Sex	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	female	BDI	0	97	97	24.26	12.65	25	16	33	0	55
numeric	male	BDI	0	63	63	13.62	11.44	13	4	20	0	45

EQ5D (3L)

Continuous variables

```

eq5d %>%
  select_if(is.numeric) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrrrrrrr')

```

Table 13: Whole cohort

type	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	EQ5D_index	18	142	160	0.62	0.19	0.69	0.49	0.76	-0.21	1
numeric	EQ5D_VAS	16	144	160	59.52	21.17	60	50	76.25	0	100

```

eq5d %>%
  select(Study_site, EQ5D_index, EQ5D_VAS) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By study site',
        align = 'lllrrrrrrrrrr')

```

Table 14: By study site

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	R1	EQ5D_index	12	35	47	0.67	0.21	0.73	0.66	0.78	-0.21	1
numeric	R1	EQ5D_VAS	12	35	47	62.46	23.1	69	50	80	10	100
numeric	R2	EQ5D_index	0	49	49	0.66	0.18	0.73	0.62	0.8	-0.05	0.8
numeric	R2	EQ5D_VAS	0	49	49	59.9	16.6	60	50	70	20	90

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	U1	EQ5D_index	6	41	47	0.56	0.17	0.5	0.43	0.73	0.36	1
numeric	U1	EQ5D_VAS	4	43	47	59.77	20.64	60	55	75	0	90
numeric	U2	EQ5D_index	0	17	17	0.53	0.22	0.52	0.46	0.66	0.008	0.85
numeric	U2	EQ5D_VAS	0	17	17	51.76	29.04	60	40	75	0	80

```
eq5d %>%
  select(Sex, EQ5D_index, EQ5D_VAS) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By sex',
        align = 'lllrrrrrrrrrr')
```

Table 15: By sex

type	Sex	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	female	EQ5D_index	5	92	97	0.61	0.19	0.66	0.47	0.76	-0.05	1
numeric	female	EQ5D_VAS	4	93	97	58.12	21.02	60	50	75	0	90
numeric	male	EQ5D_index	13	50	63	0.64	0.2	0.73	0.51	0.76	-0.21	1
numeric	male	EQ5D_VAS	12	51	63	62.08	21.42	70	50	80	10	100

Self-efficacy Questionnaire 6

Continuous variables

```
se6 %>%
  select_if(is.numeric) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'Whole cohort',
        align = 'llrrrrrrrrrr')
```

Table 16: Whole cohort

type	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	SE6	18	142	160	6.93	2.25	7.5	5	9	1	10

```
se6 %>%
  select(Study_site, SE6) %>%
  group_by(Study_site) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By study site',
        align = 'lllrrrrrrrrrr')
```

Table 17: By study site

type	Study_site	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	R1	SE6	12	35	47	6.4	2.24	7	5	8	1	10
numeric	R2	SE6	0	49	49	6.16	2.32	6	5	8	1	10
numeric	U1	SE6	6	41	47	8.54	1.43	9	8	10	4	10
numeric	U2	SE6	0	17	17	6.35	1.8	6	5	8	2	9

```
se6 %>%
  select(Sex, SE6) %>%
  group_by(Sex) %>%
  skim_to_wide(.) %>%
  kable(., caption = 'By sex',
        align = 'lllrrrrrrrrrr')
```

Table 18: By sex

type	Sex	variable	missing	complete	n	mean	sd	median	q25	q75	min	max
numeric	female	SE6	5	92	97	6.9	2.31	8	5	9	1	10
numeric	male	SE6	13	50	63	6.98	2.16	7	6	8.75	1	10

Session information

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## 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_1.22      skimr_1.0.5     magrittr_1.5    forcats_0.4.0
## [5] stringr_1.4.0   dplyr_0.8.0.1   purrr_0.3.2     readr_1.3.1
## [9] tidyr_0.8.3     tibble_2.1.1    ggplot2_3.1.1   tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.1      highr_0.8       cellranger_1.1.0 pillar_1.3.1
## [5] compiler_3.6.0  plyr_1.8.4      tools_3.6.0     digest_0.6.18
## [9] lubridate_1.7.4 jsonlite_1.6     evaluate_0.13    nlme_3.1-139
## [13] gtable_0.3.0    lattice_0.20-38 pkgconfig_2.0.2  rlang_0.3.4
## [17] cli_1.1.0       rstudioapi_0.10 yaml_2.2.0       haven_2.1.0
## [21] xfun_0.6        withr_2.1.2.9000 xml2_1.2.0       httr_1.4.0
## [25] hms_0.4.2       generics_0.0.2  grid_3.6.0      tidyselect_0.2.5
## [29] glue_1.3.1      R6_2.4.0        readxl_1.3.1     rmarkdown_1.12
## [33] modelr_0.1.4    backports_1.1.4 scales_1.0.0     htmltools_0.3.6
## [37] rvest_0.3.3     assertthat_0.2.1 colorspace_1.4-1 stringi_1.4.3
## [41] lazyeval_0.2.2  munsell_0.5.0   broom_0.5.2     crayon_1.3.4
```

Supplement 2

Data completeness

Peter Kamerman

Last updated: 07 May 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

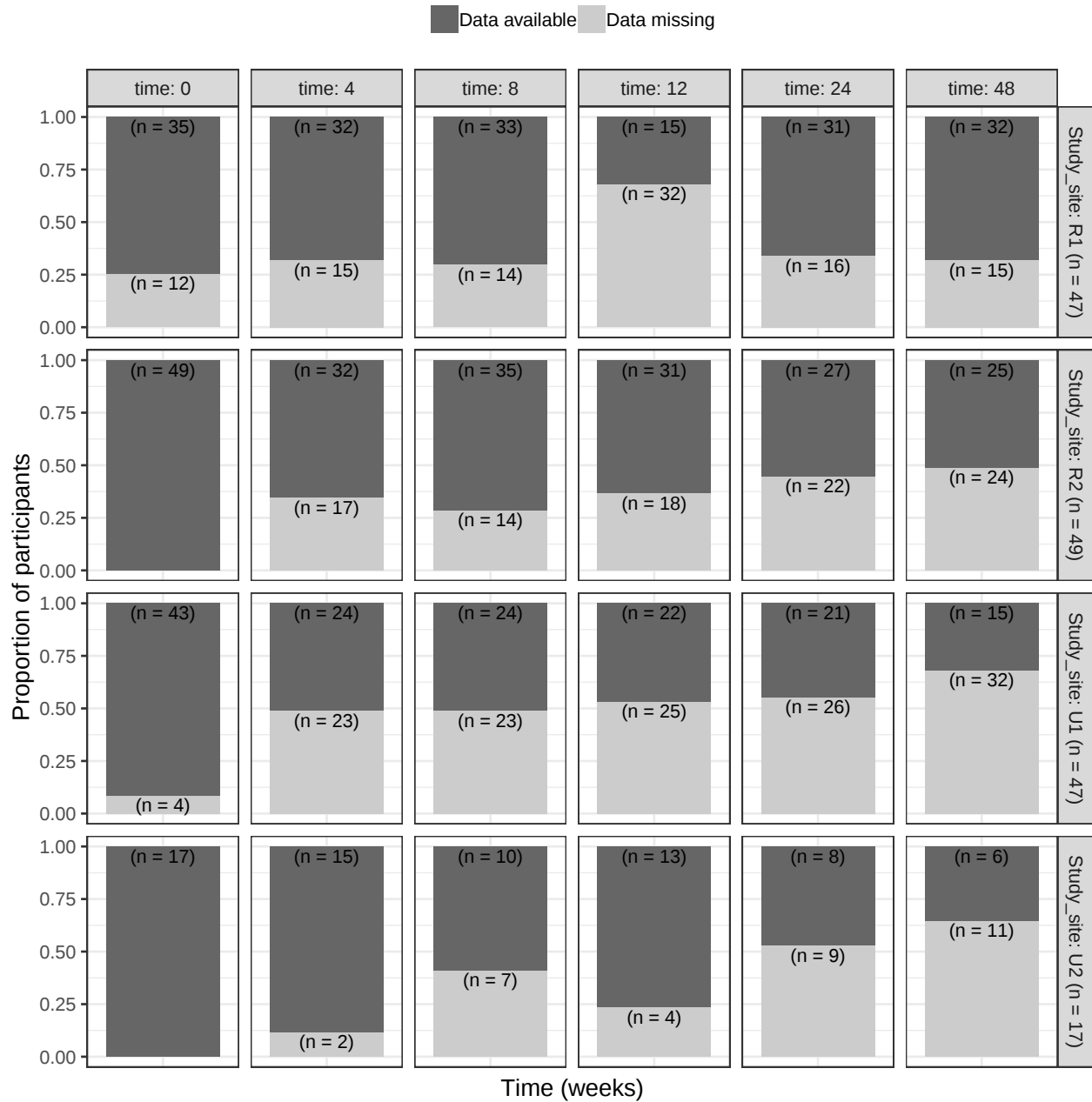
```

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()))))

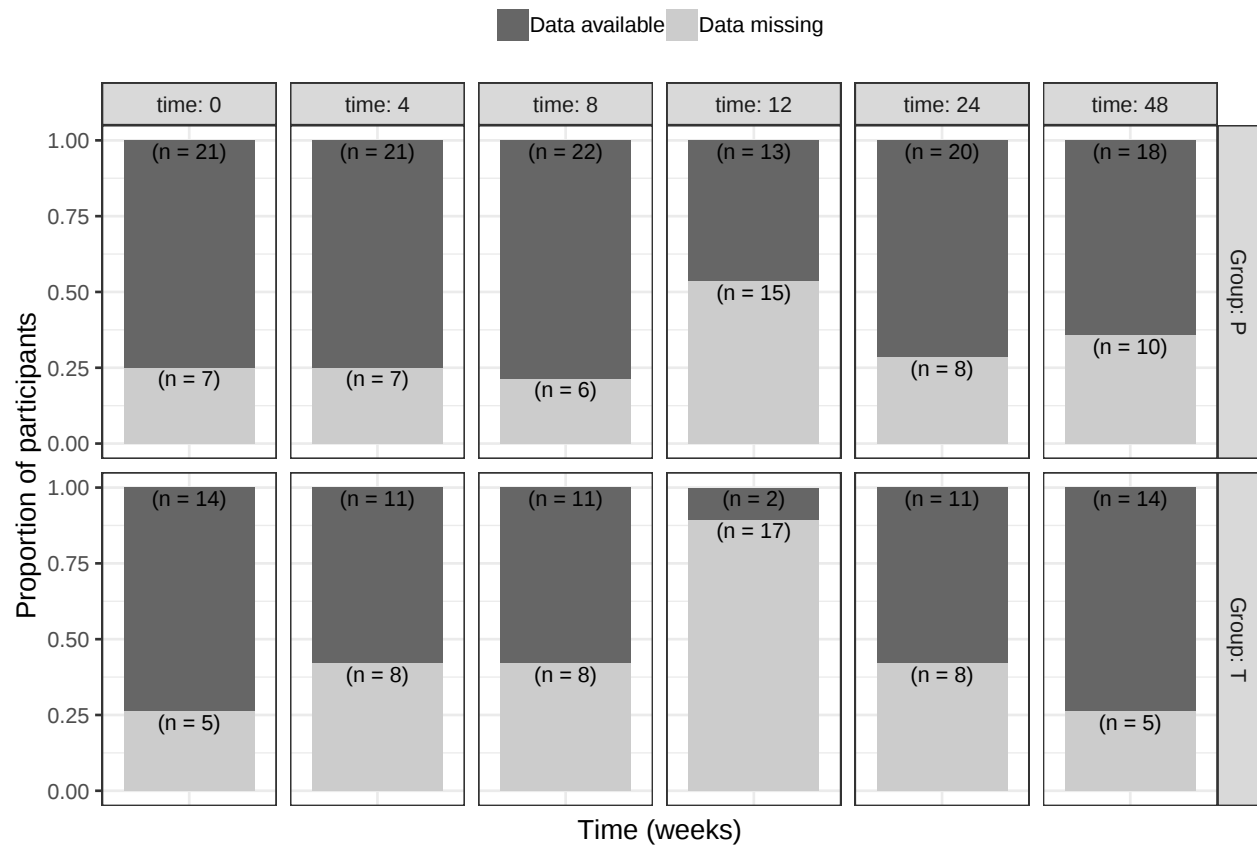
# Print output
walk(.x = complete_group$plot, ~ print(.x))

```

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

Nominal sample size (Group P): 28

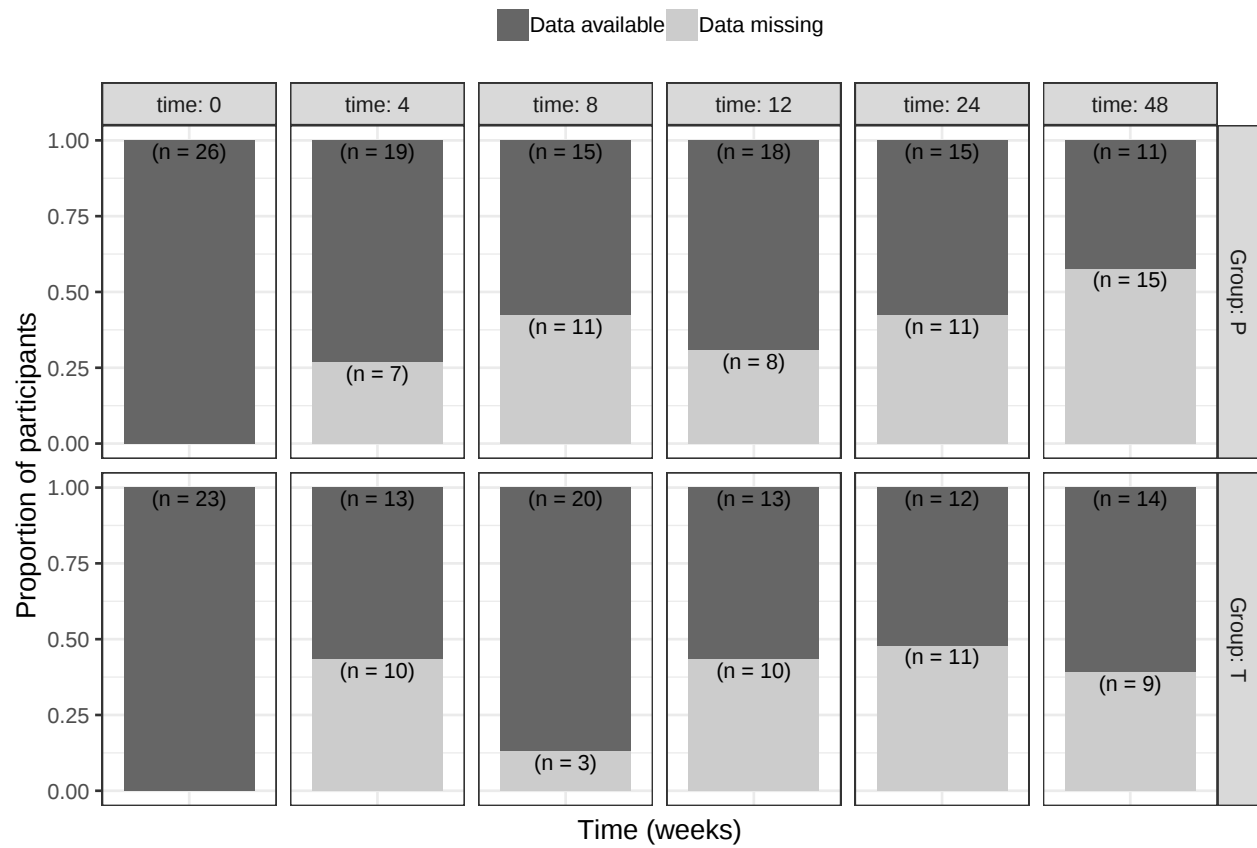
Nominal sample size (Group T): 19



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

Nominal sample size (Group P): 26

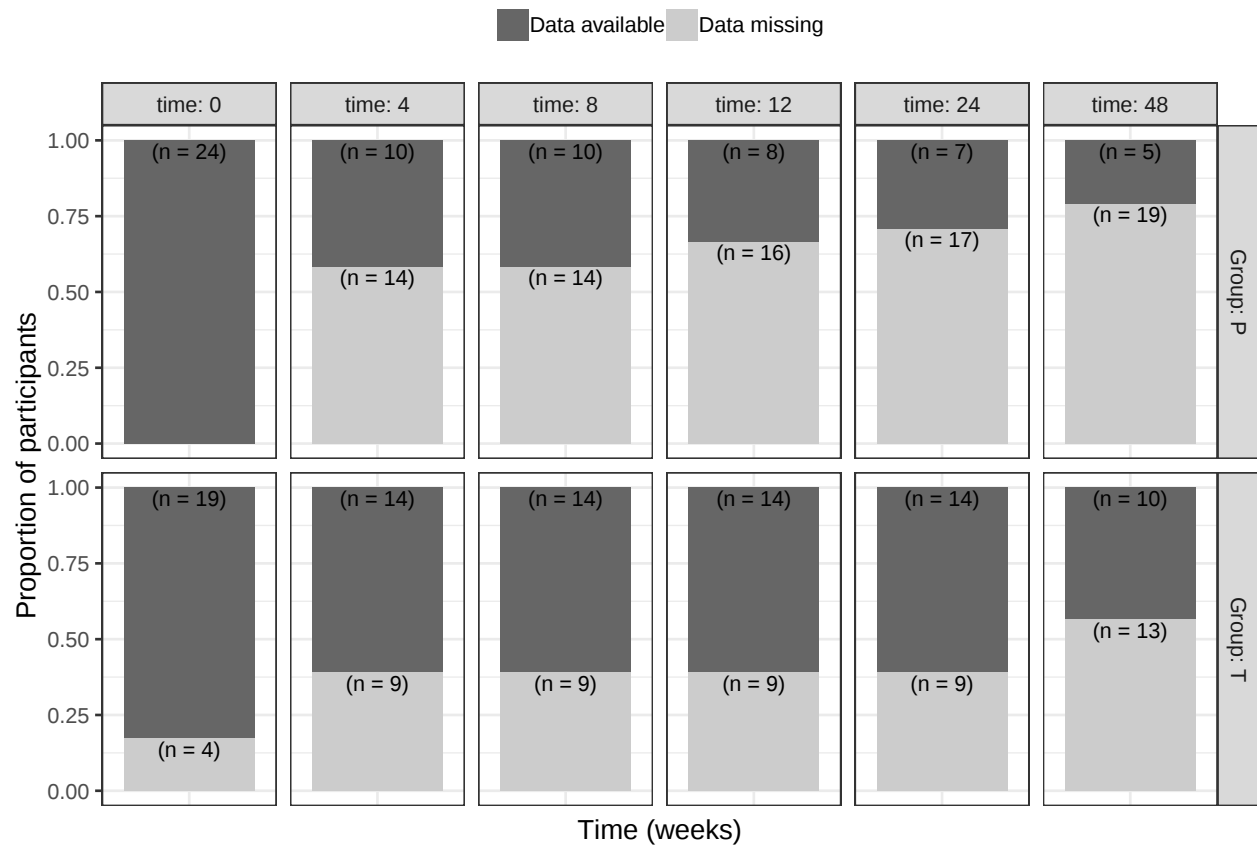
Nominal sample size (Group T): 23



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

Nominal sample size (Group P): 24

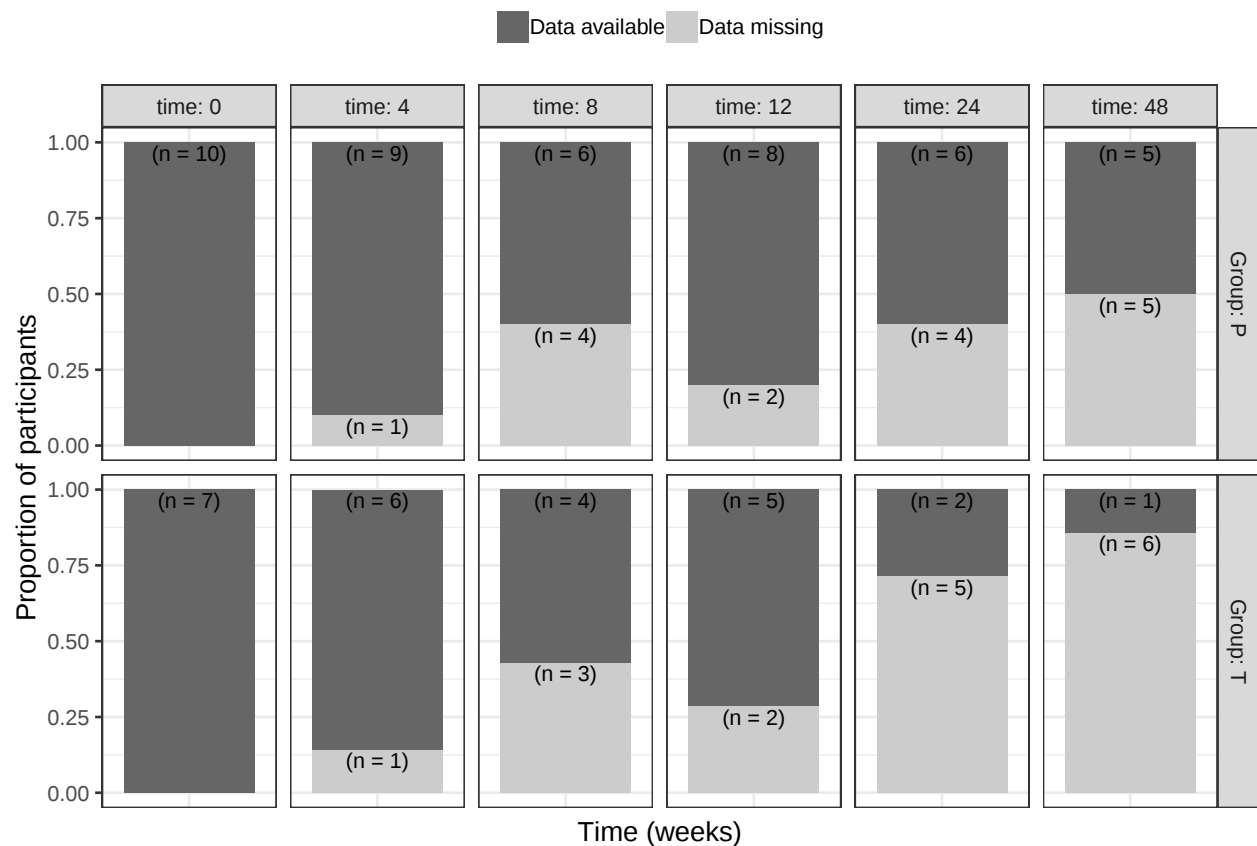
Nominal sample size (Group T): 23



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.

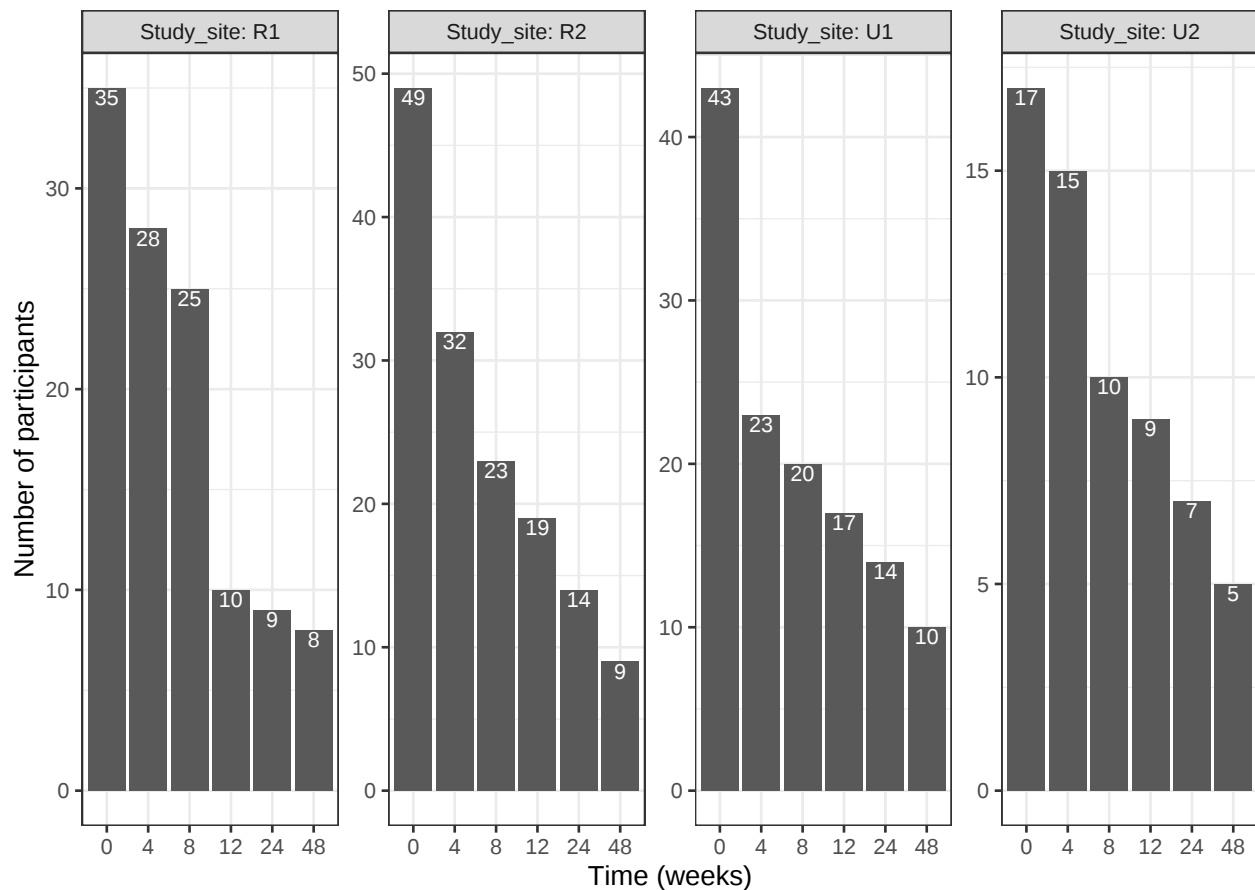
Prepare data

```
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

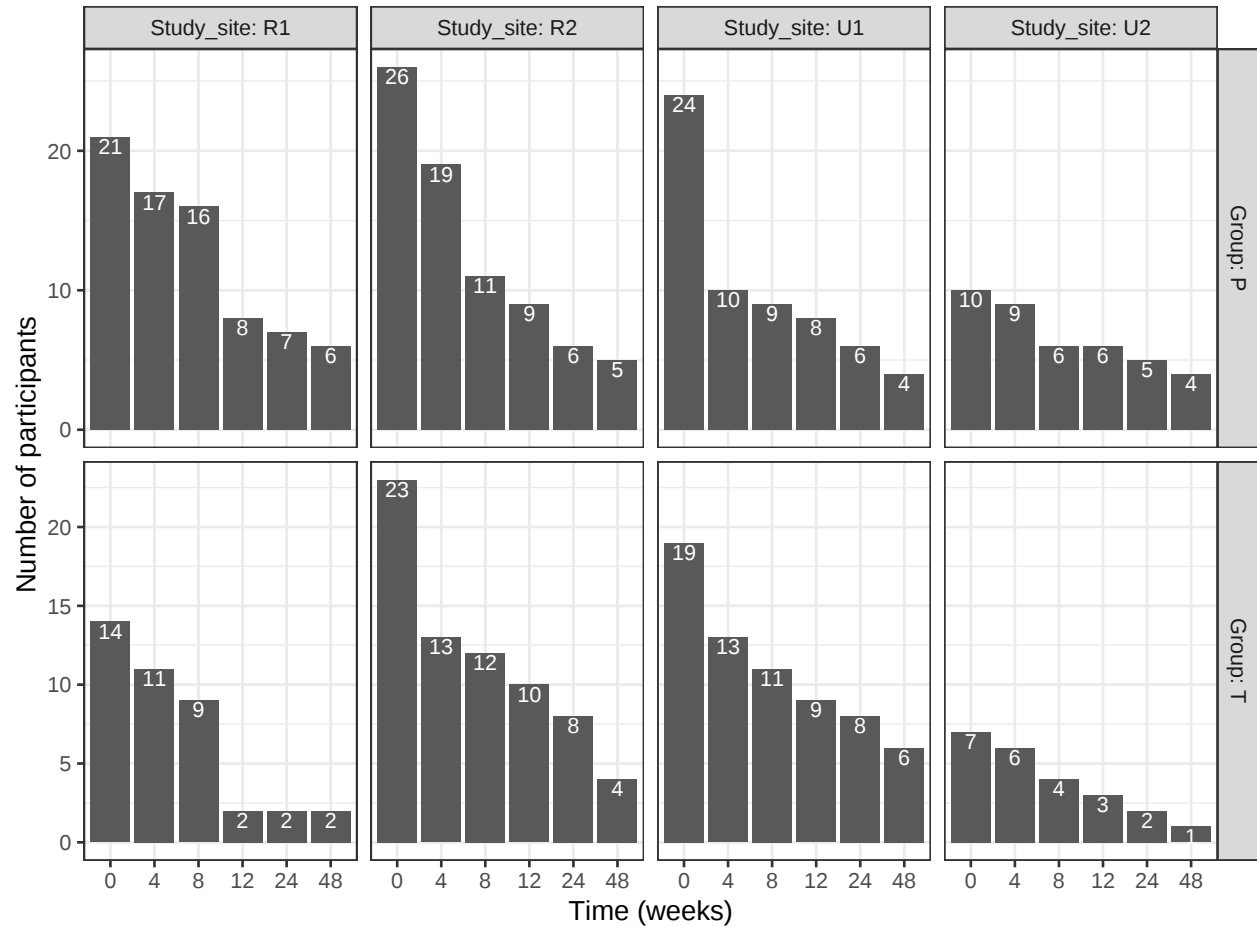
```
bpi_successive %>%
  group_by(Study_site, Group, time, ID) %>%
  summarise(count = sum(!is.na(cumulative_data))) %>%
  ungroup() %>%
  # Filter for counts > 0
  filter(count > 0) %>%
  # Recount by Study_site and time
  group_by(Study_site, Group, time) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  # 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)

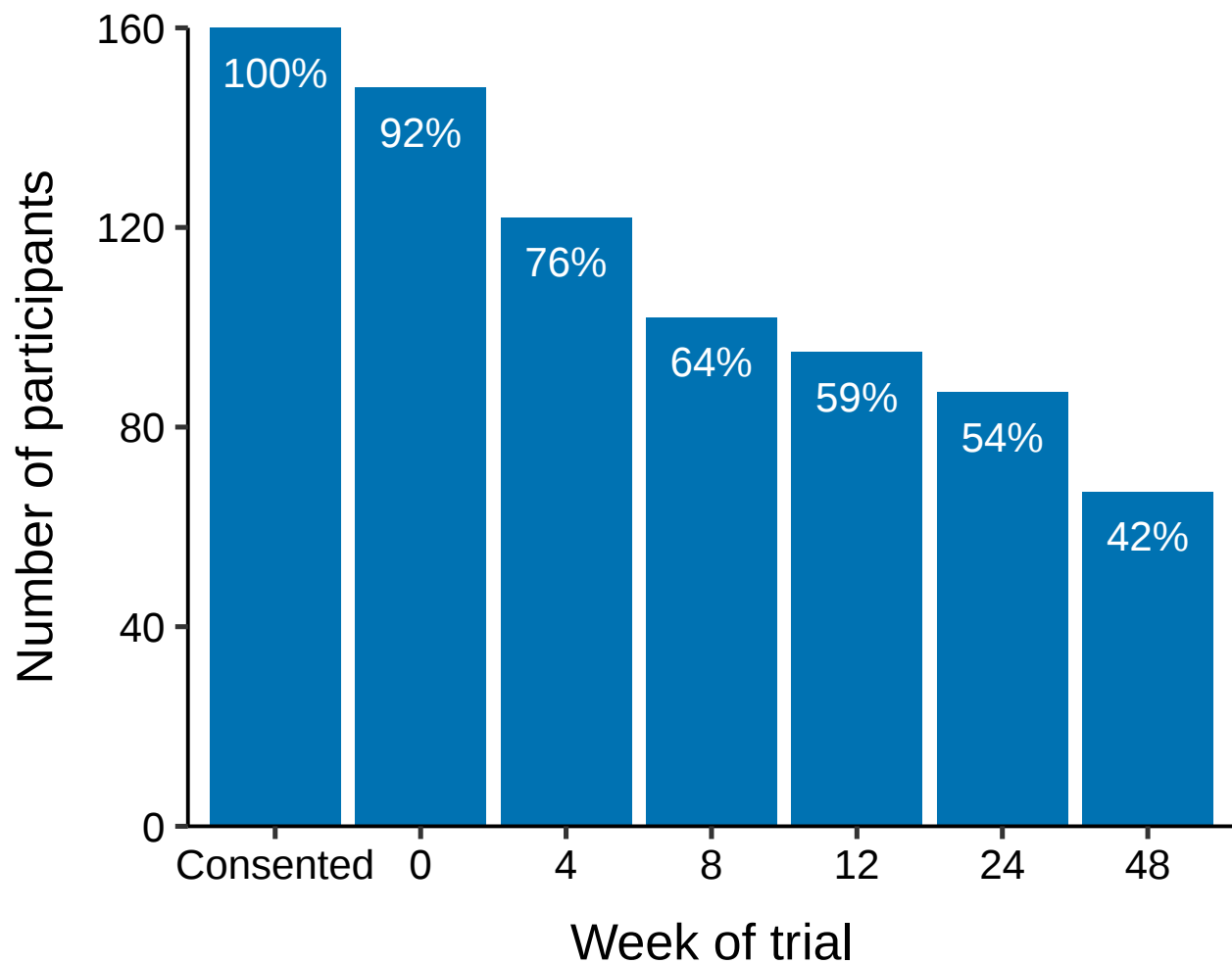
# Colour (dark)
p1 <- ggplot(data = bpi_summary) +
  aes(x = Time_of_loss,

```

```

    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

```

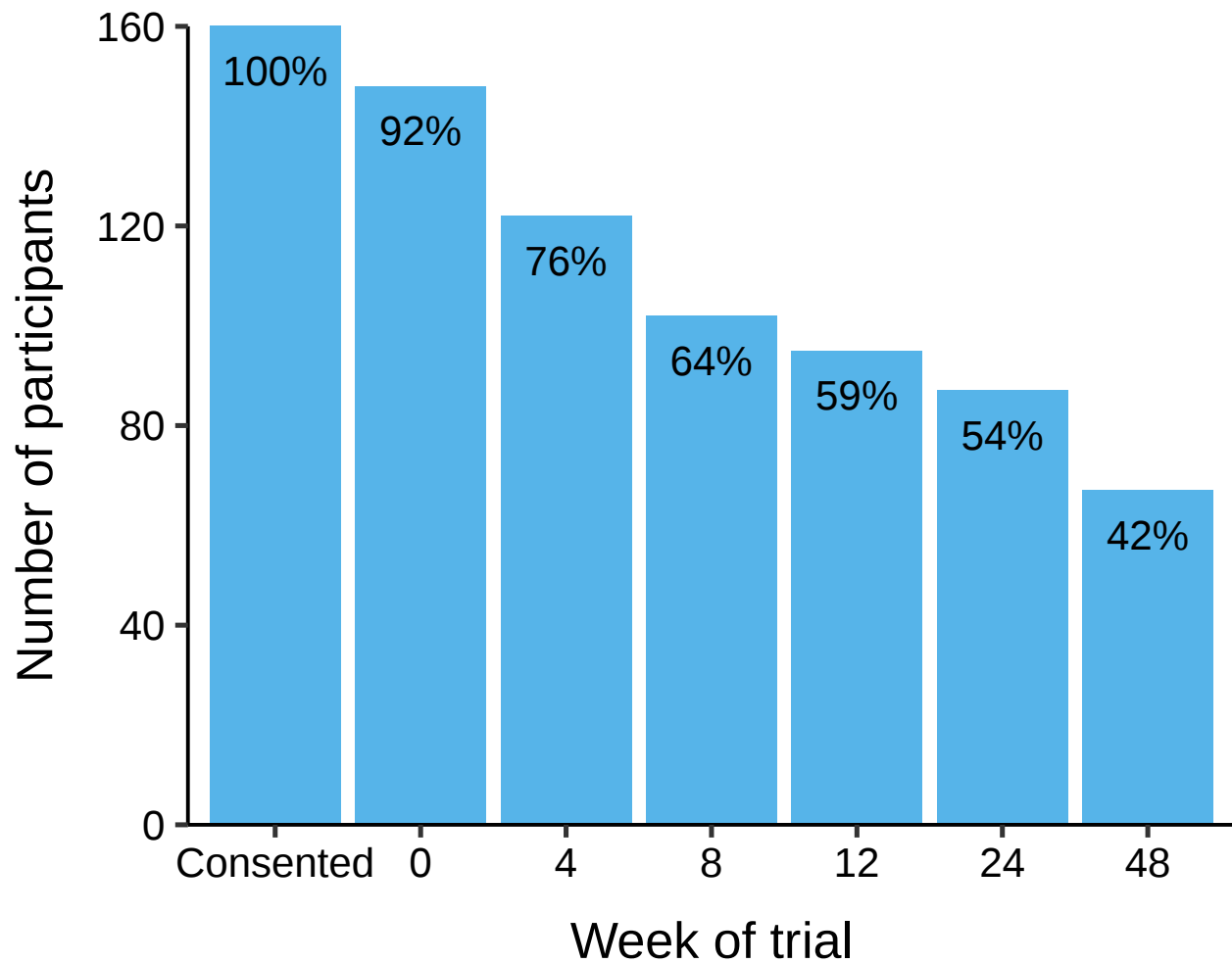


```

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

```



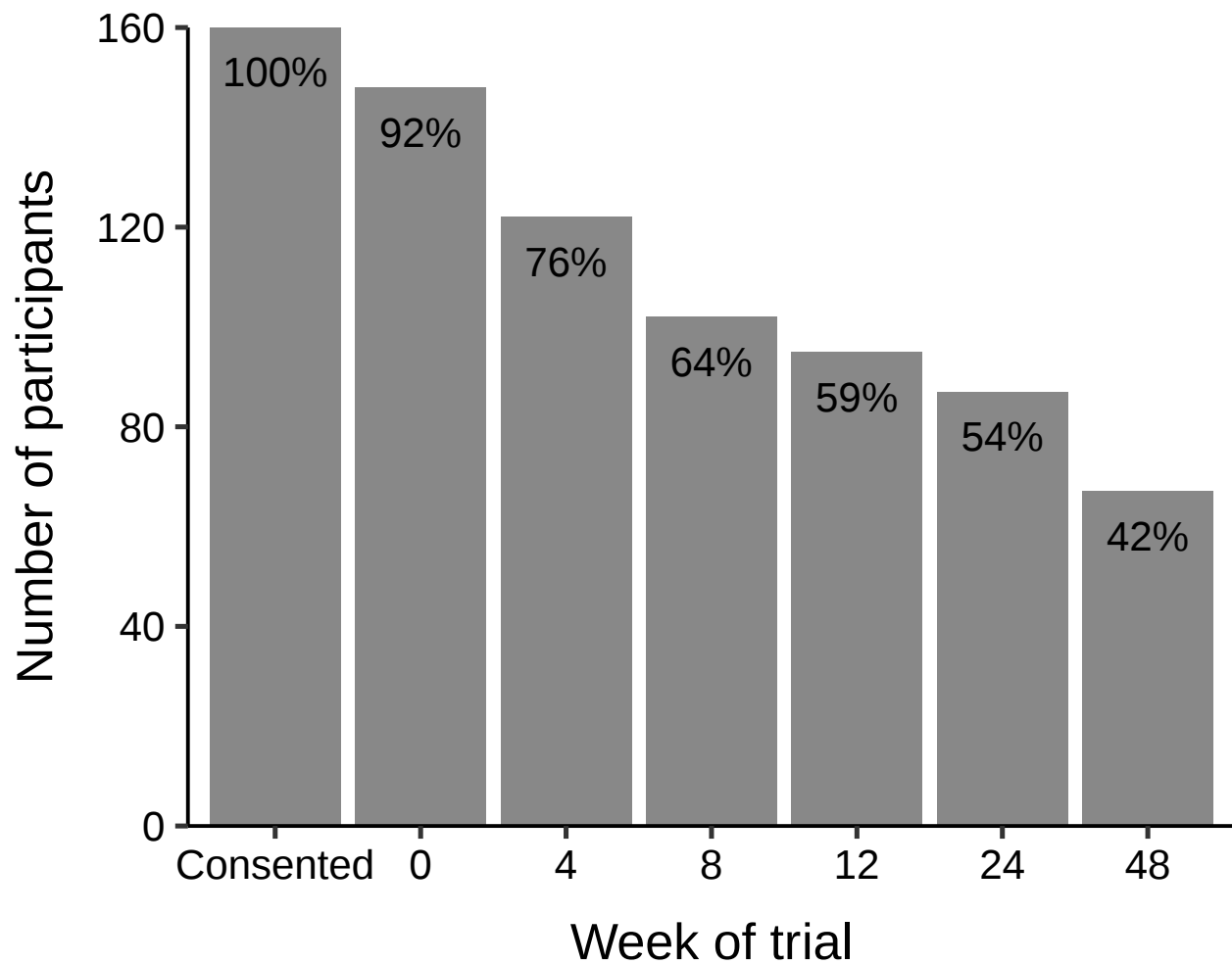
```
ggsave(filename = 'figures/figure-1_colourB.pdf',
        plot = p2,
        height = 8,
        width = 10,
        units = 'in')

# 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 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)

```

```

## Running under: macOS Mojave 10.14.4
##
## 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_1.22      skimr_1.0.5    magrittr_1.5   forcats_0.4.0
## [5] stringr_1.4.0  dplyr_0.8.0.1  purrr_0.3.2    readr_1.3.1
## [9] tidyr_0.8.3     tibble_2.1.1   ggplot2_3.1.1  tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.1      highr_0.8       cellranger_1.1.0 pillar_1.3.1
## [5] compiler_3.6.0  plyr_1.8.4      tools_3.6.0     digest_0.6.18
## [9] lubridate_1.7.4 jsonlite_1.6     evaluate_0.13    nlme_3.1-139
## [13] gtable_0.3.0    lattice_0.20-38 pkgconfig_2.0.2  rlang_0.3.4
## [17] cli_1.1.0       rstudioapi_0.10 yaml_2.2.0       haven_2.1.0
## [21] xfun_0.6         withr_2.1.2.9000 xml2_1.2.0       httr_1.4.0
## [25] hms_0.4.2        generics_0.0.2   grid_3.6.0       tidyselect_0.2.5
## [29] glue_1.3.1       R6_2.4.0         readxl_1.3.1     rmarkdown_1.12
## [33] reshape2_1.4.3  modelr_0.1.4     backports_1.1.4  scales_1.0.0
## [37] htmltools_0.3.6 rvest_0.3.3      assertthat_0.2.1 colorspace_1.4-1
## [41] labeling_0.3     stringi_1.4.3    lazyeval_0.2.2   munsell_0.5.0
## [45] broom_0.5.2      crayon_1.3.4

```

Supplement 3

Exploratory analysis of predictors of study dropout by week 8

Peter Kamerman and Tory Madden

07 May 2019

Contents

Import data	1
Quick look	2
Clean data	3
Employment/income stability	4
Tabulate	4
Null hypothesis significance testing (NHST)	4
Depression and anxiety	6
Tabulate	6
Null hypothesis significance testing (NHST)	6
Study group allocation	8
Tabulate	8
Null hypothesis significance testing (NHST)	9
Sex	11
Tabulate	11
Null hypothesis significance testing (NHST)	11
Summary	12
Session information	17

We assessed four predictors of dropout (employment status, depression, study group allocation, and sex) from the study by week 8. Week 8 is the time point 2 weeks after the completion on the 6 week programme.

Import data

```
# Get data
## BPI
bpi <- read_rds('data-cleaned/bpi.rds') %>%
  select(ID, Pain_present.Wk8)

## Demographics
demo <- read_rds('data-cleaned/demographics.rds') %>%
  select(ID, Study_site, Group, Sex, Occupation)
```

```
## BDI
bdi <- read_rds('data-cleaned/bdi.rds') %>%
  select(ID, ends_with('BL'))
```

Quick look

```
glimpse(bpi)

## Observations: 160
## Variables: 2
## $ ID          <chr> "J1", "J3", "J4", "J5", "J6", "J7", "J9", "J1..."
## $ Pain_present.Wk8 <chr> NA, NA, "Yes", NA, NA, NA, "Yes", "Yes", "Yes..."

glimpse(demo)

## Observations: 160
## Variables: 5
## $ ID          <chr> "J1", "J3", "J4", "J5", "J6", "J7", "J9", "J10", "J..."
## $ Study_site  <chr> "U1", "U1", "U1", "U1", "U1", "U1", "U1", "U1", "U1..."
## $ Group       <chr> "P", "T", "P", "P", "P", "T", "T", "T", "P", "T", "...
## $ Sex         <chr> "female", "female", "female", "female", "female", "...
## $ Occupation <chr> "employed", NA, "employed", "unemployed - looking f..."

glimpse(bdi)

## Observations: 160
## Variables: 22
## $ ID          <chr> "J1", "J3", "J4", "J5", "J6", "J7"..."
## $ Sadness.BL   <int> 1, 3, 0, 0, 2, 1, 0, 1, 3, 0, 3, N..."
## $ Pessimism.BL <int> 2, 3, 0, 1, 0, 0, 0, 0, 0, 1, 2, N..."
## $ Past_failures.BL <int> 2, 3, 0, 2, 1, 0, 0, 0, 2, 0, 0, N..."
## $ Loss_of_pleasure.BL <int> 2, 2, 1, 3, 2, 0, 1, 1, 2, 1, 0, N..."
## $ Guilty_feelings.BL <int> 2, 3, 0, 0, 1, 3, 0, 0, 2, 0, 0, N..."
## $ Punishment_feelings.BL <int> 3, 3, 0, 3, 3, 0, 0, 0, 3, 0, 0, N..."
## $ Self_dislike.BL <int> 1, 2, 0, 1, 1, 0, 0, 0, 2, 0, 0, N..."
## $ Self_critical.BL <int> 3, 3, 0, 3, 0, 0, 3, 0, 3, 0, 3, N..."
## $ Suicidal.BL   <int> 3, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, N..."
## $ Crying.BL     <int> 3, 1, 0, 0, 2, 2, 3, 0, 3, 0, 3, N..."
## $ Agitation.BL  <int> 3, 1, 3, 2, 3, 3, 3, 3, 3, 0, 3, N..."
## $ Loss_of_interest.BL <int> 3, 1, 0, 1, 1, 0, 3, 0, 0, 0, 0, N..."
## $ Indecisiveness.BL <int> 2, 2, 0, 3, 2, 0, 1, 0, 0, 0, 0, N..."
## $ Worthlessness.BL <int> 2, 3, 1, 2, 2, 0, 0, 1, 0, 0, 0, N..."
## $ Loss_of_energy.BL <int> 1, 1, 0, 1, 0, 0, 2, 1, 2, 0, 2, N..."
## $ Sleep.BL      <int> 3, 0, 2, 1, 1, 2, 2, 2, 3, 2, 2, N..."
## $ Irritability.BL <int> 2, 3, 0, 1, 2, 3, 1, 0, 1, 0, 2, N..."
## $ Appetite.BL   <int> 1, 1, 0, 2, 0, 2, 0, 3, 3, 2, 0, N..."
## $ Concentration_difficulty.BL <int> 3, 1, 0, 2, 1, 0, 2, 0, 0, 0, 0, N..."
## $ Fatigue.BL    <int> 2, 0, 0, 1, 0, 2, 1, 0, 3, 1, 2, N..."
## $ Loss_of_interest_in_sex.BL <int> 2, 1, 1, 3, 3, 2, 1, 1, 3, 1, 0, N..."
```

Clean data

```
#####
#
#                               BPI                               #
#                               #
#                               #
#####
# Recode whether there is pain data at week 8 (data completeness)
bpi %<>%
  select(ID, Pain_present.Wk8) %>%
  mutate(coding = ifelse(is.na(Pain_present.Wk8),
    yes = 'Data missing',
    no = 'Data available')) %>%
  select(-Pain_present.Wk8)

#####
#
#                               Demographics                       #
#                               #
#                               #
#####
# Mutate new column to reclassify employment status into income grouping
# Employment status was recoded as stable income (employed or on a grant)
# or unstable income (all other categories, including being a student).
demo %<>%
  mutate(income_stability = case_when(
    Occupation == "employed" |
      Occupation == "unable to work - disability grant" ~ "Stable income",
    Occupation == "student/volunteer" |
      Occupation == "unemployed - looking for work" |
      Occupation == "unemployed - not looking for work" ~ "Unstable or no income"
  )) %>%
  select(ID, Study_site, Group, Sex, income_stability)

# Join with completeness ('bpi') data
demo %<>%
  left_join(bpi)

#####
#
#                               BDI                               #
#                               #
#                               #
#####
# Calculate BDI total score
bdi %<>%
  mutate_at(.vars = 2:22,
    .funs = as.integer) %>%
  mutate(Total.BL = rowSums(.[2:22]))

# Join with demo to get site info
bdi %<>%
  left_join(demo) %>%
  select(ID, Study_site, Total.BL)

# Convert total BDI scores into categories
```

```
## Site U1 used BDI II
## Site U2, R1, and R2 used BDI I
bdi %<>%
  mutate(bdi_category = case_when(
    Study_site == "U1" & Total.BL <= 13 ~ "none-minimal",
    Study_site == "U1" & Total.BL > 13 & Total.BL <= 19 ~ "mild",
    Study_site == "U1" & Total.BL > 19 & Total.BL <= 28 ~ "moderate-severe",
    Study_site == "U1" & Total.BL > 28 ~ "severe",
    Study_site != "U1" & Total.BL <= 9 ~ "none-minimal",
    Study_site != "U1" & Total.BL > 9 & Total.BL <= 18 ~ "mild",
    Study_site != "U1" & Total.BL > 18 & Total.BL <= 29 ~ "moderate-severe",
    Study_site != "U1" & Total.BL > 29 ~ "severe"))

# Convert bdi category into an ordered factor
bdi %<>% mutate(bdi_category = factor(bdi_category,
                                     levels = c("none-minimal",
                                                "mild",
                                                "moderate-severe",
                                                "severe"),
                                     ordered = TRUE))

# Drop Site column
bdi %<>%
  select(-Study_site)

# Join with completeness ('bpi') data
bdi %<>%
  left_join(bpi)
```

Employment/income stability

Tabulate

(no stratification by study site)

```
demo %>% group_by(income_stability) %>%
  summarise(count = n()) %>%
  kable(., caption = 'Access to stable income',
        col.names = c('', 'Count'))
```

Table 1: Access to stable income

	Count
NA	3
Stable income	59
Unstable or no income	98

Null hypothesis significance testing (NHST)

```
# xtabulate the data
employ <- xtabs(~ income_stability + coding,
```

```

data = demo)

# Produce mosaic plot
mosaicplot(employ,
  main = 'Income stability vs data completeness',
  xlab = '',
  ylab = '',
  cex = 1.2,
  color = c('#B2B2B2', '#323232'))

```



```

# Fishers exact test
kable(tidy(fisher.test(employ)),
  caption = 'Association between income stability and data completeness',
  col.names = c('Estimate', 'p-value',
    'Lower 95% CI', 'Upper 95% CI',
    'Method', 'Alternative'),
  digits = 3)

```

Table 2: Association between income stability and data completeness

Estimate	p-value	Lower 95% CI	Upper 95% CI	Method	Alternative
1.439	0.309	0.691	3.067	Fisher's Exact Test for Count Data	two.sided

Depression and anxiety

Tabulate

(no stratification by study site)

```
bdi %>% group_by(bdi_category) %>%
  summarise(count = n()) %>%
  kable(., caption = 'BDI severity category',
        col.names = c('', 'Count'))
```

Table 3: BDI severity category

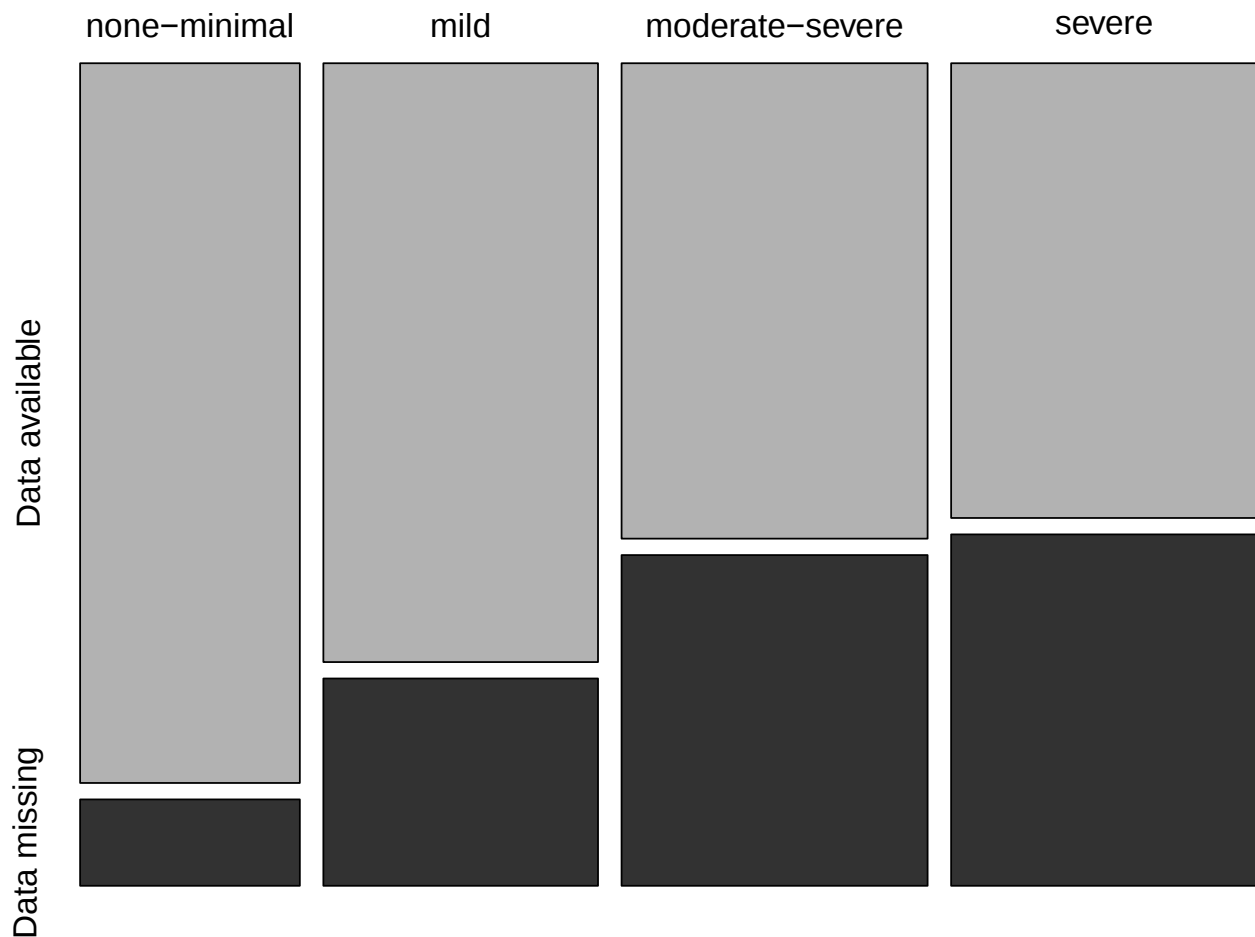
	Count
none-minimal	28
mild	35
moderate-severe	39
severe	39
NA	19

Null hypothesis significance testing (NHST)

```
# xtabulate the data
depression <- xtabs(~ bdi_category + coding,
                  data = bdi)

# Produce mosaic plot
mosaicplot(depression,
            main = 'Depression severity vs data completeness',
            xlab = '',
            ylab = '',
            cex = 1.2,
            color = c('#B2B2B2', '#323232'))
```

Depression severity vs data completeness



```
# Logistic regression on ordered independent variable
```

```
model <- glm(factor(coding) ~ bdi_category,
             data = bdi,
             family = binomial(link = "logit"))
```

```
# Model summary
```

```
Anova(model)
```

```
## Analysis of Deviance Table (Type II tests)
```

```
##
```

```
## Response: factor(coding)
```

```
##          LR Chisq Df Pr(>Chisq)
```

```
## bdi_category    11.4  3    0.00975 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model summary
```

```
summary(model)
```

```
##
```

```
## Call:
```

```
## glm(formula = factor(coding) ~ bdi_category, family = binomial(link = "logit"),
```

```
##      data = bdi)
```

```
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.070  -1.028  -0.771   1.289   2.114
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.95047    0.21406  -4.440 8.99e-06 ***
## bdi_category.L  1.40543    0.47717   2.945 0.00323 **
## bdi_category.Q -0.47716    0.42812  -1.115 0.26504
## bdi_category.C -0.05176    0.37266  -0.139 0.88954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 176.6  on 140  degrees of freedom
## Residual deviance: 165.2  on 137  degrees of freedom
## (19 observations deleted due to missingness)
## AIC: 173.2
##
## Number of Fisher Scoring iterations: 4

# Print odds ratios
ci <- exp(confint(model))[c(-1, -4)]
oddR <- tibble('Item' = names(exp(coef(model))[-1]),
              'Odds ratio' = round(exp(coef(model))[-1], 3),
              'Lower 95% CI' = round(ci[1:3], 3),
              'Upper 95% CI' = round(ci[4:6], 3))

kable(oddR,
      caption = 'Odds ratio of regression coefficients')
```

Table 4: Odds ratio of regression coefficients

Item	Odds ratio	Lower 95% CI	Upper 95% CI
bdi_category.L	4.077	1.718	11.754
bdi_category.Q	0.621	0.252	1.393
bdi_category.C	0.950	0.575	1.968

Study group allocation

Tabulate

(no stratification by study site)

```
demo %>% group_by(Group) %>%
  summarise(count = n()) %>%
  kable(., caption = 'Study group allocation',
        col.names = c('', 'Count'))
```

Table 5: Study group allocation

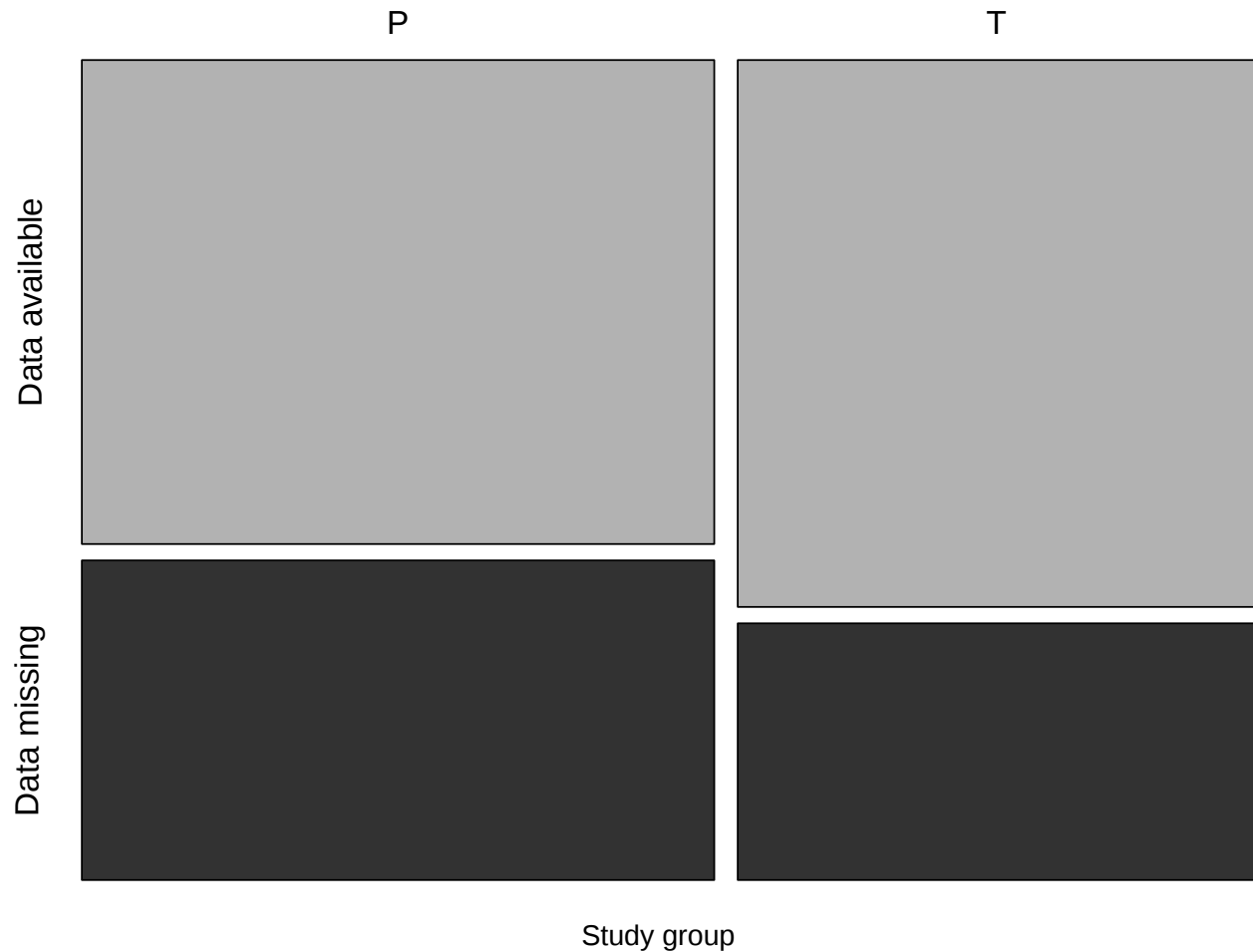
	Count
P	88
T	72

Null hypothesis significance testing (NHST)

```
# xtabulate the data
group <- xtabs(~ Group + coding,
               data = demo)

# Produce mosaic plot
mosaicplot(group,
            main = 'Study group allocation vs data completeness',
            xlab = 'Study group',
            ylab = '',
            cex = 1.2,
            color = c('#B2B2B2', '#323232'))
```

Study group allocation vs data completeness



```
# Fishers exact test
kable(tidy(fisher.test(group)),
      caption = 'Association between study group allocation and data completeness',
      col.names = c('Estimate', 'p-value',
                    'Lower 95% CI', 'Upper 95% CI',
                    'Method', 'Alternative'),
      digits = 3)
```

Table 6: Association between study group allocation and data completeness

Estimate	p-value	Lower 95% CI	Upper 95% CI	Method	Alternative
0.712	0.326	0.35	1.435	Fisher's Exact Test for Count Data	two.sided

Sex

Tabulate

(no stratification by study site)

```
demo %>% group_by(Sex) %>%  
  summarise(count = n()) %>%  
  kable(., caption = 'Sex',  
        col.names = c('', 'Count'))
```

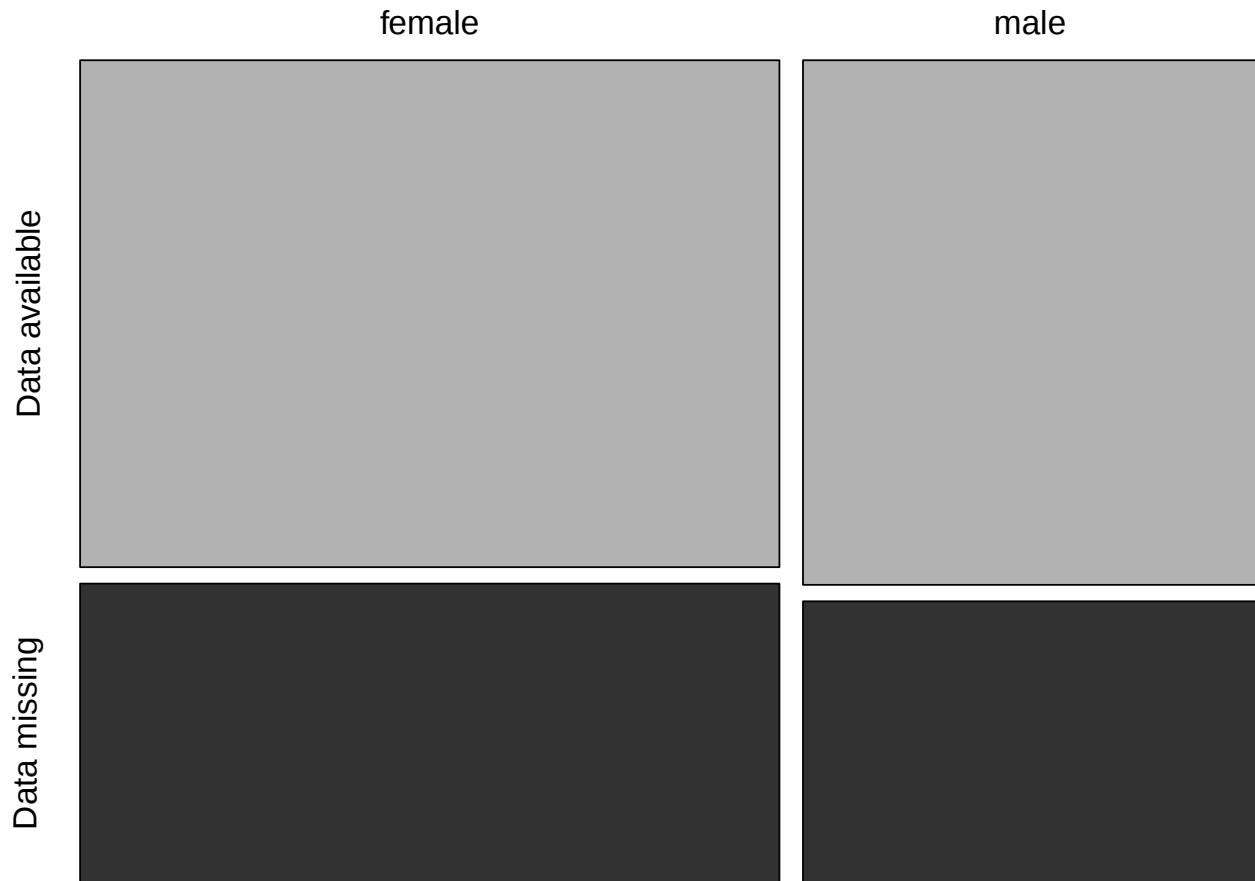
Table 7: Sex

	Count
female	97
male	63

Null hypothesis significance testing (NHST)

```
# xtabulate the data  
sex <- xtabs(~ Sex + coding,  
             data = demo)  
  
# Produce mosaic plot  
mosaicplot(sex,  
            main = 'Sex vs data completeness',  
            xlab = '',  
            ylab = '',  
            cex = 1.2,  
            color = c('#B2B2B2', '#323232'))
```

Sex vs data completeness



```
# Fishers exact test
kable(tidy(fisher.test(sex)),
      caption = 'Association between sex and data completeness',
      col.names = c('Estimate', 'p-value',
                    'Lower 95% CI', 'Upper 95% CI',
                    'Method', 'Alternative'),
      digits = 3)
```

Table 8: Association between sex and data completeness

Estimate	p-value	Lower 95% CI	Upper 95% CI	Method	Alternative
0.91	0.867	0.442	1.85	Fisher's Exact Test for Count Data	two.sided

Summary

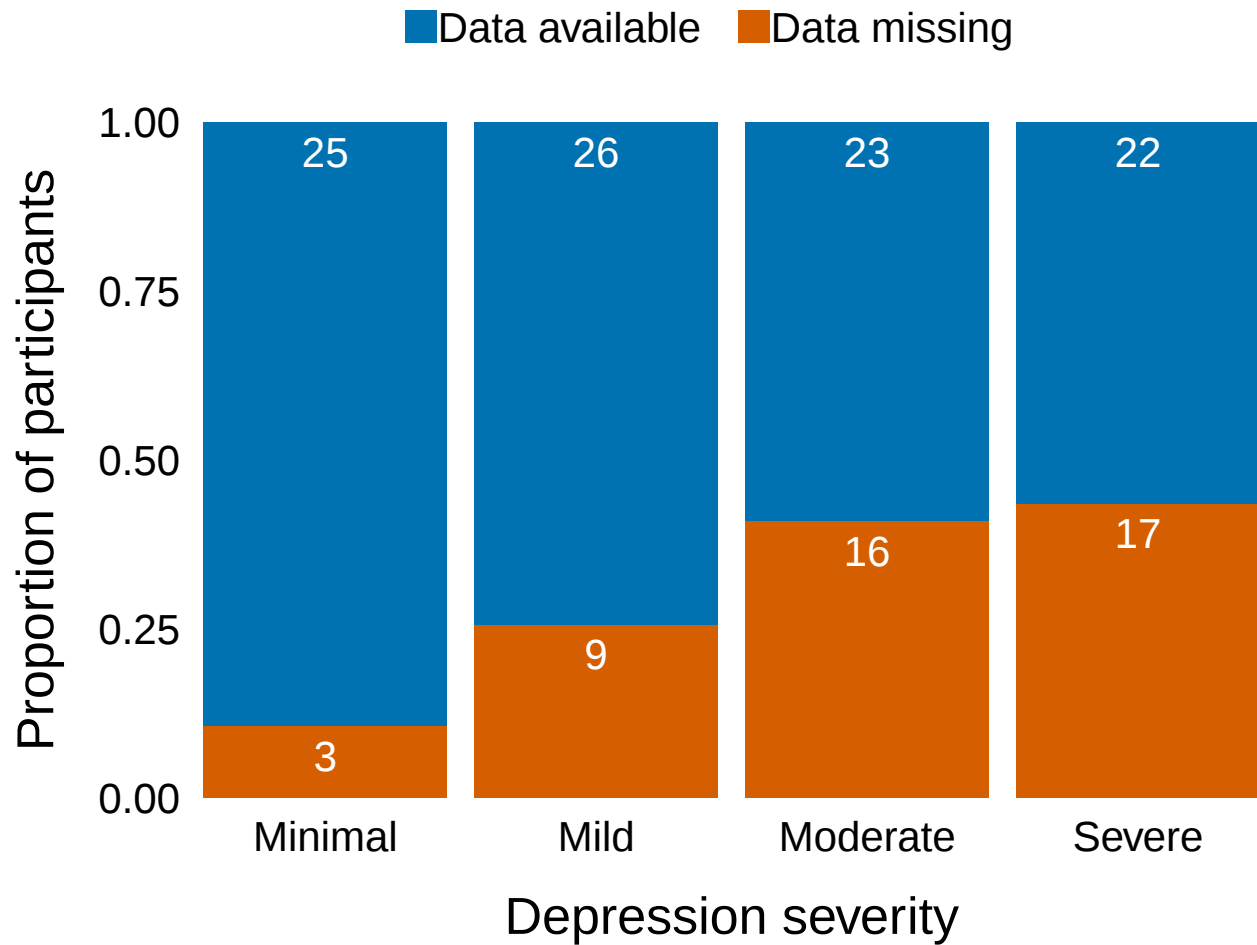
Income stability, sex, and group allocation did not predict whether or not an individual's data were present at 8 weeks. However, depression did: those with greater depression (on BDI) were more likely to have been lost to follow-up at the 8-week time point (main effect of depression severity: likelihood ratio = 11.31, df = 3, p = 0.01; OR for linear component of logistic regression = 4.01, 95% CI = 1.68 - 11.59).

```

#-- Manuscript plot --#
# Plot of proportion of participants with missing data at each level of
# depression severity, as rated on the Beck's Depression Inventory.
# Numbers in the blocks show the absolute counts.
# Note: 19 participants are missing baseline BDI data, so n = 141

# Colour (dark)
p1 <- bdi %>%
  filter(!is.na(bdi_category)) %>%
  mutate(bdi_category = fct_recode(bdi_category,
    Minimal = 'none-minimal',
    Mild = 'mild',
    Moderate = 'moderate-severe',
    Severe = 'severe'),
    coding = case_when(
      coding == 'Data available' ~ 'Data available',
      coding == 'Data missing' ~ 'Data missing'
    )) %>%
  ggplot(data = .) +
  aes(bdi_category,
    fill = coding) +
  geom_bar(position = position_fill()) +
  geom_text(stat = 'count',
    position = position_fill(),
    aes(label = ..count..),
    colour = '#FFFFFF',
    vjust = 1.5,
    size = 7.5) +
  labs(x = 'Depression severity',
    y = 'Proportion of participants') +
  scale_x_discrete(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_fill_manual(values = c('#0072B2', '#D55E00')) +
  theme_bw(base_size = 26) +
  theme(legend.position = 'top',
    legend.title = element_blank(),
    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_blank(),
    axis.ticks = element_blank()); p1

```



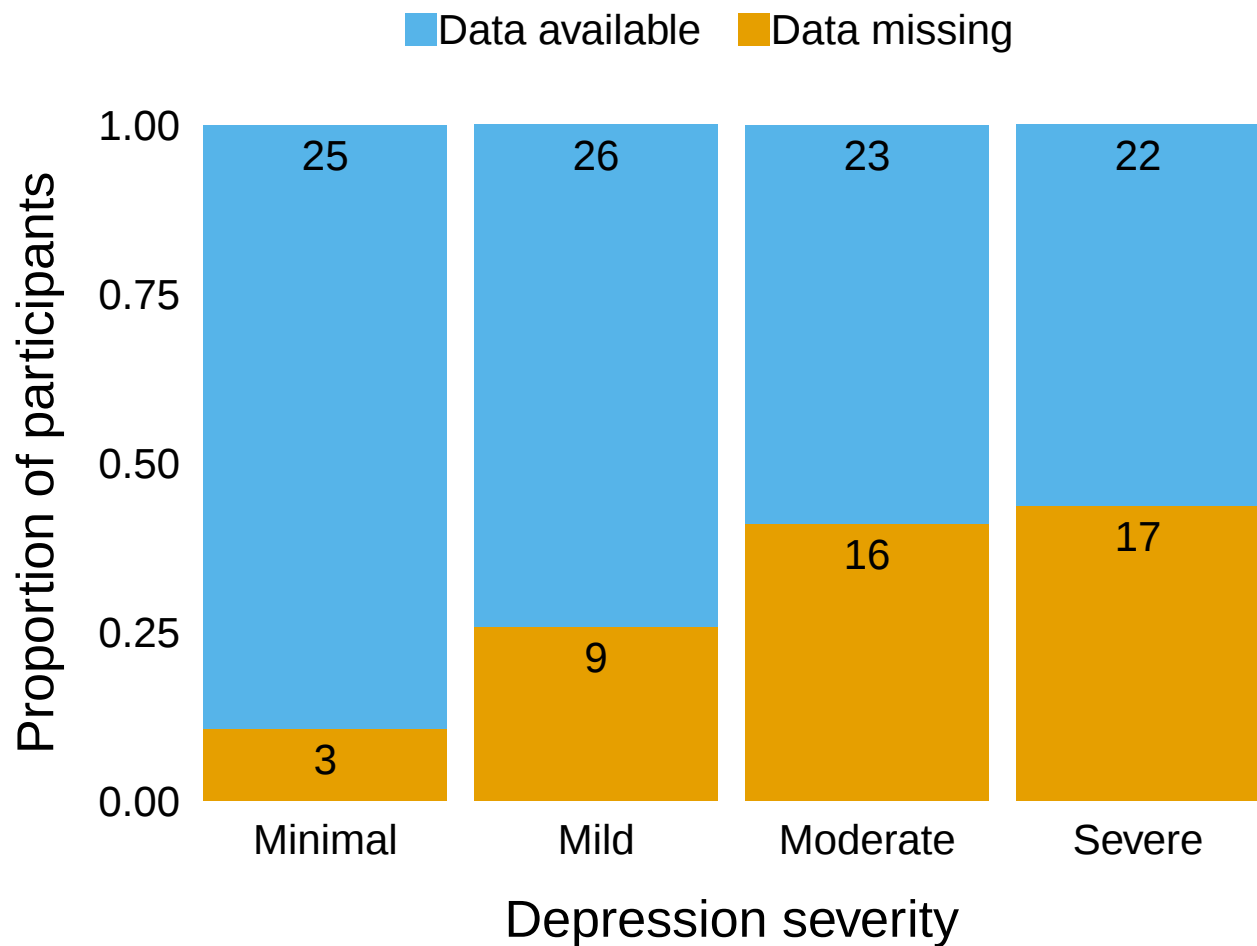
```
ggsave(filename = 'figures/figure-2_colourA.pdf',
        plot = p1,
        height = 8,
        width = 10,
        units = 'in')

# Colour (light)
p2 <- bdi %>%
  filter(!is.na(bdi_category)) %>%
  mutate(bdi_category = fct_recode(bdi_category,
                                   Minimal = 'none-minimal',
                                   Mild = 'mild',
                                   Moderate = 'moderate-severe',
                                   Severe = 'severe'),
         coding = case_when(
           coding == 'Data available' ~ 'Data available',
           coding == 'Data missing' ~ 'Data missing'
         )) %>%
  ggplot(data = .) +
  aes(bdi_category,
      fill = coding) +
  geom_bar(position = position_fill()) +
  geom_text(stat = 'count',
            position = position_fill(),
```

```

aes(label = ..count..),
colour = '#000000',
vjust = 1.5,
size = 7.5) +
labs(x = 'Depression severity',
     y = 'Proportion of participants') +
scale_x_discrete(expand = c(0, 0)) +
scale_y_continuous(expand = c(0, 0)) +
scale_fill_manual(values = c('#56B4E9', '#E69F00')) +
theme_bw(base_size = 26) +
theme(legend.position = 'top',
      legend.title = element_blank(),
      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_blank(),
      axis.ticks = element_blank()); p2

```



```

ggsave(filename = 'figures/figure-2_colourB.pdf',
        plot = p2,
        height = 8,

```

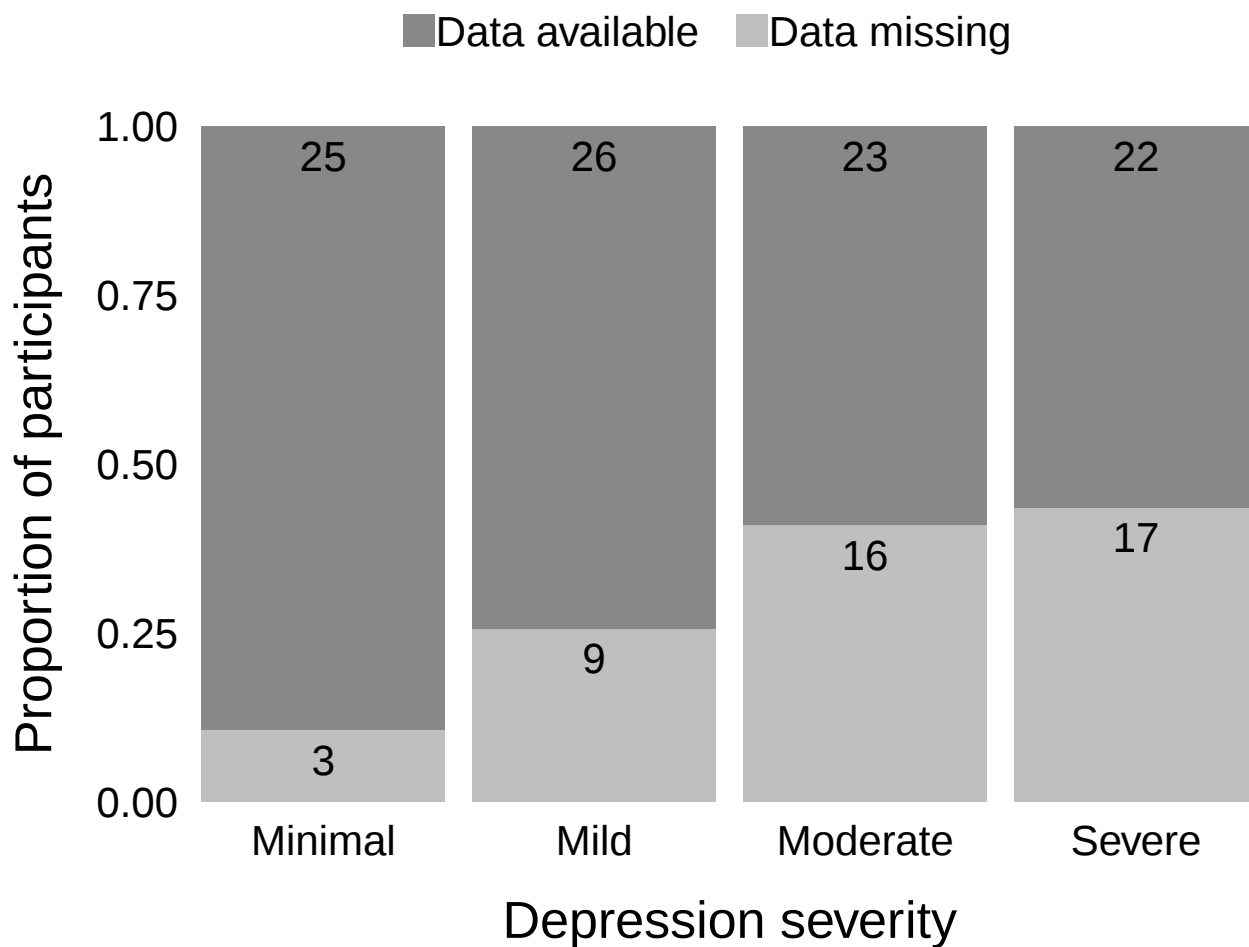
```

width = 10,
units = 'in')

# Greyscale
p3 <- bdi %>%
  filter(!is.na(bdi_category)) %>%
  mutate(bdi_category = fct_recode(bdi_category,
                                   Minimal = 'none-minimal',
                                   Mild = 'mild',
                                   Moderate = 'moderate-severe',
                                   Severe = 'severe'),

         coding = case_when(
           coding == 'Data available' ~ 'Data available',
           coding == 'Data missing' ~ 'Data missing'
         )) %>%
  ggplot(data = .) +
  aes(bdi_category,
      fill = coding) +
  geom_bar(position = position_fill()) +
  geom_text(stat = 'count',
            position = position_fill(),
            aes(label = ..count..),
            colour = '#000000',
            vjust = 1.5,
            size = 7.5) +
  labs(x = 'Depression severity',
       y = 'Proportion of participants') +
  scale_x_discrete(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_fill_manual(values = c('#888888', '#BFBFBF')) +
  theme_bw(base_size = 26) +
  theme(legend.position = 'top',
        legend.title = element_blank(),
        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_blank(),
        axis.ticks = element_blank()); p3

```



```
ggsave(filename = 'figures/figure-2_greyscale.pdf',
        plot = p3,
        height = 8,
        width = 10,
        units = 'in')
```

Session information

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## 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] car_3.0-2      carData_3.0-2  knitr_1.22     broom_0.5.2
## [5] forcats_0.4.0  stringr_1.4.0  dplyr_0.8.0.1  purrr_0.3.2
## [9] readr_1.3.1    tidyr_0.8.3    tibble_2.1.1   ggplot2_3.1.1
## [13] tidyverse_1.2.1 magrittr_1.5
##
## loaded via a namespace (and not attached):
## [1] tidyselect_0.2.5 xfun_0.6       haven_2.1.0
## [4] lattice_0.20-38  colorspace_1.4-1 generics_0.0.2
## [7] htmltools_0.3.6  yaml_2.2.0     utf8_1.1.4
## [10] rlang_0.3.4      pillar_1.3.1   foreign_0.8-71
## [13] glue_1.3.1       withr_2.1.2.9000 modelr_0.1.4
## [16] readxl_1.3.1     plyr_1.8.4     munsell_0.5.0
## [19] gtable_0.3.0     cellranger_1.1.0 zip_2.0.1
## [22] rvest_0.3.3      evaluate_0.13   labeling_0.3
## [25] rio_0.5.16       curl_3.3        fansi_0.4.0
## [28] highr_0.8        Rcpp_1.0.1      scales_1.0.0
## [31] backports_1.1.4  jsonlite_1.6    abind_1.4-5
## [34] hms_0.4.2        digest_0.6.18   openxlsx_4.1.0
## [37] stringi_1.4.3    grid_3.6.0      cli_1.1.0
## [40] tools_3.6.0      lazyeval_0.2.2  crayon_1.3.4
## [43] pkgconfig_2.0.2  MASS_7.3-51.4   data.table_1.12.2
## [46] xml2_1.2.0       lubridate_1.7.4 assertthat_0.2.1
## [49] rmarkdown_1.12   httr_1.4.0      rstudioapi_0.10
## [52] R6_2.4.0         nlme_3.1-139    compiler_3.6.0
```

Supplement 4

Analysis of the primary outcome

Peter Kamerman and Tory Madden

07 May 2019

Contents

Import and sort data	1
Quick look	2
BOCF data processing and analysis	3
Per protocol data processing and analysis	8
Treatment dose	13
Number of assessments attended on average	13
Relationship between treatment session attendance (Group P) and outcome	14
Session information	15

The primary outcome was the reduction in pain severity score (PSS) between week 0 (BL) and week 48 (Wk48). The PSS is a composite measure of the mean of: pain at the assessment, and worst, least, and average pain in the last week.

The baseline observation carried forward (BOCF) analysis used data from week 0 to interpolate missing week 48 data. Some participants did not arrive for their week 0 appointment and therefore did not have data at baseline. These missing data were interpolated using the week 0 study site average PSS. No heed was paid to whether data were available at intermediate time points between week 0 and week 48.

The per protocol (PP) analysis included all individuals with complete data (i.e., no interpolation required).

Statistical significance was assessed using a permutation test that tested the independence of the therapeutic relationship only group (T, control) and the P groups (positive-living programme and therapeutic relationship). The conditional null distribution of the test statistic was calculated using Monte Carlo resampling ($n = 100000$).

Dosage was calculated as the number of assessments attended, on average) by participants in each group, and the number of intervention sessions attended by participants in group P and the change in NRS. *Note that the latter analysis excludes site U1 because the data were inadvertently destroyed.*

Import and sort data

```
# Get BPI data
## BPI
bpi <- read_rds('data-cleaned/bpi.rds') %>%
  select(ID, contains('Pain'))

## Demographics
```

```

demo <- read_rds('data-cleaned/demographics.rds') %>%
  select(ID, Study_site, Group)

## Dosage
dosage <- read_rds('data-cleaned/dosage.rds')

## Join BPI and demographics
data <- demo %>%
  left_join(bpi)

## Add dosage data
data %<>% left_join(dosage)

## Primary outcome
primary <- data %>%
  select(-contains('present'))

```

Quick look

```
glimpse(primary)
```

```

## Observations: 160
## Variables: 28
## $ ID <chr> "J1", "J3", "J4", "J5", "J6", "J7", "J9", "J...
## $ Study_site <chr> "U1", "U1", "U1", "U1", "U1", "U1", "U1", "U...
## $ Group <chr> "P", "T", "P", "P", "P", "T", "T", "T", "P",...
## $ Worst_pain.BL <int> 8, 9, 5, 7, 7, 8, 10, 10, 9, 0, 10, NA, 9, 8...
## $ Worst_pain.Wk4 <int> NA, NA, 3, NA, NA, 8, 8, 9, 8, NA, NA, NA, 8...
## $ Worst_pain.Wk8 <int> NA, NA, 0, NA, NA, NA, 8, 9, 10, 4, NA, NA, ...
## $ Worst_pain.Wk12 <int> NA, NA, 3, NA, NA, NA, 7, 9, 10, 7, 10, NA, ...
## $ Worst_pain.Wk24 <int> NA, NA, 6, NA, NA, NA, 7, 9, NA, 6, NA, NA, ...
## $ Worst_pain.Wk48 <int> NA, NA, 6, NA, NA, NA, 7, 8, NA, 8, NA, NA, ...
## $ Least_pain.BL <int> 4, 1, 1, 10, 2, 3, 3, 5, 3, 0, 3, NA, 3, 3, ...
## $ Least_pain.Wk4 <int> NA, NA, 1, NA, NA, 3, 5, 4, 2, NA, NA, NA, 3...
## $ Least_pain.Wk8 <int> NA, NA, 0, NA, NA, NA, 3, 4, 5, 1, NA, NA, 4...
## $ Least_pain.Wk12 <int> NA, NA, 1, NA, NA, NA, 3, 6, 5, 4, 5, NA, NA...
## $ Least_pain.Wk24 <int> NA, NA, 2, NA, NA, NA, 3, 5, NA, 2, NA, NA, ...
## $ Least_pain.Wk48 <int> NA, NA, 2, NA, NA, NA, 3, 5, NA, 2, NA, NA, ...
## $ Average_pain.BL <int> 4, 4, 3, 5, 4, 6, 6, 7, 6, 0, 6, NA, 6, 6, 5...
## $ Average_pain.Wk4 <int> NA, NA, 1, NA, NA, 5, 5, 6, 5, NA, NA, NA, 5...
## $ Average_pain.Wk8 <int> NA, NA, 0, NA, NA, NA, 5, 7, 8, 2, NA, NA, 6...
## $ Average_pain.Wk12 <int> NA, NA, 2, NA, NA, NA, 4, 7, 7, 6, 8, NA, NA...
## $ Average_pain.Wk24 <int> NA, NA, 4, NA, NA, NA, 5, 7, NA, 4, NA, NA, ...
## $ Average_pain.Wk48 <int> NA, NA, 4, NA, NA, NA, 5, 7, NA, 5, NA, NA, ...
## $ Pain_now.BL <int> 6, 4, 0, 5, 2, 9, 0, 0, 9, 0, 3, NA, 0, 8, 6...
## $ Pain_now.Wk4 <int> NA, NA, 0, NA, NA, 8, 8, 4, 6, NA, NA, NA, 0...
## $ Pain_now.Wk8 <int> NA, NA, 0, NA, NA, NA, 3, 5, 10, 1, NA, NA, ...
## $ Pain_now.Wk12 <int> NA, NA, 2, NA, NA, NA, 8, 5, 10, 3, 8, NA, N...
## $ Pain_now.Wk24 <int> NA, NA, 2, NA, NA, NA, 3, 8, NA, 3, NA, NA, ...
## $ Pain_now.Wk48 <int> NA, NA, 4, NA, NA, NA, 3, 2, NA, 5, NA, NA, ...
## $ dosage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...

```

BOCF data processing and analysis

```
# Calculate Pain Severity Score (PSS) for weeks 0 (BL) and 48 (Wk48)
BOCF <- primary %>%
  gather(key = 'Time',
         value = 'NRS',
         - ID, -Study_site, -Group) %>%
  separate(col = Time,
         into = c('Pain', 'Period'),
         sep = '\\\\.') %>%
  group_by(ID, Study_site, Group, Period) %>%
  nest() %>%
  mutate(PPS = map(.x = data,
                   ~ summarise(.x, Mean = mean(NRS, na.rm = TRUE)))) %>%
  select(-data) %>%
  unnest() %>%
  filter(Period %in% c('BL', 'Wk48')) %>%
  mutate(Mean = ifelse(is.nan(Mean),
                      yes = NA,
                      no = Mean))

# Missing baseline data
BOCF[is.na(BOCF$Mean) & BOCF$Period == 'BL', ]

## # A tibble: 16 x 5
##   ID      Study_site Group Period  Mean
##   <chr> <chr>      <chr> <chr> <dbl>
## 1 J18    U1          T     BL    NA
## 2 J29    U1          T     BL    NA
## 3 J59    U1          T     BL    NA
## 4 J67    U1          T     BL    NA
## 5 M1     R1          T     BL    NA
## 6 M8     R1          T     BL    NA
## 7 M9     R1          T     BL    NA
## 8 M12    R1          T     BL    NA
## 9 M20    R1          P     BL    NA
## 10 M21   R1          P     BL    NA
## 11 M23   R1          P     BL    NA
## 12 M25   R1          P     BL    NA
## 13 M29   R1          P     BL    NA
## 14 M38   R1          P     BL    NA
## 15 M41   R1          P     BL    NA
## 16 M45   R1          T     BL    NA

# Number of participants with missing baselines
nrow(BOCF[is.na(BOCF$Mean) & BOCF$Period == 'BL', ])

## [1] 16

# Calculate baseline mean for study sites R1 and U1 (sites with missing data)
R1 <- mean(BOCF$Mean[!is.na(BOCF$Mean) &
                  BOCF$Study_site == 'R1' &
```

```

      BOCF$Period == 'BL'])

U1 <- mean(BOCF$Mean[!is.na(BOCF$Mean) &
      BOCF$Study_site == 'U1' &
      BOCF$Period == 'BL'])

# Substitute missing baselines with Study_site means
BOCF %<>%
  mutate(Mean = ifelse(is.na(Mean) & Study_site == 'R1',
    yes = R1,
    no = Mean),
    Mean = ifelse(is.na(Mean) & Study_site == 'U1',
    yes = U1,
    no = Mean))

# Spread data
BOCF_wide <- BOCF %>%
  spread(key = Period,
    value = Mean)

# Number of participants
BOCF_wide %>%
  summarise(Count = n())

## # A tibble: 1 x 1
##   Count
##   <int>
## 1    160

# Number with missing data at week 48
BOCF_wide %>%
  filter(is.na(Wk48)) %>%
  summarise(Count = n())

## # A tibble: 1 x 1
##   Count
##   <int>
## 1     35

# Number with missing data at week 48 (by intervention)
BOCF_wide %>%
  filter(is.na(Wk48)) %>%
  group_by(Group) %>%
  summarise(Count = n())

## # A tibble: 2 x 2
##   Group Count
##   <chr> <int>
## 1 P      20
## 2 T      15

# Create BOCF dataframe
BOCF_wide %<>%
  mutate(Wk48 = ifelse(is.na(Wk48),
    yes = BL,
    no = Wk48))

```

```
# Plot of BOCF NRS data at weeks 0 and 48 (by intervention)
```

```
BOCF_wide %>%
```

```
  gather(key = Period,
         value = NRS,
         BL, Wk48) %>%
```

```
  ggplot(data = .) +
```

```
  aes(x = Period,
      y = NRS,
      colour = Group,
      fill = Group) +
```

```
  geom_boxplot(alpha = 0.6) +
```

```
  geom_point(position = position_jitterdodge(jitter.height = 0,
                                             jitter.width = 0.3),
```

```
          shape = 21,
          fill = '#FFFFFF',
          size = 2) +
```

```
  labs(title = 'BOCF: NRS at week 0 and week 48',
```

```
       subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship only',
```

```
       x = 'Week') +
```

```
  scale_fill_manual(values = pal) +
```

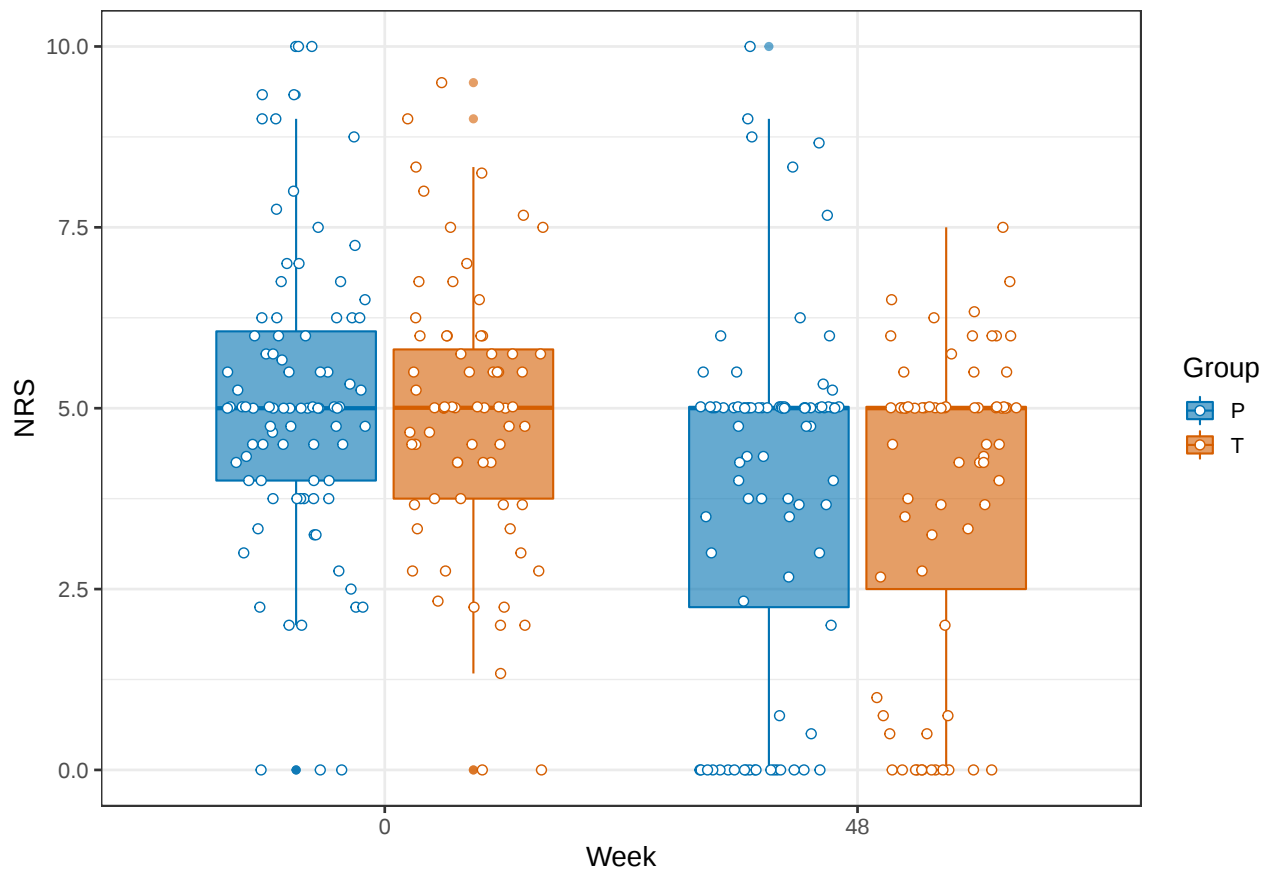
```
  scale_colour_manual(values = pal) +
```

```
  scale_x_discrete(labels = c(0, 48))
```

BOCF: NRS at week 0 and week 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



```

# Calculate change from baseline to Wk48
BOCF_wide %<>%
  mutate(Delta = Wk48 - BL)

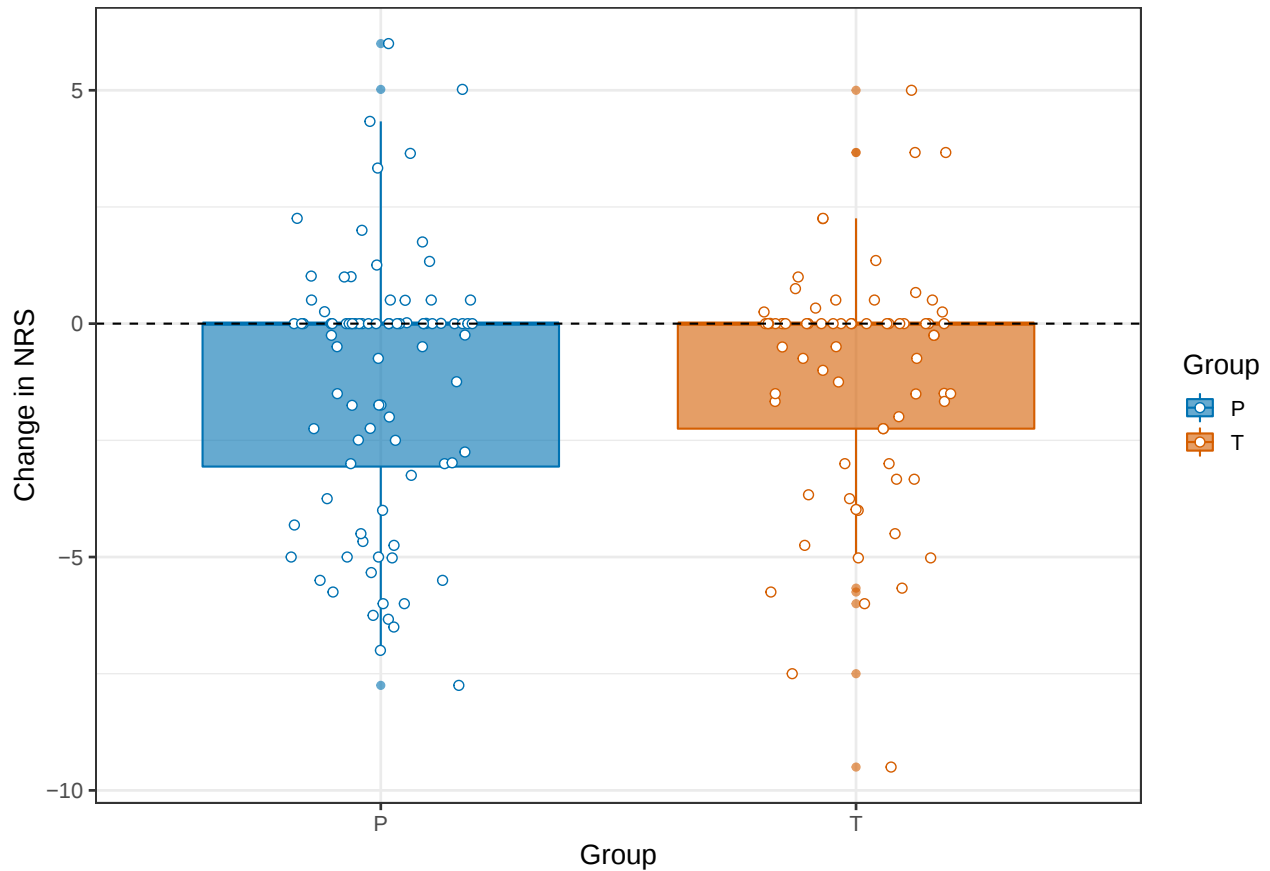
# Compare the P and T groups
## Plot of BOCF change in NRS between week 0 and 48 (by intervention)
ggplot(data = BOCF_wide) +
  aes(x = Group,
      y = Delta,
      fill = Group,
      colour = Group) +
  geom_boxplot(alpha = 0.6) +
  geom_point(position = position_jitterdodge(jitter.height = 0,
                                             jitter.width = 0.4),
             shape = 21,
             size = 2,
             fill = '#FFFFFF') +
  geom_hline(yintercept = 0,
             linetype = 2) +
  labs(title = 'BOCF: Change in NRS from week 0 to week 48',
       subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship',
       y = 'Change in NRS') +
  scale_fill_manual(values = pal) +
  scale_colour_manual(values = pal)

```

BOCF: Change in NRS from week 0 to week 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



Slope plot of individual responses

BOCF_wide %>%

```
mutate(Change = ifelse(Delta < 0,
                        yes = 'Decreased',
                        no = 'Increased')) %>%
```

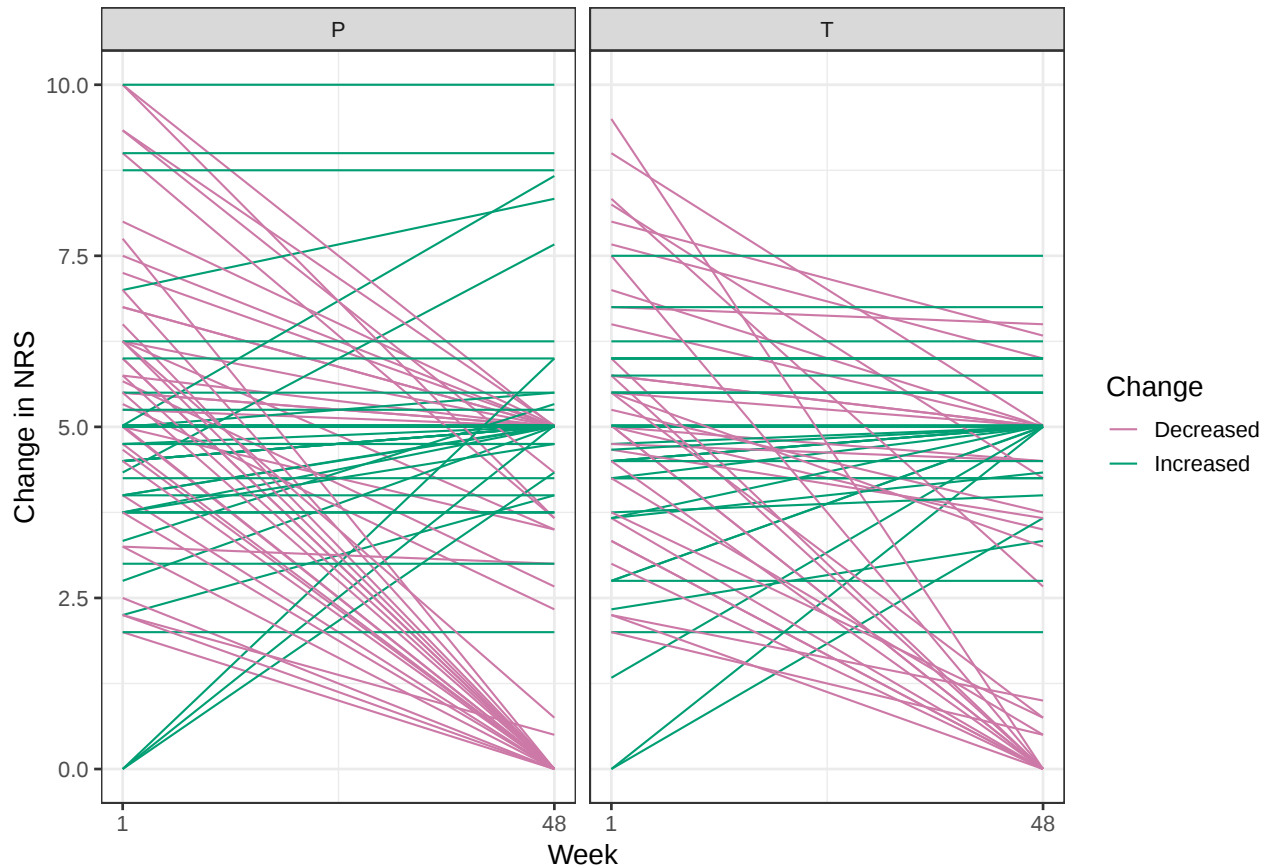
```
ggplot(data = .) +
  aes(colour = Change) +
  geom_segment(aes(x = 1, xend = 2,
                  y = BL, yend = Wk48)) +
  scale_x_continuous(breaks = c(1, 2),
                    labels = c(1, 48)) +
  scale_colour_manual(values = pal2) +
```

```
labs(title = "BOCF: Individuals' changes in NRS from week 0 to week 48",
     subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship only',
     x = 'Week',
     y = 'Change in NRS') +
  facet_wrap(~ Group)
```

BOCF: Individuals' changes in NRS from week 0 to week 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



```
## Statistical test
independence_test(Delta ~ factor(Group),
                  data = BOCF_wide,
                  distribution = approximate(nresample = 100000))

##
## Approximative General Independence Test
##
## data: Delta by factor(Group) (P, T)
## Z = -0.4306, p-value = 0.6695
## alternative hypothesis: two.sided
```

Per protocol data processing and analysis

```
# Calculate Pain Severity Score (PSS) for weeks 0 (BL) and 48 (Wk48)
PP_wide <- primary %>%
  gather(key = 'Time',
         value = 'NRS',
         - ID, -Study_site, -Group) %>%
  separate(col = Time,
```

```

        into = c('Pain', 'Period'),
        sep = '\\\\.') %>%
group_by(ID, Study_site, Group, Period) %>%
nest() %>%
mutate(PPS = map(.x = data,
~ summarise(.x, Mean = mean(NRS, na.rm = TRUE)))) %>%
select(-data) %>%
unnest() %>%
mutate(Mean = ifelse(is.nan(Mean),
yes = NA,
no = Mean)) %>%
mutate(Period = case_when(
Period == 'BL' ~ 'T00',
Period == 'Wk4' ~ 'T04',
Period == 'Wk8' ~ 'T08',
Period == 'Wk12' ~ 'T12',
Period == 'Wk24' ~ 'T24',
Period == 'Wk48' ~ 'T48'
)) %>%
spread(key = Period,
value = Mean) %>%
select(-`<NA>`)

# PP cohort
PP_wide %>%
filter(complete.cases())

# Number of participants with complete data
PP_wide %>%
summarise(Count = n())

## # A tibble: 1 x 1
##   Count
##   <int>
## 1     32

# Number of participants with complete data (by intervention)
PP_wide %>%
group_by(Group) %>%
summarise(Count = n())

## # A tibble: 2 x 2
##   Group Count
##   <chr> <int>
## 1 P      19
## 2 T      13

# Plot of BOCF NRS data at weeks 0 and 48 (by intervention)
PP_wide %>%
gather(key = Period,
value = NRS,
-ID, -Study_site, -Group) %>%
ggplot(data = .) +
aes(x = Period,
y = NRS,
colour = Group,

```

```

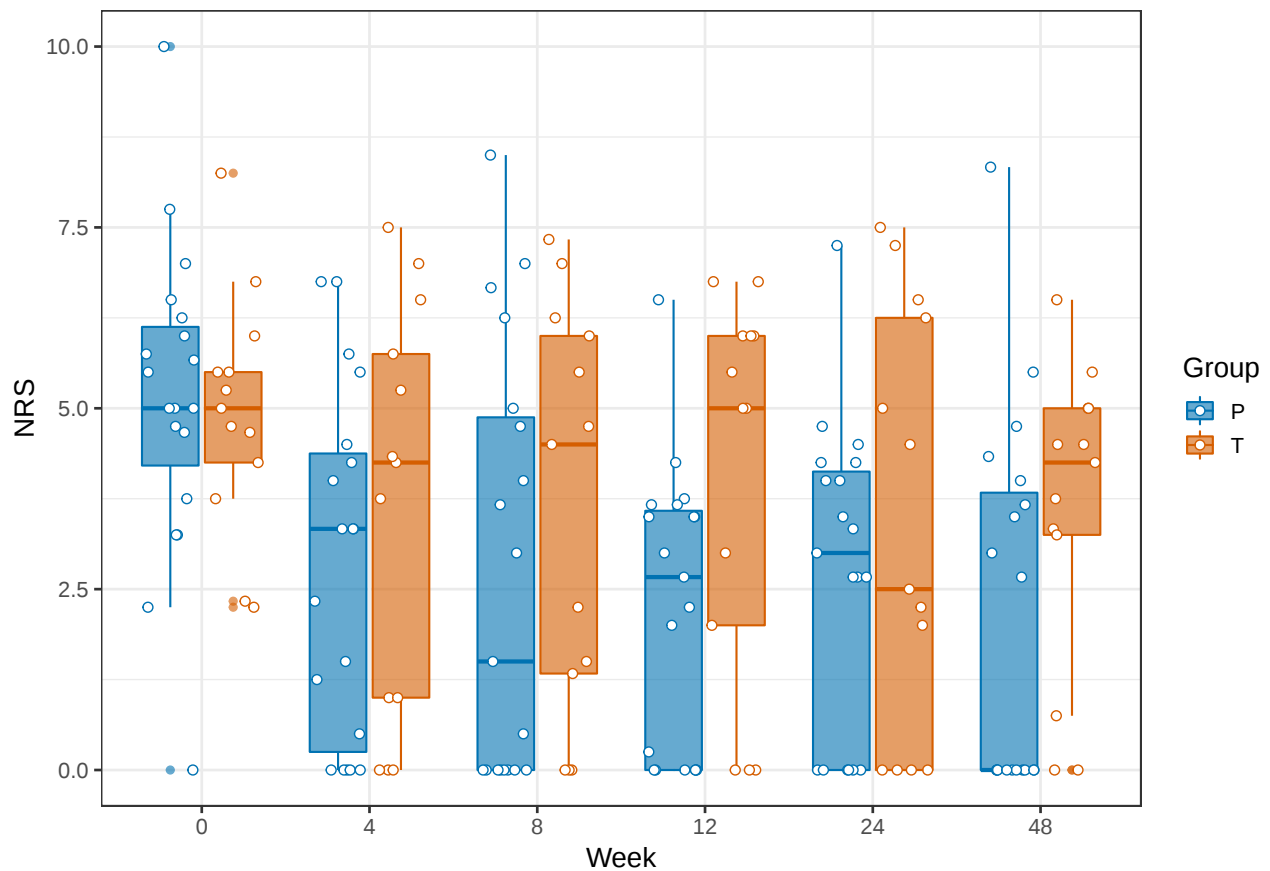
    fill = Group) +
  geom_boxplot(alpha = 0.6) +
  geom_point(position = position_jitterdodge(jitter.height = 0,
                                             jitter.width = 0.3),
             shape = 21,
             fill = '#FFFFFF',
             size = 2) +
  labs(title = 'PP: NRS at weeks 0 through to 48',
        subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship only',
        x = 'Week') +
  scale_fill_manual(values = pal) +
  scale_colour_manual(values = pal) +
  scale_x_discrete(labels = c(0, 4, 8, 12, 24, 48))

```

PP: NRS at weeks 0 through to 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



```
# Calculate change from baseline to Wk48
```

```
PP_wide %<>%
```

```
  mutate(Delta = T48 - T00)
```

```
# Compare the P and T groups
```

```
## Plot of BOCF change in NRS between week 0 and 48 (by intervention)
```

```
ggplot(data = PP_wide) +
```

```
  aes(x = Group,
```

```
      y = Delta,
```

```

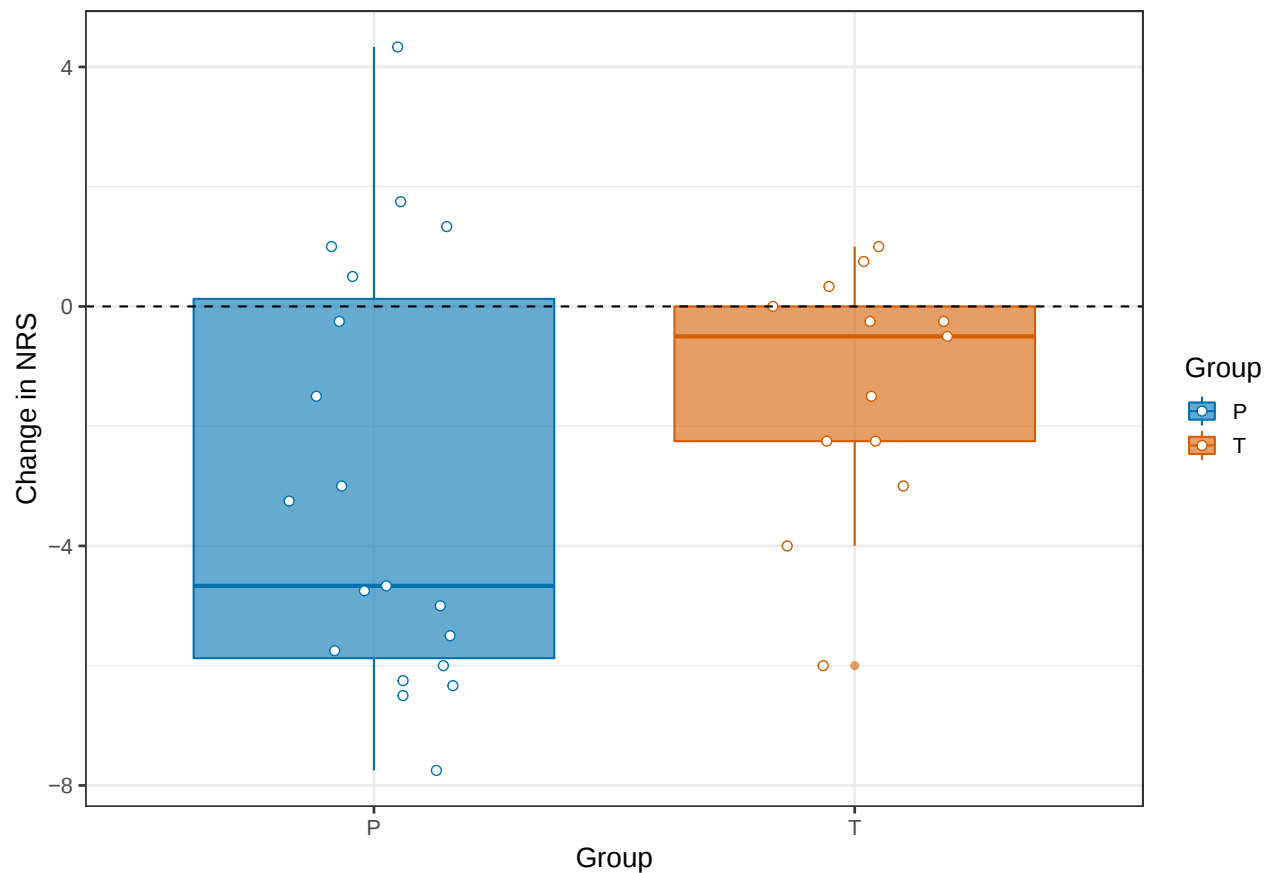
    fill = Group,
    colour = Group) +
  geom_boxplot(alpha = 0.6) +
  geom_point(position = position_jitterdodge(jitter.height = 0,
                                             jitter.width = 0.4),
             shape = 21,
             size = 2,
             fill = '#FFFFFF') +
  geom_hline(yintercept = 0,
             linetype = 2) +
  labs(title = 'PP: Change in NRS from week 0 to week 48',
       subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship only',
       y = 'Change in NRS') +
  scale_fill_manual(values = pal) +
  scale_colour_manual(values = pal)

```

PP: Change in NRS from week 0 to week 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



```

# Slope plot of individual responses
PP_wide %>%
  mutate(Change = ifelse(Delta < 0,
                          yes = 'Decreased',
                          no = 'Increased')) %>%
  ggplot(data = .) +
  aes(colour = Change) +

```

```

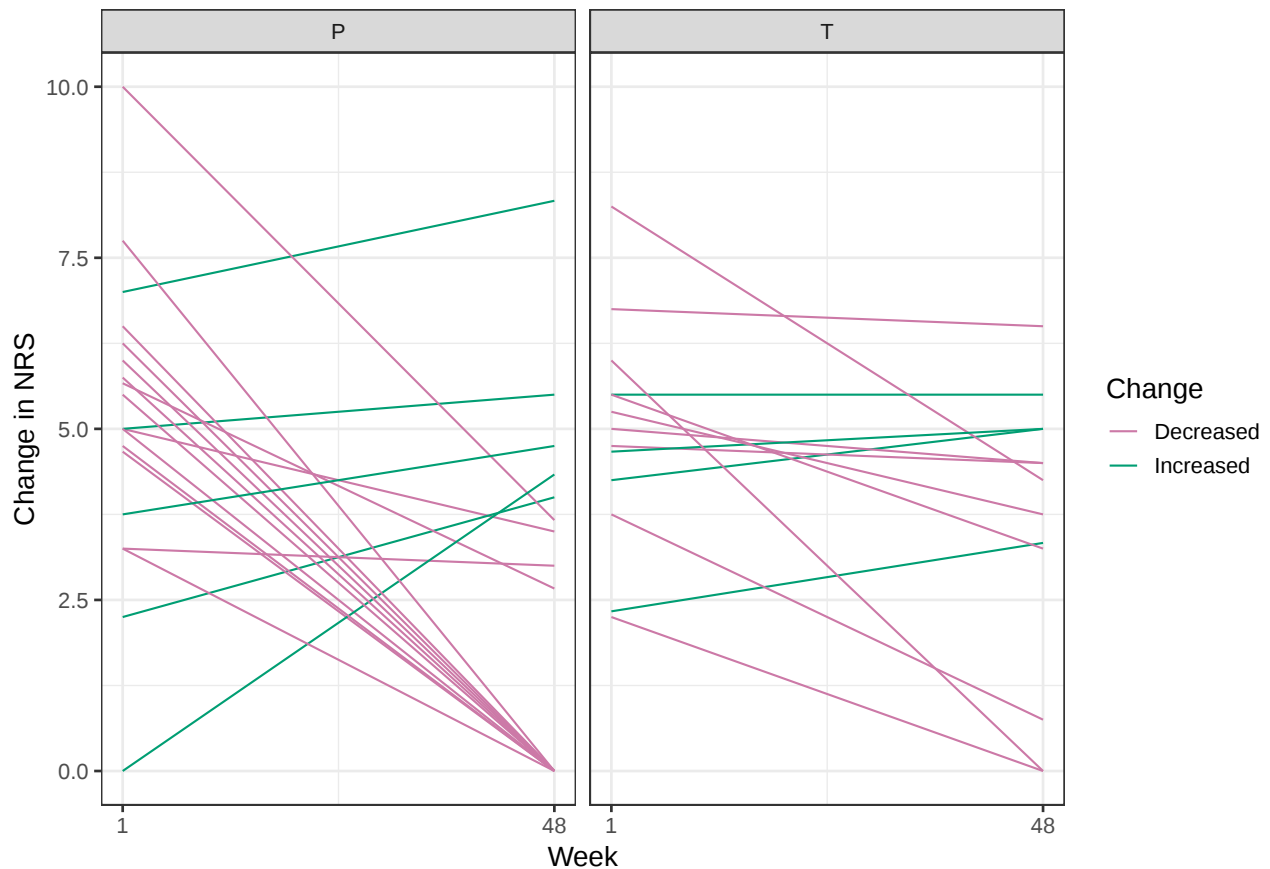
geom_segment(aes(x = 1, xend = 2,
                 y = T00, yend = T48)) +
scale_x_continuous(breaks = c(1, 2),
                  labels = c(1, 48)) +
scale_colour_manual(values = pal2) +
labs(title = "PP: Individuals' changes in NRS from week 0 to week 48",
     subtitle = 'P: Positive-living programme and therapeutic relationship\nT: Therapeutic relationship only',
     x = 'Week',
     y = 'Change in NRS') +
facet_wrap(~ Group)

```

PP: Individuals' changes in NRS from week 0 to week 48

P: Positive-living programme and therapeutic relationship

T: Therapeutic relationship only



```

## Statistical test
independence_test(Delta ~ factor(Group),
                 data = PP_wide,
                 distribution = approximate(nresample = 100000))

##
## Approximative General Independence Test
##
## data: Delta by factor(Group) (P, T)
## Z = -1.4904, p-value = 0.1387
## alternative hypothesis: two.sided

```

Treatment dose

Number of assessments attended on average

```
dose <- primary %>%
  gather(key = 'Time',
         value = 'NRS',
         - ID, -Study_site, -Group) %>%
  separate(col = Time,
           into = c('Pain', 'Period'),
           sep = '\\\\.') %>%
  group_by(ID, Study_site, Group, Period) %>%
  nest() %>%
  mutate(PPS = map(.x = data,
                  ~ summarise(.x, Mean = mean(NRS, na.rm = TRUE)))) %>%
  select(-data) %>%
  unnest() %>%
  mutate(Mean = ifelse(is.nan(Mean),
                      yes = NA,
                      no = Mean)) %>%
  mutate(Period = case_when(
    Period == 'BL' ~ 'T00',
    Period == 'Wk4' ~ 'T04',
    Period == 'Wk8' ~ 'T08',
    Period == 'Wk12' ~ 'T12',
    Period == 'Wk24' ~ 'T24',
    Period == 'Wk48' ~ 'T48'
  ))

# Calculate dose per individual
dose_b <- dose %>%
  mutate(Mean = ifelse(is.na(Mean),
                      yes = 0,
                      no = 1)) %>%
  group_by(Group, Study_site, ID) %>%
  summarise(Count = sum(Mean))

# Calculate summary stat for dose
dose_b %>%
  ungroup() %>%
  summarise(Median = median(Count),
            q25 = quantile(Count, probs = 0.25),
            a75 = quantile(Count, probs = 0.75),
            min = min(Count),
            max = max(Count))

## # A tibble: 1 x 5
##   Median q25 a75 min max
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     5     3     6     0     7

# Calculate summary stat for dose (by intervention)
dose_b %>%
  group_by(Group) %>%
```

```

summarise(Median = median(Count),
          q25 = quantile(Count, probs = 0.25),
          a75 = quantile(Count, probs = 0.75),
          min = min(Count),
          max = max(Count))

## # A tibble: 2 x 6
##   Group Median   q25   a75   min   max
##   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 P         5     3     6     1     7
## 2 T         4     2     5     0     6

```

Relationship between treatment session attendance (Group P) and outcome

Excludes site U1 (data missing).

```

# Spread dose data
dose %<>%
  spread(key = Period,
         value = Mean)

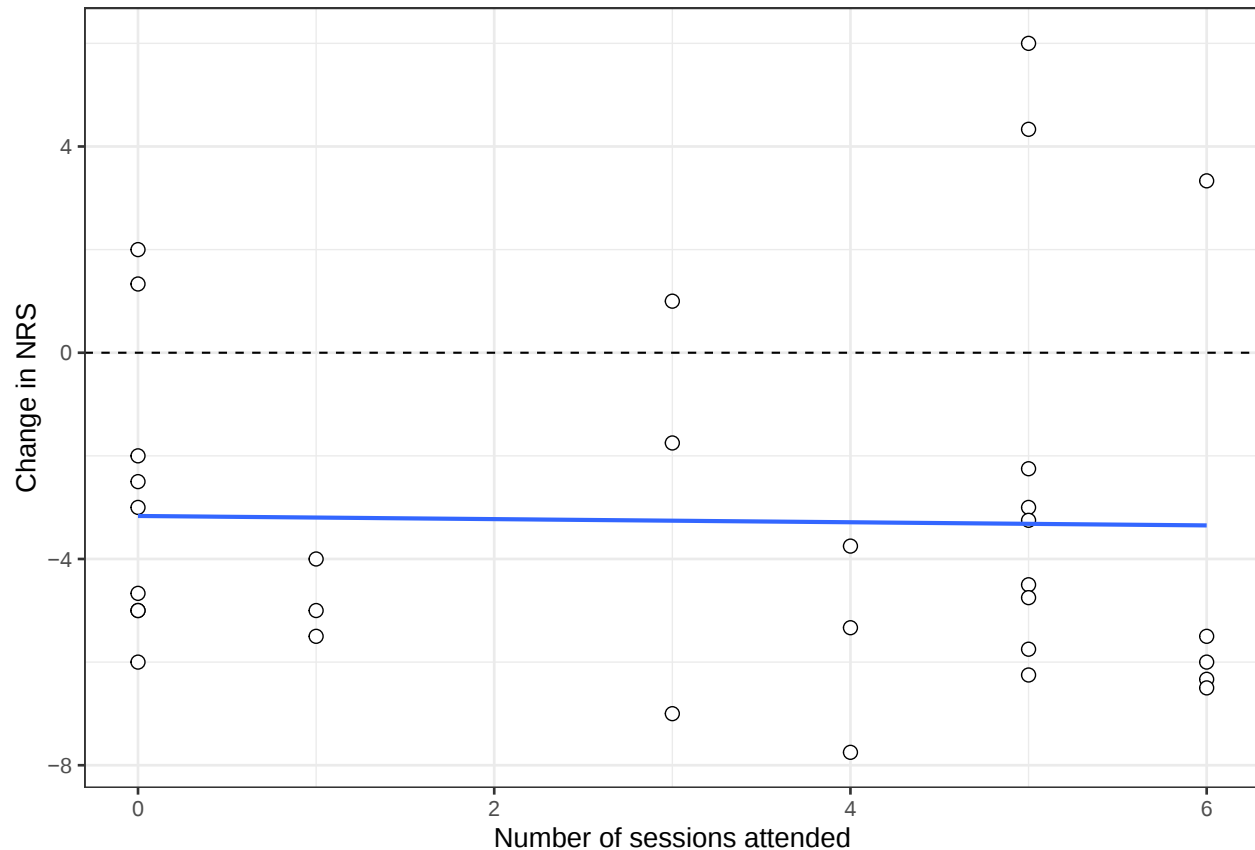
# Calculate change from baseline to Wk48, PL group only
dose %<>%
  mutate(Delta = T48 - T00) %>%
  left_join(dosage) %>%
  filter(Group == 'P') %>%
  filter(Delta != 'NA') %>%
  filter(dosage != 'NA') # filters out 5 participants all from J site

# Plot of P group's change in NRS between week 0 and 48
# (by PL session attendance)
ggplot(data = dose) +
  aes(x = dosage,
      y = Delta) +
  geom_point(shape = 21,
            fill = '#FFFFFF',
            size = 3) +
  geom_hline(yintercept = 0,
            linetype = 2) +
  geom_smooth(method = 'lm',
            se = FALSE) +
  labs(title = 'Change in NRS from week 0 to week 48\nby number of sessions attended',
       subtitle = 'Positive-living group only',
       y = 'Change in NRS',
       x = 'Number of sessions attended')

```

Change in NRS from week 0 to week 48 by number of sessions attended

Positive-living group only



Session information

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## 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] coin_1.3-0      survival_2.44-1.1 forcats_0.4.0
## [4] stringr_1.4.0   dplyr_0.8.0.1    purrr_0.3.2
## [7] readr_1.3.1     tidyr_0.8.3      tibble_2.1.1
```

```

## [10] ggplot2_3.1.1      tidyverse_1.2.1    magrittr_1.5
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.1          lubridate_1.7.4     mvtnorm_1.0-10
## [4] lattice_0.20-38     zoo_1.8-5           assertthat_0.2.1
## [7] digest_0.6.18       utf8_1.1.4          R6_2.4.0
## [10] cellranger_1.1.0    plyr_1.8.4          backports_1.1.4
## [13] stats4_3.6.0        evaluate_0.13       httr_1.4.0
## [16] pillar_1.3.1        rlang_0.3.4         lazyeval_0.2.2
## [19] multcomp_1.4-10     readxl_1.3.1        rstudioapi_0.10
## [22] Matrix_1.2-17       rmarkdown_1.12      labeling_0.3
## [25] splines_3.6.0       munsell_0.5.0       broom_0.5.2
## [28] compiler_3.6.0      modelr_0.1.4         xfun_0.6
## [31] pkgconfig_2.0.2     libcoin_1.0-4       htmltools_0.3.6
## [34] tidyselect_0.2.5    codetools_0.2-16    matrixStats_0.54.0
## [37] fansi_0.4.0         crayon_1.3.4        withr_2.1.2.9000
## [40] MASS_7.3-51.4       grid_3.6.0          nlme_3.1-139
## [43] jsonlite_1.6        gtable_0.3.0        scales_1.0.0
## [46] cli_1.1.0           stringi_1.4.3       xml2_1.2.0
## [49] generics_0.0.2      sandwich_2.5-1      TH.data_1.0-10
## [52] tools_3.6.0         glue_1.3.1          hms_0.4.2
## [55] parallel_3.6.0      yaml_2.2.0          colorspace_1.4-1
## [58] rvest_0.3.3         knitr_1.22          haven_2.1.0
## [61] modeltools_0.2-22

```