Supplement 3

Descriptive statistics on participant follow-up

Peter Kamerman and Prinisha Pillay

18 May 2019

Contents

This analysis describes the follow-up pattern of participants in the study.

Import data

```
data <- read_rds('data-cleaned/clean_data.rds') %>%
    # Select columns
    select(ID, visit_number, visit_day, visit_months, hivsn_present)
```

Process data

Add a column indicating whether a participant developed SN at any time during the follow-up period.

```
# Identify andf extract information on SN development (at anytime)
# by looking at the presence of SN at the final visit
data_sn <- data %>%
    select(ID, visit_number, hivsn_present) %>%
    group_by(ID) %>%
    mutate(max_visit = max(visit_number)) %>%
    filter(visit_number == max_visit) %>%
    select(ID, hivsn_present) %>%
    rename(sn = hivsn_present)
```

```r
# Join data_sn to data
data %<>%
    left_join(data_sn)
```

Inspect data

```r
# Dimensions
dim(data)
```

```
## [1] 371   6
```

```r
# Column names
names(data)
```

```
## [1] "ID"            "visit_number"  "visit_day"     "visit_months"
## [5] "hivsn_present" "sn"
```

```r
# Head and tail
head(data)
```

```
## # A tibble: 6 x 6
##   ID    visit_number visit_day visit_months hivsn_present sn
##   <chr>        <int>     <int>        <dbl> <fct>         <fct>
## 1 001              1         0            0 no            no
## 2 001              2        40            1 no            no
## 3 001              3       210            7 no            no
## 4 002              1         0            0 no            no
## 5 002              2        71            2 no            no
## 6 002              3       191            6 no            no
```

```r
tail(data)
```

```
## # A tibble: 6 x 6
##   ID    visit_number visit_day visit_months hivsn_present sn
##   <chr>        <int>     <int>        <dbl> <fct>         <fct>
## 1 119              1         0            0 no            no
## 2 119              2        72            2 no            no
## 3 119              3       132            4 no            no
## 4 120              1         0            0 no            no
## 5 120              2        88            3 no            no
## 6 120              3       124            4 no            no
```

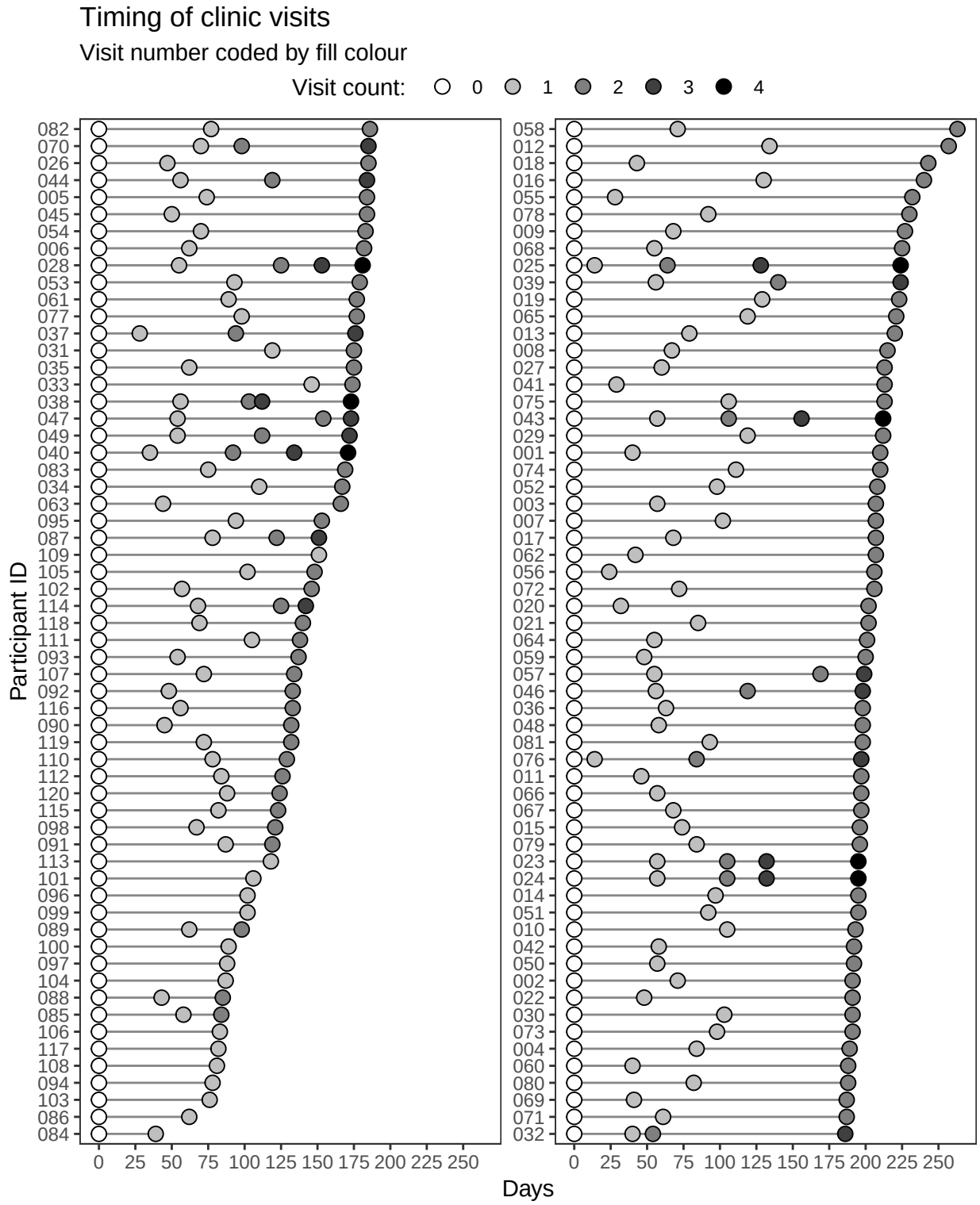
```r
# Data structure
glimpse(data)
```

```
## Observations: 371
## Variables: 6
## $ ID            <chr> "001", "001", "001", "002", "002", "002", "003",...
## $ visit_number  <int> 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, ...
## $ visit_day     <int> 0, 40, 210, 0, 71, 191, 0, 57, 207, 0, 84, 189, ...
## $ visit_months  <dbl> 0, 1, 7, 0, 2, 6, 0, 2, 7, 0, 3, 6, 0, 2, 6, 0, ...
## $ hivsn_present <fct> no, no, no, no, no, no, no, no, no, no, no, no, ...
## $ sn            <fct> no, no, no, no, no, no, no, no, no, no, no, no, ...
```

Clinic visits

Whole cohort

```
data %>%
    # Process data for plotting
    group_by(ID) %>%
    mutate(visit = ifelse(visit_day > 0,
                          yes = 1,
                          no = 0),
           visit_count = cumsum(visit),
           max_visits = max(visit_count),
           max_duration = max(visit_day)) %>%
    ungroup() %>%
    select(ID, visit_day, visit_months,
           visit_count, max_visits,
           max_duration) %>%
    # Clean-up and order dataframe
    arrange(max_duration, max_visits, desc(ID)) %>%
    mutate(ID = as_factor(ID),
           visit_count = as.character(visit_count)) %>%
    complete(ID, visit_months) %>%
    # Add a dummy variable for faceting
    bind_cols(facet = c(rep('Panel 1', 600), rep('Panel 2', 1202-600))) %>%
    # Clean-up and order dataframe
    filter(complete.cases(.)) %>%
    # Plot
    ggplot(data = .) +
    aes(x = visit_day,
        y = ID) +
    geom_path(colour = '#888888') +
    geom_point(aes(fill = visit_count),
               shape = 21,
               size = 3) +
    scale_x_continuous(breaks = seq(0, 250, 25)) +
    scale_fill_manual(name = 'Visit count: ', values = grey_pal) +
    labs(title = 'Timing of clinic visits',
         subtitle = 'Visit number coded by fill colour',
         y = 'Participant ID',
         x = 'Days') +
    facet_wrap(~facet,
               ncol = 2,
               scales = 'free_y') +
    theme(legend.position = 'top',
          legend.margin = margin(t = -0.3,
                                 r = 0,
                                 b = -0.3,
                                 l = 0, 'lines'),
          panel.grid = element_blank(),
          strip.background = element_blank(),
          strip.text = element_blank())
```

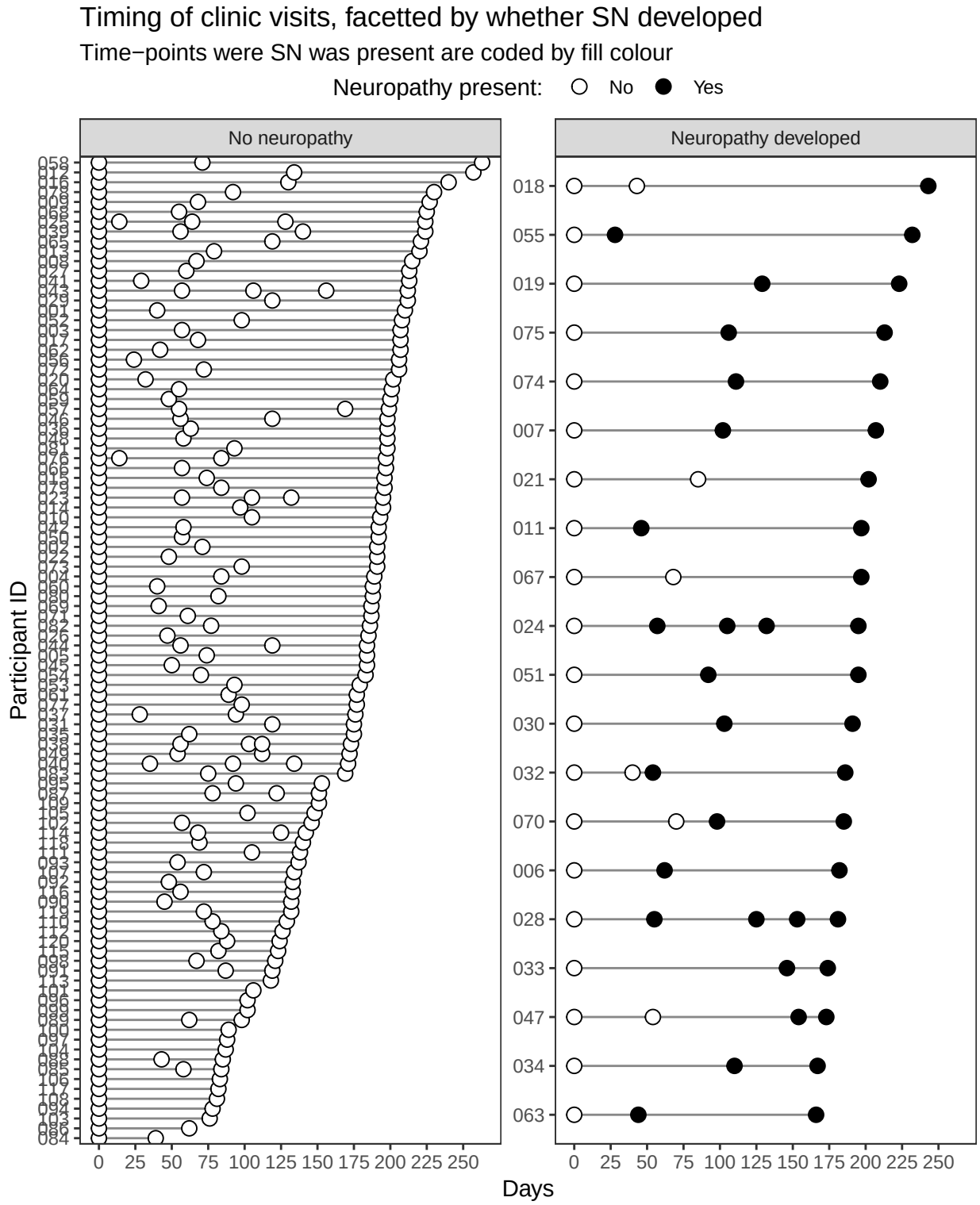


By SN status

```
data_plot <- data %>%
    # Process data for plotting
    group_by(ID) %>%
```

```
    mutate(visit = ifelse(visit_day > 0,
                          yes = 1,
                          no = 0),
           visit_count = cumsum(visit),
           max_visits = max(visit_count),
           max_duration = max(visit_day)) %>%
    ungroup() %>%
    select(ID, hivsn_present, visit_day, visit_months,
           visit_count, max_visits,
           max_duration, sn) %>%
    # Clean-up and order dataframe
    arrange(max_duration, max_visits, desc(ID)) %>%
    mutate(ID = as_factor(ID),
           visit_count = as.character(visit_count)) %>%
    complete(ID, visit_months) %>%
    # Clean-up and order dataframe
    filter(complete.cases(.)) %>%
    # Recode facetting column
    mutate(sn = as.character(sn),
           sn = ifelse(sn == 'yes',
                       yes = 'Neuropathy developed',
                       no = 'No neuropathy'),
           sn = factor(sn,
                       levels = c('No neuropathy', 'Neuropathy developed'),
                       ordered = TRUE)) %>%
    mutate(hivsn_present = str_to_title(hivsn_present))

# Plot
ggplot(data = data_plot) +
    aes(x = visit_day,
        y = ID) +
    geom_path(colour = '#888888') +
    geom_point(aes(fill = hivsn_present),
               shape = 21,
               size = 3) +
    labs(title = 'Timing of clinic visits, facetted by whether SN developed',
         subtitle = 'Time-points were SN was present are coded by fill colour',
         x = 'Days',
         y = 'Participant ID') +
    scale_x_continuous(breaks = seq(0, 250, 25)) +
    scale_fill_manual(name = 'Neuropathy present: ',
                      values = c('#FFFFFF', '#000000')) +
    facet_wrap(~ sn,
               ncol = 2,
               scales = 'free_y') +
    theme(legend.position = 'top',
          legend.margin = margin(t = -0.3,
                                 r = 0,
                                 b = -0.3,
                                 l = 0, 'lines'),
          panel.grid = element_blank())
```




```
# Publication plot
gg_plot <- data_plot %>%
    ggplot(data = .) +
    aes(x = visit_day,
        y = ID) +
    geom_path(colour = '#888888') +
```

```
    geom_point(aes(fill = hivsn_present),
               shape = 21,
               size = 3) +
    labs(x = 'Days',
         y = 'Participant ID') +
    scale_x_continuous(breaks = seq(0, 250, 25)) +
    scale_fill_manual(name = 'Neuropathy present: ',
                      values = c('#FFFFFF', '#000000')) +
    facet_wrap(~ sn,
               ncol = 2,
               scales = 'free_y') +
    theme(legend.position = 'top',
          legend.text = element_text(size = 11),
          legend.title = element_text(size = 11),
          axis.text.y = element_text(colour = '#000000'),
          axis.text.x = element_text(colour = '#000000',
                                     size = 12),
          axis.title = element_text(size = 14),
          panel.grid = element_blank(),
          panel.border = element_rect(size = 1),
          strip.text = element_text(size = 12))

ggsave(filename = 'figures/follow-up.png',
       plot = gg_plot,
       width = 9,
       height = 11)
```

Analysis

Process data

```
# Generate visit data
data_v1 <- data %>%
    # Calculations per individual
    group_by(ID) %>%
    mutate(# Extract the max number of visits
           max_visits = max(visit_number),
           # Time from visit 1 to the last visit
           max_duration = max(visit_day),
           # Time between successive visits
           visit_break = visit_day - lag(visit_day),
           # Running mean time (days) between visits
           cummulative_mean = round(cummean(visit_day))) %>%
    ungroup() %>%
    # Select required columns
    select(ID, visit_number, visit_day, visit_break, max_visits,
           max_duration, cummulative_mean) %>%
    left_join(data_sn)

# Filter based on maximum number of visits
data_v2 <- data_v1 %>%
```

```
    # Per individual
    group_by(ID) %>%
    # Get maximum number of visits
    filter(visit_number == max_visits) %>%
    ungroup()
```

Number of clinic visits

```
# Summary table
data_v2 %>%
    summarise(n = n(),
              Median.visits = median(max_visits),
              Q25 = quantile(max_visits, 0.25),
              Q75 = quantile(max_visits, 0.75),
              min = min(max_visits),
              max = max(max_visits))
```

```
## # A tibble: 1 x 6
##       n Median.visits   Q25   Q75   min   max
##   <int>         <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   120             3     3     3     2     5
```

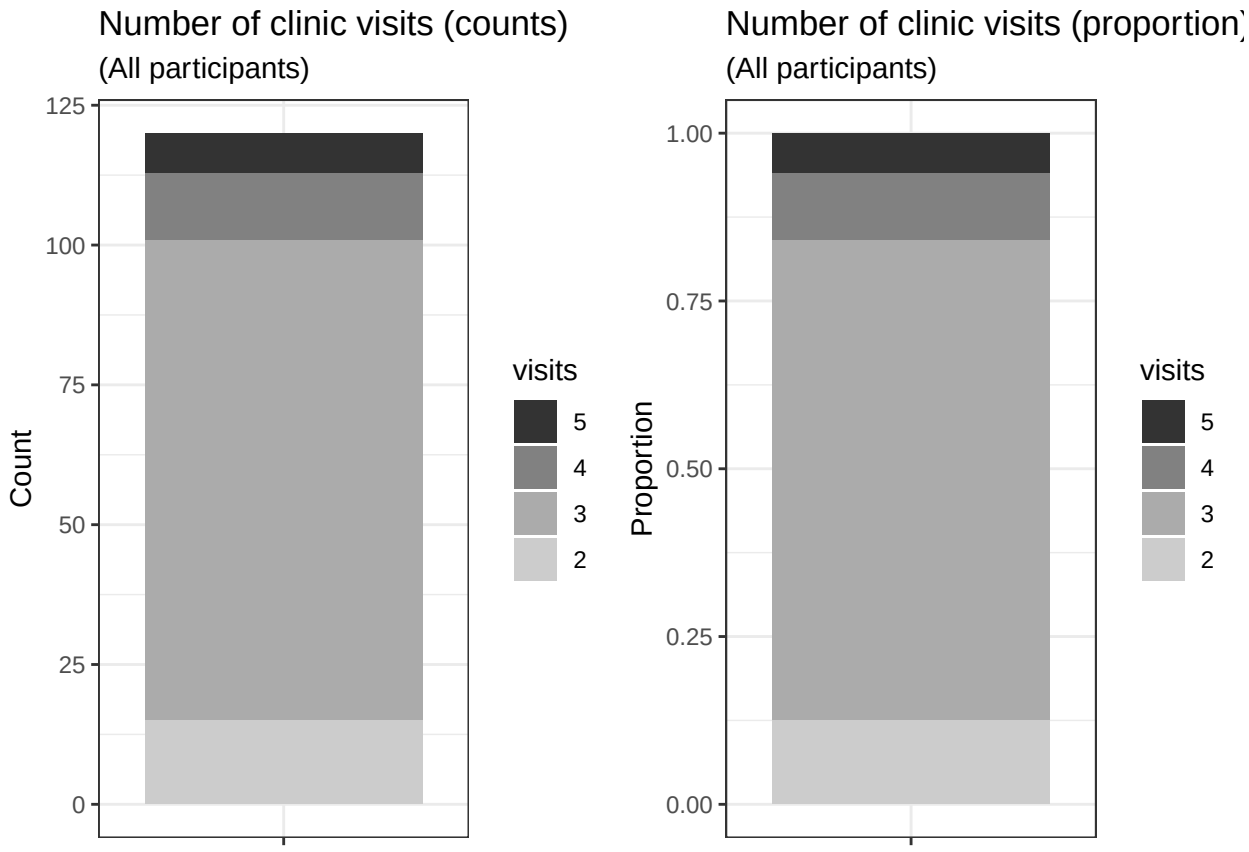
```
# Summary table (conditioned on sn)
data_v2 %>%
    group_by(sn) %>%
    summarise(n = n(),
              Median.visits = median(max_visits),
              Q25 = quantile(max_visits, 0.25),
              Q75 = quantile(max_visits, 0.75),
              min = min(max_visits),
              max = max(max_visits))
```

```
## # A tibble: 2 x 7
##   sn        n Median.visits   Q25   Q75   min   max
##   <fct> <int>         <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 no      100             3     3  3        2     5
## 2 yes      20             3     3  3.25     3     5
```

```
# Plots
gg_v2a <- data_v2 %>%
    group_by(max_visits) %>%
    summarise(count = n()) %>%
    mutate(visits = fct_relevel(factor(max_visits), '5', '4', '3')) %>%
    ggplot(data = .) +
    aes(y = count,
        x = 'all',
        fill = visits) +
    geom_bar(stat = 'identity') +
    scale_fill_grey() +
    labs(title = 'Number of clinic visits (counts)',
         subtitle = '(All participants)',
         y = 'Count') +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank())
```

```
gg_v2b <- data_v2 %>%
    group_by(max_visits) %>%
    summarise(count = n()) %>%
    mutate(visits = fct_relevel(factor(max_visits), '5', '4', '3')) %>%
    ggplot(data = .) +
    aes(y = count,
        x = 'all',
        fill = visits) +
    geom_bar(stat = 'identity',
             position = position_fill()) +
    scale_fill_grey() +
    labs(title = 'Number of clinic visits (proportion)',
         subtitle = '(All participants)',
         y = 'Proportion') +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank())

# Plot next to each other using patchwork package
gg_v2a + gg_v2b
```

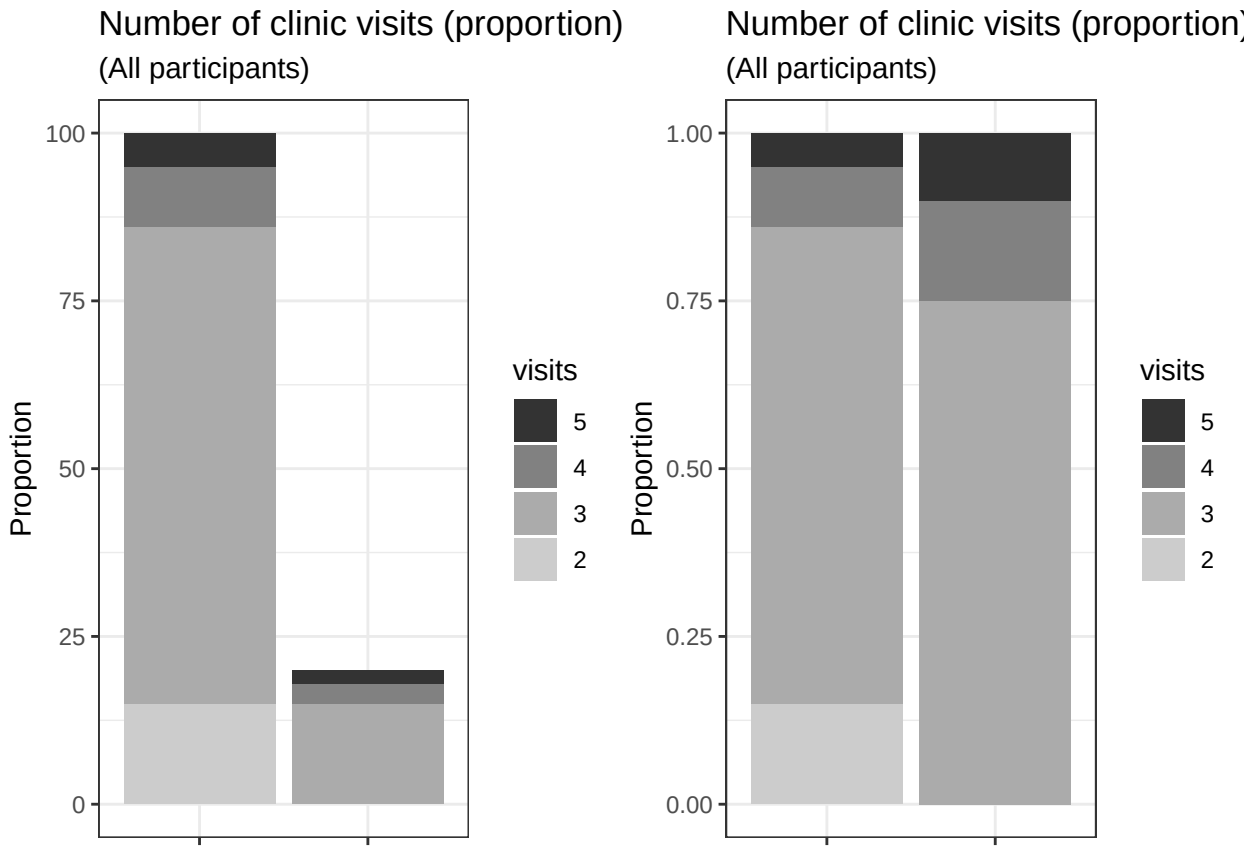



```
## Conditional on SN
gg_v2c <- data_v2 %>%
    group_by(sn, max_visits) %>%
    summarise(count = n()) %>%
    mutate(visits = factor(max_visits,
                           levels = c('5', '4', '3', '2'),
                           ordered = TRUE)) %>%
```

```
    ggplot(data = .) +
    aes(y = count,
        x = sn,
        fill = visits) +
    geom_bar(stat = 'identity') +
    scale_fill_grey() +
    labs(title = 'Number of clinic visits (proportion)',
         subtitle = '(All participants)',
         y = 'Proportion') +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank())

gg_v2d <- data_v2 %>%
    group_by(sn, max_visits) %>%
    summarise(count = n()) %>%
    mutate(visits = factor(max_visits,
                           levels = c('5', '4', '3', '2'),
                           ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(y = count,
        x = sn,
        fill = visits) +
    geom_bar(stat = 'identity',
             position = position_fill()) +
    scale_fill_grey() +
    labs(title = 'Number of clinic visits (proportion)',
         subtitle = '(All participants)',
         y = 'Proportion') +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank())

# Plot next to each other using patchwork package
gg_v2c + gg_v2d
```




```
# Stats
chisq_test(xtabs(~ sn + visit_number,
                 data = data_v2),
           distribution = approximate(nresample =10000))
```

```
##
##  Approximative Pearson Chi-Squared Test
##
## data:  visit_number by sn (no, yes)
## chi-squared = 4.3515, p-value = 0.225
```

Time between first and last visit

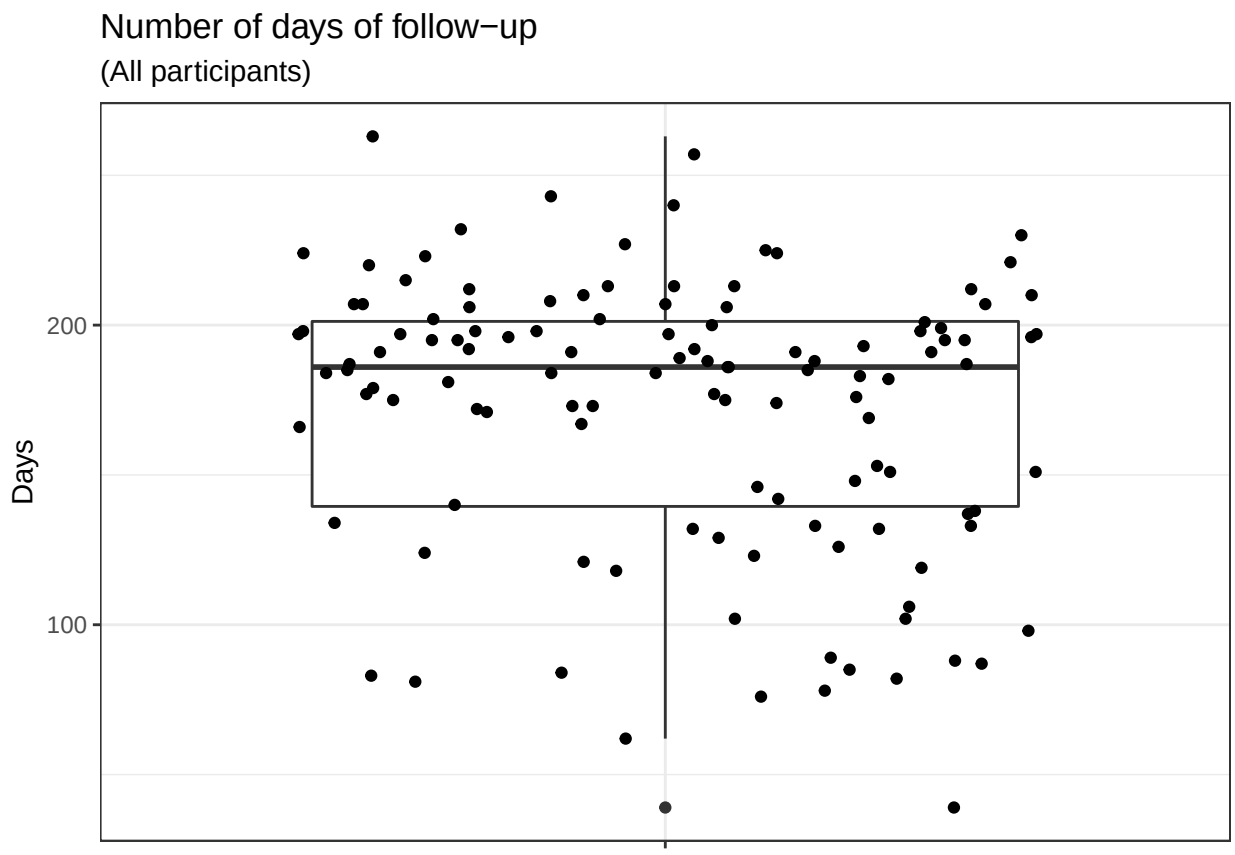
```
# Summary table
data_v2 %>%
    summarise(n = n(),
              Median.days = median(max_duration),
              Q25 = quantile(max_duration, 0.25),
              Q75 = quantile(max_duration, 0.75),
              min = min(max_duration),
              max = max(max_duration))
```

```
## # A tibble: 1 x 6
##       n Median.days   Q25   Q75   min   max
##   <int>       <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   120         186  140.  201.    39   263
```

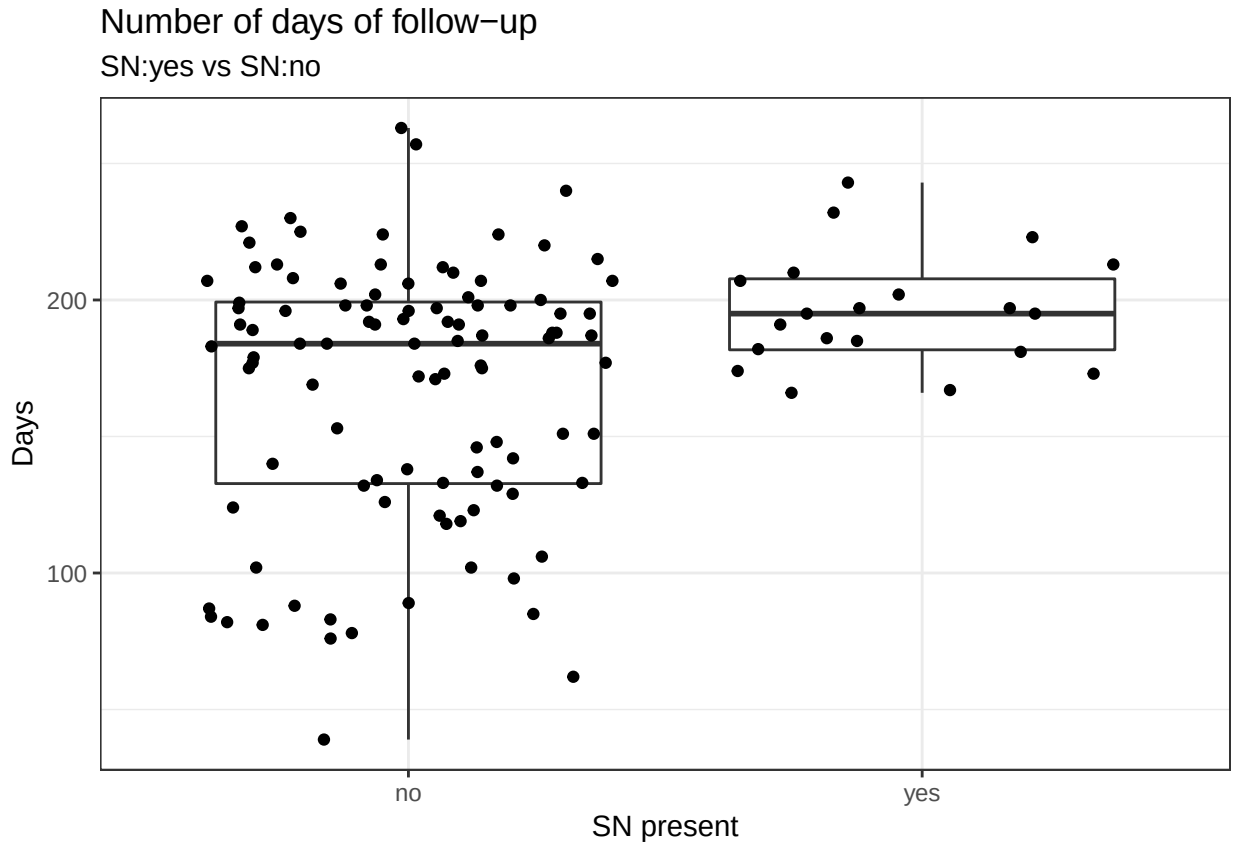
```r
# Summary table (conditioned on sn)
data_v2 %>%
    group_by(sn) %>%
    summarise(n = n(),
              Median.days = median(max_duration),
              Q25 = quantile(max_duration, 0.25),
              Q75 = quantile(max_duration, 0.75),
              min = min(max_duration),
              max = max(max_duration))
```

```
## # A tibble: 2 x 7
##   sn        n Median.days   Q25   Q75   min   max
##   <fct> <int>       <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 no      100         184  133.  199.    39   263
## 2 yes      20         195  182.  208.   166   243
```

```r
# Plots
data_v2 %>%
    ggplot(data = .) +
    aes(y = max_duration,
        x = 'All patients') +
    geom_boxplot() +
    geom_jitter(height = 0) +
    labs(title = 'Number of days of follow-up',
         subtitle = '(All participants)',
         y = 'Days') +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank())
```



```
## Conditional on SN
data_v2 %>%
    ggplot(data = .) +
    aes(y = max_duration,
        x = sn) +
    geom_boxplot() +
    geom_jitter(height = 0) +
    labs(title = 'Number of days of follow-up',
         subtitle = 'SN:yes vs SN:no',
         y = 'Days',
         x = 'SN present')
```




```
# Stats
normal_test(max_duration ~ factor(sn),
            data = data_v2,
            distribution = approximate(nresample =10000))
```

```
##
##  Approximative Two-Sample van der Waerden (Normal Quantile) Test
##
## data:  max_duration by factor(sn) (no, yes)
## Z = -2.103, p-value = 0.0334
## alternative hypothesis: true mu is not equal to 0
```

Time between successive clinic visits

```
# Summary table
data_v1 %>%
    filter(visit_number != 1) %>%
    group_by(visit_number) %>%
    summarise(n = n(),
              Mean.days = mean(visit_break),
              Median.days = median(visit_break),
              SD = sd(visit_break),
              Q25 = quantile(visit_break, 0.25),
              Q75 = quantile(visit_break, 0.75),
              min = min(visit_break),
              max = max(visit_break))
```

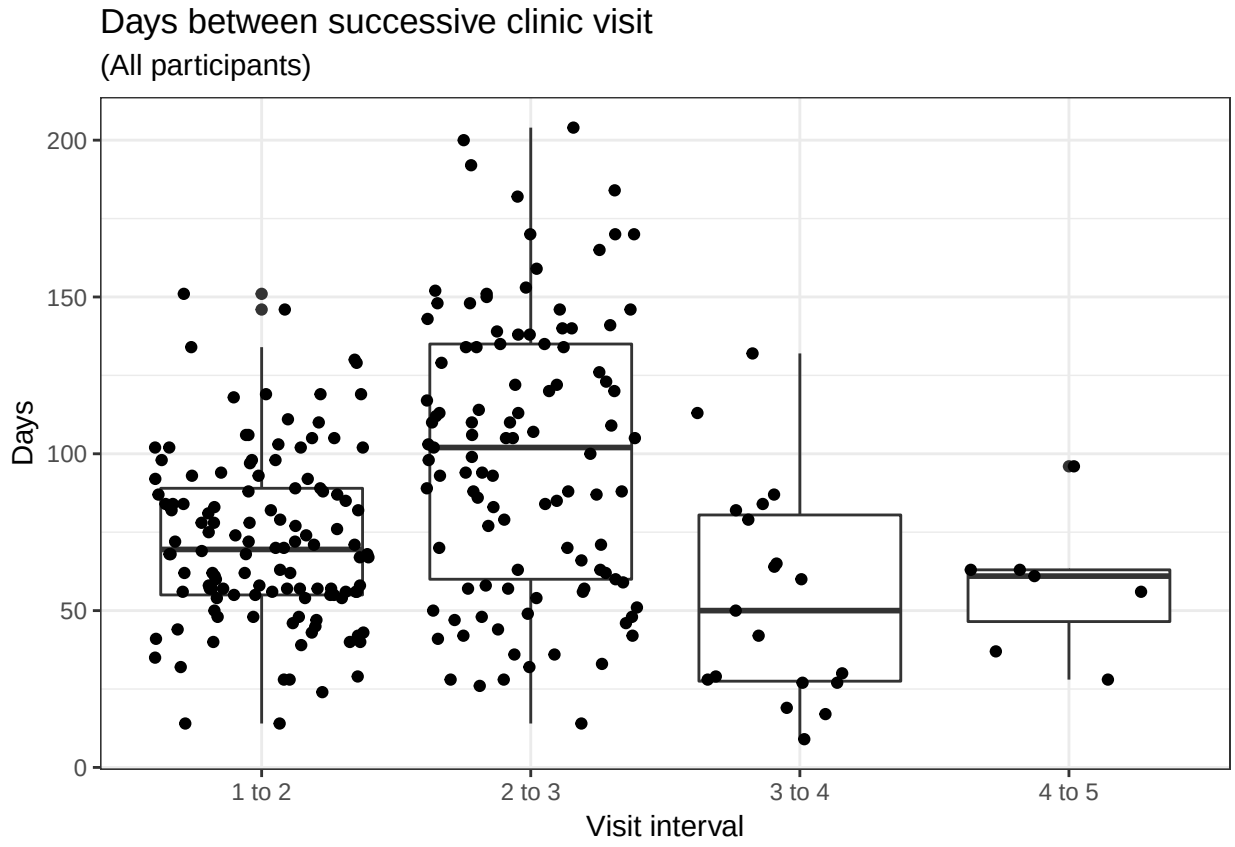
```
## # A tibble: 4 x 9
##   visit_number     n Mean.days Median.days    SD   Q25   Q75   min   max
##          <int> <int>     <dbl>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1            2   120      72.4        69.5  27.1  55    89      14   151
## 2            3   105     100.        102    44.4  60   135      14   204
## 3            4    19      54.9        50    34.4  27.5  80.5     9   132
## 4            5     7      57.7        61    21.8  46.5  63      28    96
```

```
# Summary table (conditioned on sn)
data_v1 %>%
    filter(visit_number != 1) %>%
    group_by(visit_number, sn) %>%
    summarise(n = n(),
              Mean.days = mean(visit_break),
              Median.days = median(visit_break),
              SD = sd(visit_break),
              Q25 = quantile(visit_break, 0.25),
              Q75 = quantile(visit_break, 0.75),
              min = min(visit_break),
              max = max(visit_break))
```

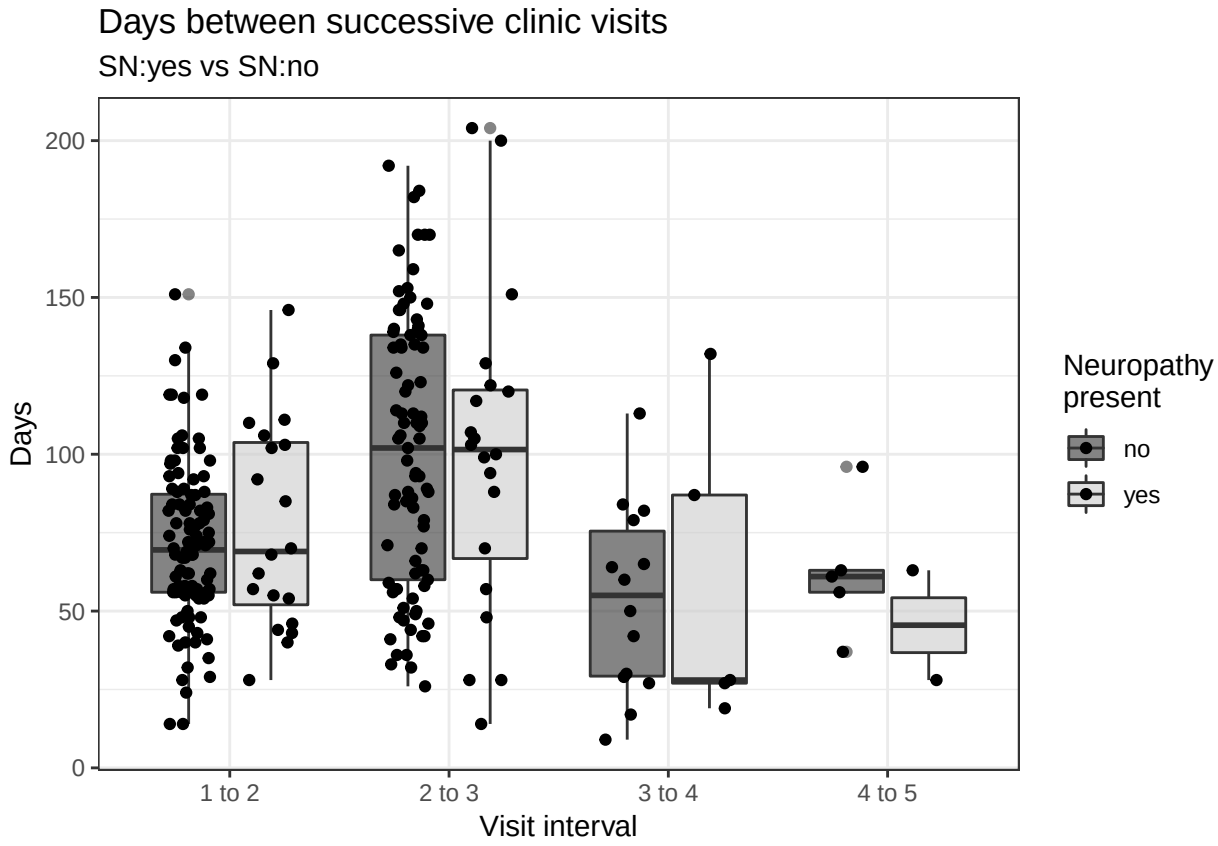
```
## # A tibble: 8 x 10
## # Groups:   visit_number [4]
##   visit_number sn        n Mean.days Median.days    SD   Q25   Q75   min
##          <int> <fct> <int>     <dbl>       <dbl> <dbl> <dbl> <dbl> <dbl>
## 1            2 no      100      71.4        69.5  25.8  56    87.2    14
## 2            2 yes      20      77.6        69    33.1  52   104.     28
## 3            3 no       85     100.        102    43.1  60   138      26
## 4            3 yes      20      99.2       102.   50.7  66.8 120.     14
## 5            4 no       14      53.6        55    29.8  29.2  75.5     9
## 6            4 yes       5      58.6        28    49.2  27    87      19
## 7            5 no        5      62.6        61    21.3  56    63      37
## 8            5 yes       2      45.5        45.5  24.7  36.8  54.2    28
## # ... with 1 more variable: max <dbl>
```

```
# Plots
data_v1 %>%
    filter(visit_number != 1) %>%
    ggplot(data = .) +
    aes(y = visit_break,
        x = factor(visit_number)) +
```

```
    geom_boxplot() +
    geom_jitter(height = 0) +
    scale_x_discrete(labels = c('1 to 2', '2 to 3',
                                '3 to 4', '4 to 5')) +
    labs(title = 'Days between successive clinic visit ',
         subtitle = '(All participants)',
         y = 'Days',
         x = 'Visit interval')
```




```
## Conditional on SN
data_v1 %>%
    filter(visit_number != 1) %>%
    ggplot(data = .) +
    aes(y = visit_break,
        x = factor(visit_number),
        fill = sn) +
    geom_boxplot(alpha = 0.6) +
    geom_point(position = position_jitterdodge(jitter.height = 0)) +
    scale_x_discrete(labels = c('1 to 2', '2 to 3',
                                '3 to 4', '4 to 5')) +
    scale_fill_grey(name = 'Neuropathy\npresent') +
    labs(title = 'Days between successive clinic visits',
         subtitle = 'SN:yes vs SN:no',
         y = 'Days',
         x = 'Visit interval')
```



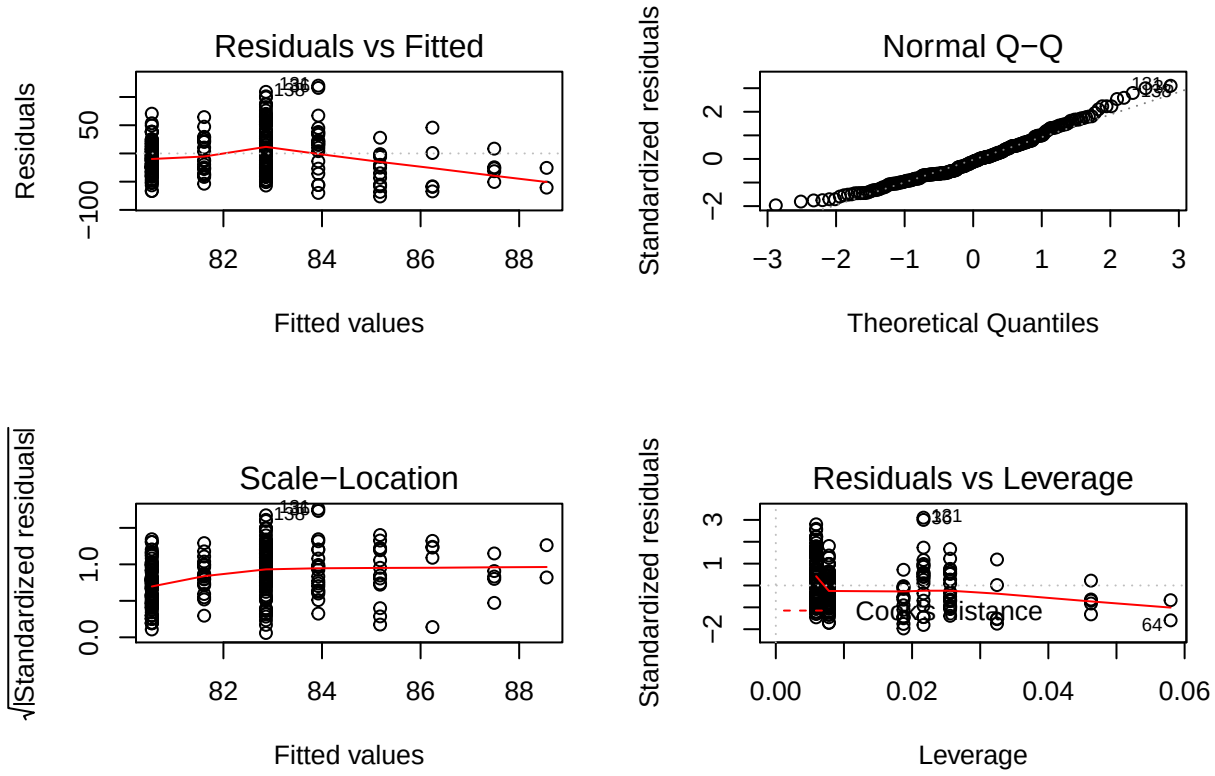

```
# Stats
## Generate data
data_lmer <- data_v1 %>%
    filter(visit_number != 1) %>%
    select(ID, visit_number, visit_break, sn)

## Generate model (ID included as random effects)
mod <- lm(visit_break ~ sn + visit_number,
            data = data_lmer)

## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
Anova(mod = mod,
      type = 'II',
      test.statistic = 'F')
```

```
## Anova Table (Type II tests)
##
## Response: visit_break
##              Sum Sq  Df F value Pr(>F)
## sn               43   1  0.0280 0.8673
## visit_number    730   1  0.4767 0.4906
## Residuals    379751 248
```

```
## Basic diagnostics
par(mfrow = c(2, 2))
plot(mod)
```




```
par(mfrow = c(1, 1))
```

Session information

```
sessionInfo()
```

```
## 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     coin_1.3-0
##  [4] survival_2.44-1.1 patchwork_0.0.1   forcats_0.4.0
##  [7] stringr_1.4.0     dplyr_0.8.0.1     purrr_0.3.2
## [10] readr_1.3.1       tidyr_0.8.3       tibble_2.1.1
## [13] ggplot2_3.1.1     tidyverse_1.2.1   magrittr_1.5
##
```

```
## loaded via a namespace (and not attached):
##  [1] httr_1.4.0         jsonlite_1.6       splines_3.6.0
##  [4] modelr_0.1.4       assertthat_0.2.1   stats4_3.6.0
##  [7] cellranger_1.1.0   yaml_2.2.0         pillar_1.3.1
## [10] backports_1.1.4    lattice_0.20-38    glue_1.3.1
## [13] digest_0.6.18      rvest_0.3.3        colorspace_1.4-1
## [16] sandwich_2.5-1     htmltools_0.3.6    Matrix_1.2-17
## [19] plyr_1.8.4         pkgconfig_2.0.2    broom_0.5.2
## [22] haven_2.1.0        mvtnorm_1.0-10     scales_1.0.0
## [25] openxlsx_4.1.0     rio_0.5.16         generics_0.0.2
## [28] ellipsis_0.1.0     TH.data_1.0-10     withr_2.1.2.9000
## [31] lazyeval_0.2.2     cli_1.1.0          crayon_1.3.4
## [34] readxl_1.3.1       evaluate_0.13      fansi_0.4.0
## [37] nlme_3.1-139       MASS_7.3-51.4      xml2_1.2.0
## [40] foreign_0.8-71     tools_3.6.0        data.table_1.12.2
## [43] hms_0.4.2          matrixStats_0.54.0 multcomp_1.4-10
## [46] munsell_0.5.0      zip_2.0.1          compiler_3.6.0
## [49] rlang_0.3.4        grid_3.6.0         rstudioapi_0.10
## [52] labeling_0.3       rmarkdown_1.12     gtable_0.3.0
## [55] codetools_0.2-16   abind_1.4-5        curl_3.3
## [58] R6_2.4.0           zoo_1.8-5          lubridate_1.7.4
## [61] knitr_1.22         utf8_1.1.4         libcoin_1.0-4
## [64] modeltools_0.2-22  stringi_1.4.3      parallel_3.6.0
## [67] Rcpp_1.0.1         tidyselect_0.2.5   xfun_0.6
```