Supplement 2

Sites of pain

Peter Kamerman

Last knitted: 03 March 2021

Contents

---

1 Import and check data

```
# Import
data <- read_rds('data-cleaned/data-pain-sites.rds')
demo <- read_rds('data-cleaned/data-demographics.rds')

# Check
## Pain sites
dim(data)
```

```
## [1] 596  21
```

```
names(data)
```

```
##  [1] "ID"              "Head"            "Throat"
##  [4] "Shoulder"        "Arms"            "Elbows"
##  [7] "Wrists.Hands"    "Chest"           "Upper_back"
## [10] "Lower_back"      "Abdomen"         "Cervical_spine"
## [13] "Thoracic_spine"  "Lumbosacral_spine" "Groin"
## [16] "Hips"            "Legs"            "Knees"
## [19] "Ankles.Feet"     "Buttocks"        "Site"
```

```
glimpse(data)
```

```
## Rows: 596
## Columns: 21
## $ ID               <chr> "RPB73", "RPB74", "RPB75", "RPB76", "RPB77", "RPB...
## $ Head             <chr> "No", "No", "No", "Yes", "Yes", "No", "No", "No",...
## $ Throat           <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Shoulder         <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Arms             <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Elbows           <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Wrists.Hands     <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Chest            <chr> "No", "No", "No", "Yes", "No", "No", "No", "No", ...
## $ Upper_back       <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Lower_back       <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Abdomen          <chr> "No", "No", "Yes", "Yes", "No", "No", "Yes", "No"...
## $ Cervical_spine   <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Thoracic_spine   <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Lumbosacral_spine <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Groin            <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Hips             <chr> "No", "No", "No", "No", "No", "No", "No", "Yes", ...
```

```
## $ Legs              <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes",...
## $ Knees             <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes",...
## $ Ankles.Feet       <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes",...
## $ Buttocks          <chr> "No", "No", "No", "No", "No", "No", "No", "Yes", ...
## $ Site              <chr> "RP", "RP", "RP", "RP", "RP", "RP", "RP", "RP", "...
## Demographics
dim(demo)
```

```
## [1] 596    8
```

```
names(demo)
```

```
## [1] "ID"            "Site"            "Sex"
## [4] "Age"           "Employment_status" "CD4_recent"
## [7] "ART_currently" "Education"
```

```
glimpse(demo)
```

```
## Rows: 596
## Columns: 8
## $ ID                <chr> "RPB73", "RPB74", "RPB75", "RPB76", "RPB77", "RPB...
## $ Site              <chr> "RP", "RP", "RP", "RP", "RP", "RP", "RP", "RP", "...
## $ Sex               <chr> "Female", "Female", "Female", "Female", "Female",...
## $ Age               <dbl> 36, 27, 39, 36, 31, 32, 28, 37, 31, 25, 31, 24, 3...
## $ Employment_status <chr> "Other", "Unemployed", "Other", "Unemployed", "Un...
## $ CD4_recent        <dbl> 391, 571, 591, 207, 126, 225, 543, 410, 74, 212, ...
## $ ART_currently     <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", ...
## $ Education         <chr> "Tertiary", "Secondary", "Secondary", "Primary", ...
```

2 Basic descriptive statistics

2.1 Pain sites

```
data %>%
    select(-ID, -Site) %>%
    mutate_if(is.character, factor) %>%
    skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	596
Number of columns	19
Column type frequency:	
factor	19
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	n_unique	top_counts
Head	0	1	2	No: 401, Yes: 195
Throat	0	1	2	No: 577, Yes: 19
Shoulder	0	1	2	No: 549, Yes: 47
Arms	0	1	2	No: 571, Yes: 25
Elbows	0	1	2	No: 574, Yes: 22
Wrists.Hands	0	1	2	No: 561, Yes: 35
Chest	0	1	2	No: 479, Yes: 117
Upper_back	0	1	1	No: 596
Lower_back	0	1	2	No: 554, Yes: 42
Abdomen	0	1	2	No: 437, Yes: 159
Cervical_spine	0	1	2	No: 566, Yes: 30
Thoracic_spine	0	1	2	No: 522, Yes: 74
Lumbosacral_spine	0	1	2	No: 504, Yes: 92
Groin	0	1	2	No: 542, Yes: 54
Hips	0	1	2	No: 556, Yes: 40
Legs	0	1	2	No: 497, Yes: 99
Knees	0	1	2	No: 512, Yes: 84
Ankles.Feet	0	1	2	No: 412, Yes: 184
Buttocks	0	1	2	No: 577, Yes: 19

2.2 Demographics

```
demo %>%
    select(-ID, -Site) %>%
    mutate_if(is.character, factor) %>%
    skim()
```

Table 3: Data summary

Name	Piped data
Number of rows	596
Number of columns	6
Column type frequency:	
factor	4
numeric	2
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	n_unique	top_counts
Sex	0	1.00	2	Fem: 481, Mal: 115
Employment_status	49	0.92	4	Une: 330, Ful: 131, Par: 52, Oth: 34
ART_currently	5	0.99	2	Yes: 460, No: 131
Education	37	0.94	3	Sec: 395, Pri: 99, Ter: 65

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Age	8	0.99	37.28	9.06	19	31	36	42	76
CD4_recent	99	0.83	320.71	238.92	1	155	261	432	1232

2.3   Add acute/chronic pain column to demographic data

```
# Based on pain definition used in the study
## Acute assessing pain in the last 1 week or 1 month.
## Chronic assessing persistent pain in the last 3 months
demo <- demo %>%
    mutate(Pain_def = case_when(
        Site == 'RP' |
            Site == 'NM' ~ 'Acute',
        TRUE ~ 'Chronic'
    ))

# Tabulate
demo %>%
    select(Pain_def) %>%
    mutate(Pain_def = factor(Pain_def)) %>%
    skim()
```

Table 6: Data summary

Name	Piped data
Number of rows	596
Number of columns	1
Column type frequency:	
factor	1
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	n_unique	top_counts
Pain_def	0	1	2	Acu: 387, Chr: 209

2.4   Boostrap functions

```
# Proportion
prop_func <- function(d, i){
    dat <- d[i, ]
    dat_vec <- dat[[1]]
    dat_prop <- mean(dat_vec == 'Yes', na.rm = TRUE)
    dat_prop
}

# Median
median_func <- function(d, i){
```

```
    dat <- d[i, ]
    dat_vec <- dat[[1]]
    dat_median <- median(dat_vec, na.rm = TRUE)
    dat_median
}
```

---

3 Proportion point estimates with 95% CIs

3.1 Process data

```r
# Set seed
set.seed(2020)

# Remove ID and upper_back (only one outcome -- no pain) columns
prop <- data[, !(names(data) %in% c('ID', 'Upper_back'))]

# Bootstrap CIs
prop_boot <- prop %>%
    # Remove Site column
    select(-Site) %>%
    # Pivot to long format
    pivot_longer(cols = everything(),
                 names_to = 'body_site',
                 values_to = 'pain_present') %>%
    # Add body regions
    mutate(region = case_when(
        body_site == 'Chest' |
            body_site == 'Head' |
            body_site == 'Throat' |
            body_site == 'Shoulder' ~ 'Head and upper torso',
        body_site == 'Lower_back' |
            body_site == 'Abdomen' |
            body_site == 'Hips' |
            body_site == 'Buttocks' |
            body_site == 'Groin' ~ 'Lower torso',
        body_site == 'Legs' |
            body_site == 'Knees' |
            body_site == 'Ankles.Feet' ~ 'Lower limbs',
        body_site == 'Arms' |
            body_site == 'Elbows' |
            body_site == 'Wrists.Hands' ~ 'Upper limbs',
        body_site == 'Cervical_spine' |
            body_site == 'Thoracic_spine' |
            body_site == 'Lumbosacral_spine' ~ 'Spinal column',
        TRUE ~ 'other'
    )) %>%
    # Nest by body region and body site
    group_by(region, body_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
```

```r
                         ~ boot(data = .x,
                                statistic = prop_func,
                                R = 999,
                                stype = 'i',
                                parallel = 'multicore',
                                ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-data, -boot, -ci) %>%
    # Unnest
    unnest(cols = c(point_est, lower_ci, upper_ci))

# Re-nest by body region and generate figures and tables
prop_boot2 <- prop_boot %>%
    group_by(region) %>%
    nest() %>%
    # Fix site labels
    mutate(data = map(.x = data,
                      ~ .x %>%
                          mutate(body_site = str_replace_all(body_site,
                                                             pattern = '_',
                                                             replacement = ' '),
                                 body_site = str_replace_all(body_site,
                                                             pattern = '\\.',
                                                             replacement = ' & '),
                                 body_site = str_replace_all(body_site,
                                                             pattern = 'Lower back',
                                                             replacement = 'Flank')))) %>%
    # Re-order sites by point_est
    mutate(data = map(.x = data,
                      ~ .x %>%
                          mutate(body_site = fct_reorder(body_site,
                                                         point_est)))) %>%
    # Plot data
    mutate(plots = map2(.x = data,
                        .y = region,
                        ~ .x %>%
                            ggplot(data = .) +
                            aes(x = body_site,
                                y = point_est,
                                ymin = lower_ci,
                                ymax = upper_ci) +
                            geom_hline(yintercept = 0.1,
                                       linetype = 2) +
```

```
                          geom_pointrange(size = 1) +
                          coord_flip() +
                           labs(title = .y,
                                subtitle = '(Point estimate with 95%CI)',
                                y = 'Proportion with pain') +
                          scale_y_continuous(limits = c(0, 0.4)) +
                          theme_minimal(base_size = 18) +
                          theme(plot.title = element_text(size = 18),
                                plot.subtitle = element_text(size = 12),
                                axis.title.y = element_blank(),
                                panel.grid = element_blank(),
                                axis.text = element_text(colour = '#000000'),
                                axis.line = element_line(size = 0.5),
                                axis.ticks = element_line(size = 0.5)))) %>%
    # Tabulate data
    mutate(tables = map2(.x = data,
                          .y = region,
                          ~ .x %>%
                            kable(caption = .y,
                                  digits = 2)))
```

3.2   Tabulated proportions (with 95% CIs), by body region

```
walk(prop_boot2$tables, ~ print(.x))
```

Table 8: Head and upper torso

body_site	point_est	lower_ci	upper_ci
Head	0.33	0.29	0.37
Throat	0.03	0.02	0.05
Shoulder	0.08	0.06	0.10
Chest	0.20	0.17	0.23

Table 9: Upper limbs

body_site	point_est	lower_ci	upper_ci
Arms	0.04	0.03	0.06
Elbows	0.04	0.02	0.05
Wrists & Hands	0.06	0.04	0.08

Table 10: Lower torso

body_site	point_est	lower_ci	upper_ci
Flank	0.07	0.05	0.09
Abdomen	0.27	0.23	0.30
Groin	0.09	0.07	0.11
Hips	0.07	0.05	0.09
Buttocks	0.03	0.02	0.05

Table 11: Spinal column

body_site	point_est	lower_ci	upper_ci
Cervical spine	0.05	0.03	0.07
Thoracic spine	0.12	0.10	0.15
Lumbosacral spine	0.15	0.13	0.18

Table 12: Lower limbs

body_site	point_est	lower_ci	upper_ci
Legs	0.17	0.14	0.20
Knees	0.14	0.11	0.17
Ankles & Feet	0.31	0.27	0.35

3.3 Plotted proportions (with 95% CIs), by body region

```r
upper <- prop_boot2$plots[[1]] +
    theme(axis.title.x = element_blank())

arm <- prop_boot2$plots[[2]] +
    theme(axis.title.x = element_blank())

lower <- prop_boot2$plots[[3]] +
    theme(axis.title.x = element_blank())

spine <- prop_boot2$plots[[4]]

leg <- prop_boot2$plots[[5]]

# Patchwork
plot_prop <- upper + lower + arm + leg + spine +
    plot_layout(ncol = 2)

# Output
plot_prop
```
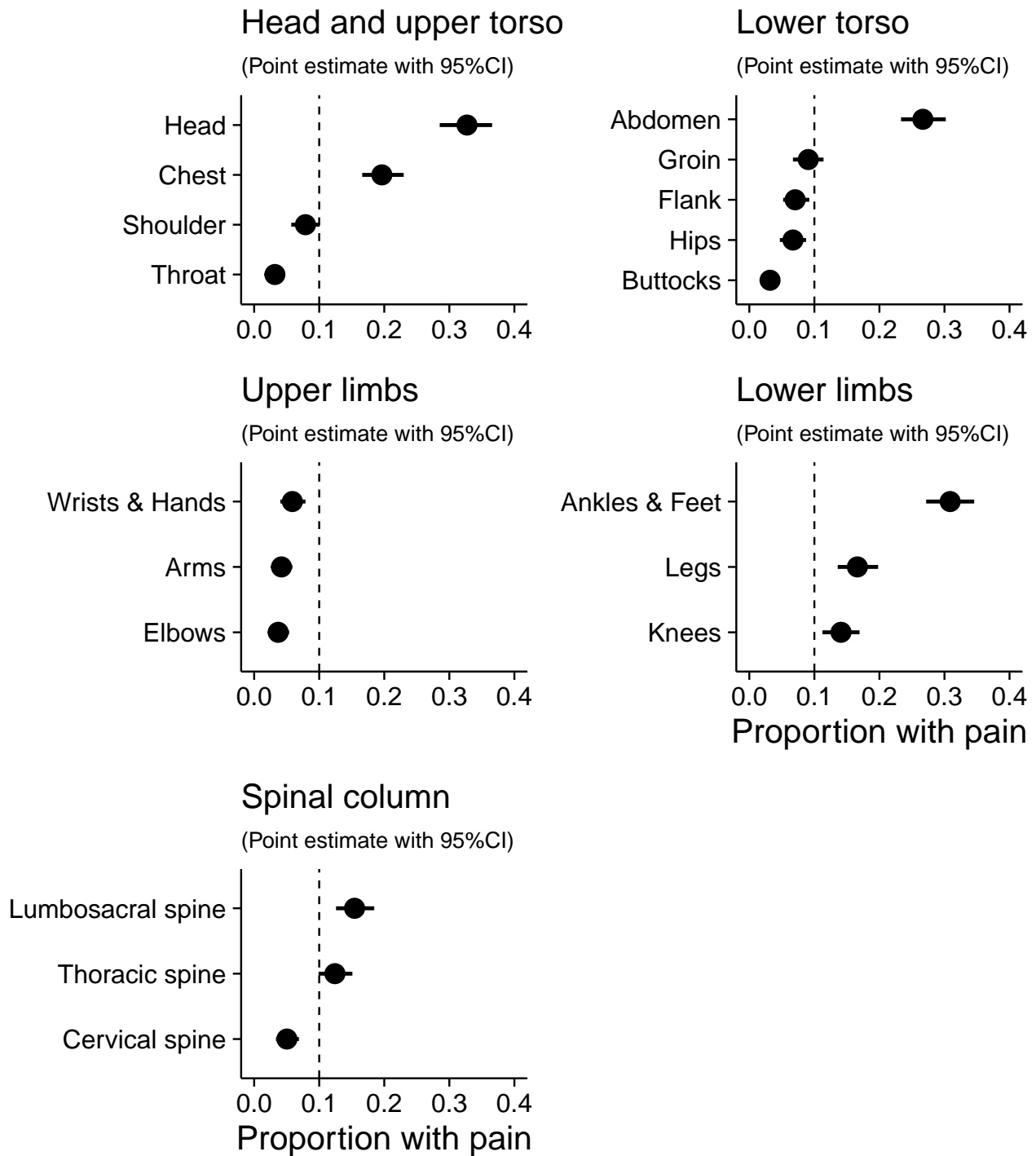
Head and upper torso
(Point estimate with 95%CI)

Lower torso
(Point estimate with 95%CI)

Upper limbs
(Point estimate with 95%CI)

Lower limbs
(Point estimate with 95%CI)

Proportion with pain

Spinal column
(Point estimate with 95%CI)

Proportion with pain

```
# Save
ggsave(filename = 'figures/figure_1.png',
       plot = plot_prop,
       width = 12,
       height = 12)
```

4 By sex

4.1 Process data

```r
# Set seed
set.seed(2020)

# Select sex data
sex <- demo[, c('ID', 'Sex')]

# Join to boot_data & remove ID, site, and upper_back (only one outcome -- no pain)
sex <- left_join(data, sex) %>%
    select(-ID, -Site, -Upper_back)

# Bootstrap CIs
sex_boot <- sex %>%
    # Pivot to long format
    pivot_longer(cols = -Sex,
                 names_to = 'body_site',
                 values_to = 'pain_present') %>%
    # Add body regions
    mutate(region = case_when(
        body_site == 'Chest' |
            body_site == 'Head' |
            body_site == 'Throat' |
            body_site == 'Shoulder' ~ 'Head and upper torso',
        body_site == 'Lower_back' |
            body_site == 'Abdomen' |
            body_site == 'Hips' |
            body_site == 'Buttocks' |
            body_site == 'Groin' ~ 'Lower torso',
        body_site == 'Legs' |
            body_site == 'Knees' |
            body_site == 'Ankles.Feet' ~ 'Lower limbs',
        body_site == 'Arms' |
            body_site == 'Elbows' |
            body_site == 'Wrists.Hands' ~ 'Upper limbs',
        body_site == 'Cervical_spine' |
            body_site == 'Thoracic_spine' |
            body_site == 'Lumbosacral_spine' ~ 'Spinal column',
        TRUE ~ 'other'
    )) %>%
    # Nest by body region and body site
    group_by(Sex, region, body_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
```

```r
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-data, -boot, -ci) %>%
    # Unnest
    unnest(cols = c(point_est, lower_ci, upper_ci))

# Re-nest by body region and generate figures and tables
sex_boot2 <- sex_boot %>%
    group_by(region) %>%
    nest() %>%
    # Fix site labels
    mutate(data = map(.x = data,
                      ~ .x %>%
                          mutate(body_site = str_replace_all(body_site,
                                                             pattern = '_',
                                                             replacement = ' '),
                                 body_site = str_replace_all(body_site,
                                                             pattern = '\\.',
                                                             replacement = ' & ')))) %>%
    # Re-order sites by point_est
    mutate(data = map(.x = data,
                      ~ .x %>%
                          mutate(body_site = fct_reorder(body_site,
                                                         point_est)))) %>%
    # Plot data
    mutate(plots = map2(.x = data,
                        .y = region,
                        ~ .x %>%
                            ggplot(data = .) +
                            aes(x = body_site,
                                y = point_est,
                                ymin = lower_ci,
                                ymax = upper_ci,
                                fill = Sex) +
                            geom_linerange(position = position_dodge2(width = 0.6),
                                           size = 1,
                                           colour = '#000000') +
                            geom_point(shape = 21,
                                       colour = '#000000',
                                       position = position_dodge2(width = 0.6),
                                       size = 6,
                                       stroke = 1) +
                            coord_flip() +
                            labs(title = .y,
                                 subtitle = '(Point estimate with 95%CI)',
```

```
                              y = 'Proportion with pain') +
                  scale_y_continuous(limits = c(0, 1)) +
                  scale_fill_manual(values = c('#000000', '#FFFFFF')) +
                  theme_minimal(base_size = 18) +
                  theme(plot.title = element_text(size = 18),
                        plot.subtitle = element_text(size = 12),
                        legend.title = element_blank(),
                        legend.position = 'top',
                        axis.title.y = element_blank(),
                        panel.grid = element_blank(),
                        axis.text = element_text(colour = '#000000'),
                        axis.line = element_line(size = 0.5),
                        axis.ticks = element_line(size = 0.5)))) %>%
  # Tabulate data
  mutate(tables = map2(.x = data,
                       .y = region,
                       ~ .x %>%
                          kable(caption = .y,
                                digits = 2)))
```

4.2 Tabulated proportions (with 95% CIs), by age and body region

```
walk(sex_boot2$tables, ~ print(.x))
```

Table 13: Head and upper torso

Sex	body_site	point_est	lower_ci	upper_ci
Female	Head	0.38	0.33	0.42
Female	Throat	0.03	0.01	0.04
Female	Shoulder	0.07	0.05	0.10
Female	Chest	0.20	0.16	0.23
Male	Head	0.12	0.06	0.18
Male	Throat	0.04	0.01	0.09
Male	Shoulder	0.11	0.06	0.18
Male	Chest	0.20	0.13	0.27

Table 14: Upper limbs

Sex	body_site	point_est	lower_ci	upper_ci
Female	Arms	0.04	0.03	0.06
Female	Elbows	0.03	0.02	0.05
Female	Wrists & Hands	0.06	0.04	0.09
Male	Arms	0.03	0.01	0.07
Male	Elbows	0.05	0.02	0.10
Male	Wrists & Hands	0.03	0.01	0.08

Table 15: Lower torso

Sex	body_site	point_est	lower_ci	upper_ci
Female	Lower back	0.07	0.05	0.10
Female	Abdomen	0.28	0.24	0.32
Female	Groin	0.10	0.07	0.12
Female	Hips	0.06	0.04	0.09
Female	Buttocks	0.03	0.01	0.04
Male	Lower back	0.05	0.02	0.10
Male	Abdomen	0.19	0.12	0.27
Male	Groin	0.07	0.03	0.12
Male	Hips	0.10	0.05	0.15
Male	Buttocks	0.04	0.01	0.09

Table 16: Spinal column

Sex	body_site	point_est	lower_ci	upper_ci
Female	Cervical spine	0.06	0.04	0.08
Female	Thoracic spine	0.14	0.11	0.17
Female	Lumbosacral spine	0.16	0.12	0.19
Male	Cervical spine	0.03	0.00	0.06
Male	Thoracic spine	0.07	0.03	0.12
Male	Lumbosacral spine	0.14	0.08	0.20

Table 17: Lower limbs

Sex	body_site	point_est	lower_ci	upper_ci
Female	Legs	0.15	0.12	0.18
Female	Knees	0.13	0.10	0.16
Female	Ankles & Feet	0.28	0.25	0.32
Male	Legs	0.23	0.16	0.31
Male	Knees	0.19	0.12	0.26
Male	Ankles & Feet	0.42	0.33	0.50

4.3 Plotted proportions (with 95% CIs), by age and body region

```
walk(sex_boot2$plots, ~ print(.x))
```
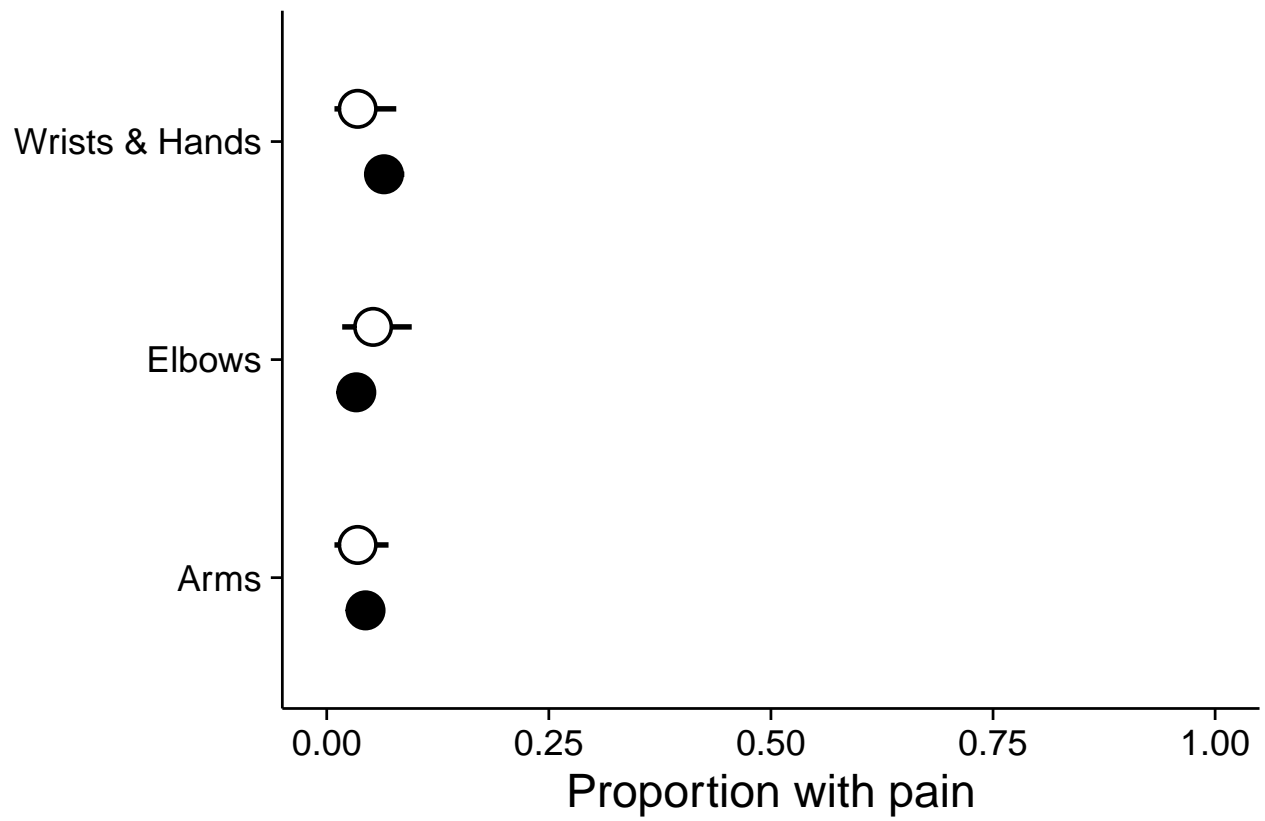
Head and upper torso

(Point estimate with 95%CI)
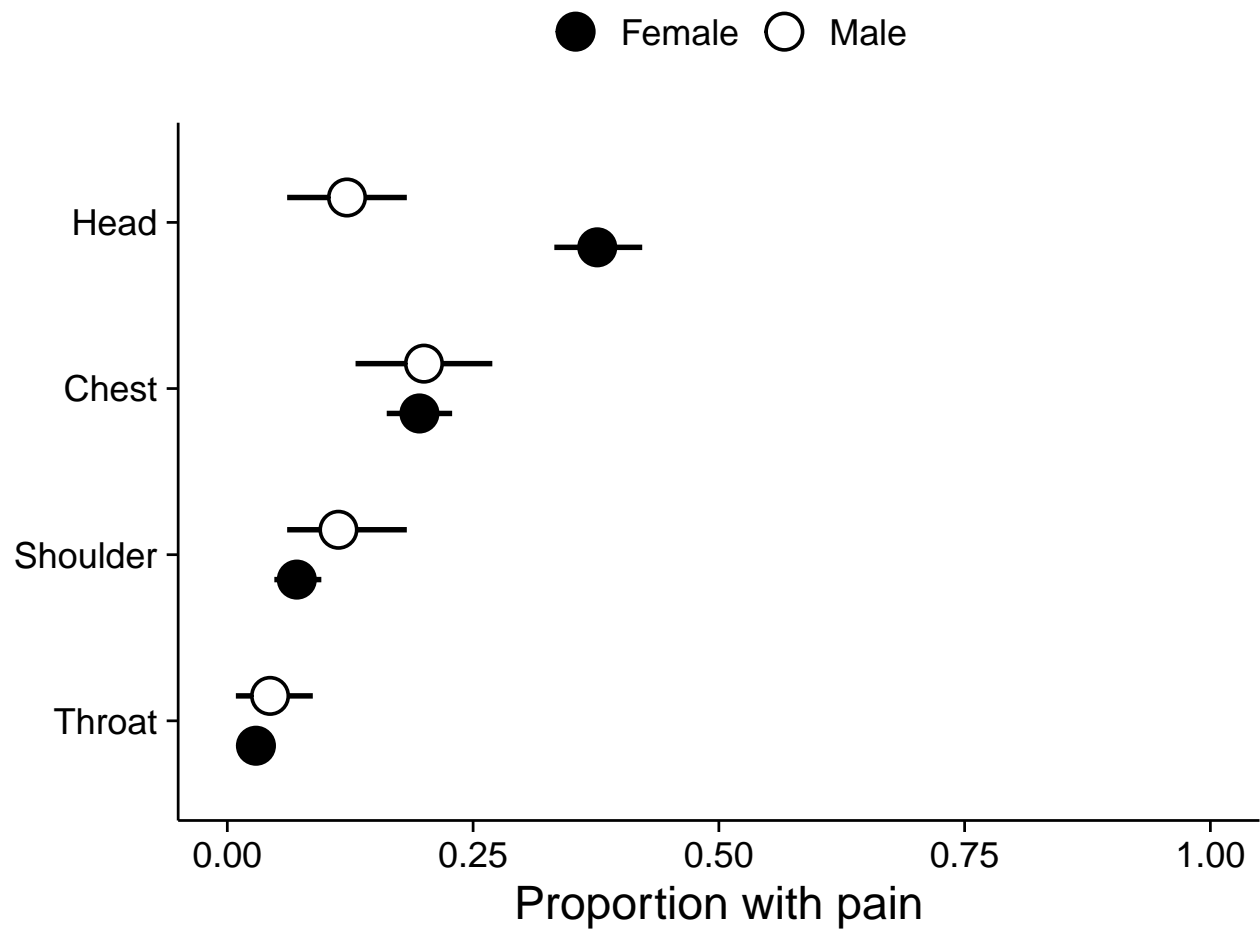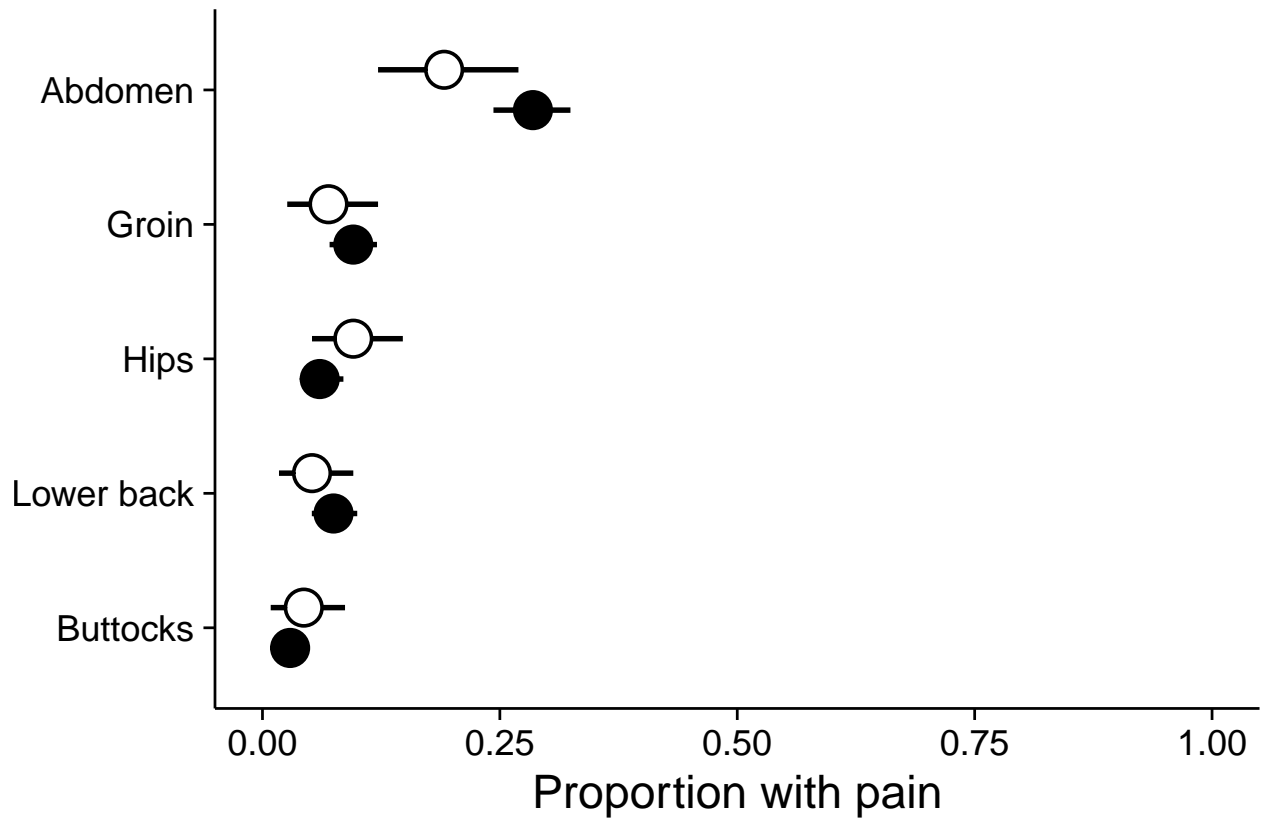
Upper limbs

(Point estimate with 95%CI)

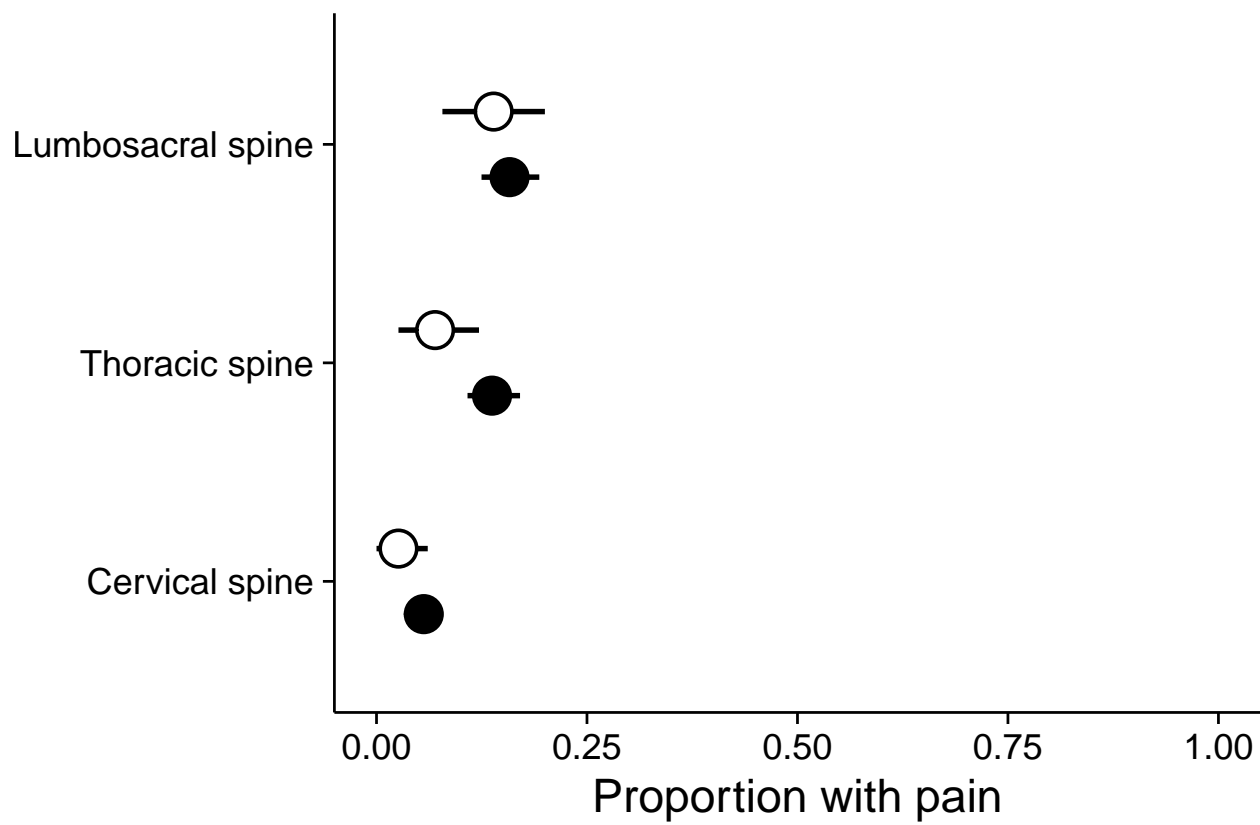● Female ○ Male

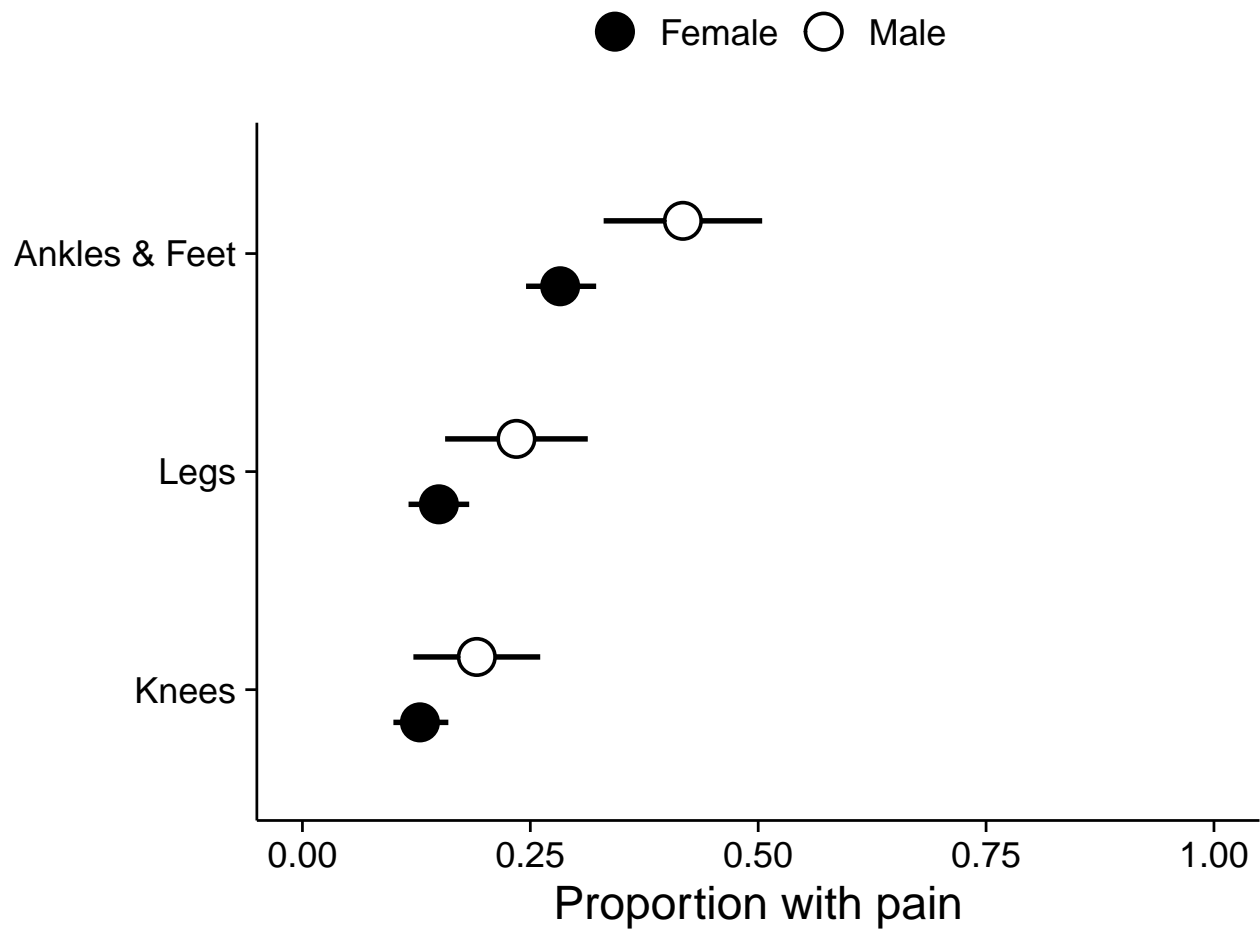# Lower torso

(Point estimate with 95%CI)

Spinal column

(Point estimate with 95%CI)

Lower limbs
(Point estimate with 95%CI)

● Female ○ Male

5 By age

For ease of tabulation and plotting in this section of the supplement, I divided age into seven age categories. However for data analysis (see: Logistic regression, section 7), I analysed age as a continuous variable.

5.1 Process data

```r
# Set seed
set.seed(2020)

# Select age data
age <- demo[, c('ID', 'Age')]

# Join to boot_data & remove ID, study site, and upper_back
# (only one outcome -- no pain)
age <- left_join(data, age) %>%
    select(-ID, -Site, -Upper_back)
```

```r
# Get complete cases
age <- age[complete.cases(age), ]

# Pivot and add age group categories (10 year periods)
age_boot <- age %>%
    # Pivot to long format
    pivot_longer(cols = -Age,
                 names_to = 'body_site',
                 values_to = 'pain_present') %>%
    # Add age categories
    mutate(age_group = case_when(
        Age < 28 ~ '18-27',
        Age >= 28 & Age < 38 ~ '28-37',
        Age >= 38 & Age < 48 ~ '38-47',
        Age >= 48 & Age < 58 ~ '48-57',
        Age >= 58 & Age < 68 ~ '58-67',
        Age >= 68 & Age < 78 ~ '68-77',
        Age >= 78 & Age < 88 ~ '78-87'
    ))

# Print count per age group
age_boot %>%
    group_by(body_site, age_group) %>%
    summarise(count = n()) %>%
    filter(body_site == 'Abdomen') %>%
    ungroup() %>%
    select(-body_site) %>%
    kable(caption = 'Participant count per age group')
```

Table 18: Participant count per age group

age_group	count
18-27	65
28-37	283
38-47	160
48-57	58
58-67	20
68-77	2

```r
# Generate CIs
age_boot2 <- age_boot %>%
    # Remove age
    select(-Age) %>%
    # Remove categories with less than 20 counts
    filter(age_group != '68-77') %>%
    # Nest by age group and body site
    group_by(age_group, body_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = prop_func,
```

```r
                            R = 999,
                            stype = 'i',
                            parallel = 'multicore',
                            ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-data, -boot, -ci) %>%
    # Unnest
    unnest(cols = c(point_est, lower_ci, upper_ci)) %>%
    ungroup()

# Re-nest by body region and generate figures and tables
age_boot2 <- age_boot2 %>%
    # Fix site labels
    mutate(body_site = str_replace_all(body_site,
                                       pattern = '_',
                                       replacement = ' '),
           body_site = str_replace_all(body_site,
                                       pattern = '\\.',
                                       replacement = ' & ')) %>%
    # Group and nest
    group_by(body_site) %>%
    nest() %>%
    # Arrange age groups
    # Plot data
    mutate(plots = map2(.x = data,
                        .y = body_site,
                        ~ .x %>%
                            ggplot(data = .) +
                            aes(x = age_group,
                                y = point_est,
                                ymin = lower_ci,
                                ymax = upper_ci) +
                            geom_linerange(size = 1,
                                           colour = '#000000') +
                            geom_point(colour = '#000000',
                                       size = 6) +
                            labs(title = .y,
                                 subtitle = '(Point estimate with 95%CI)',
                                 caption = 'Age group 68-77 years removed because n = 2',
                                 x = 'Age group (Years)',
                                 y = 'Proportion with pain') +
                            scale_y_continuous(limits = c(0, 1)) +
                            coord_flip() +
```

```
                            theme_minimal(base_size = 18) +
                            theme(plot.title = element_text(size = 18),
                                  plot.subtitle = element_text(size = 12),
                                  plot.caption = element_text(size = 12),
                                  panel.grid = element_blank(),
                                  axis.text = element_text(colour = '#000000'),
                                  axis.line = element_line(size = 0.5),
                                  axis.ticks = element_line(size = 0.5)))) %>%
  # Tabulate data
  mutate(tables = map2(.x = data,
                       .y = body_site,
                       ~ .x %>%
                         arrange(age_group) %>%
                         kable(caption = str_glue('{.y} (Age group 68-77 years removed because n =
                               digits = 2)))
```

5.2   Tabulated proportions (with 95% CIs), by age group and body site

```
walk(age_boot2$tables, ~ print(.x))
```

Table 19: Head (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.45	0.32	0.57
28-37	0.34	0.28	0.39
38-47	0.31	0.24	0.39
48-57	0.17	0.09	0.28
58-67	0.35	0.15	0.55

Table 20: Throat (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.05	0.00	0.11
28-37	0.03	0.01	0.05
38-47	0.03	0.01	0.06
48-57	0.05	0.00	0.12
58-67	0.00	0.00	0.00

Table 21: Shoulder (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.12	0.05	0.22
28-37	0.07	0.04	0.10
38-47	0.09	0.05	0.14
48-57	0.07	0.02	0.14
58-67	0.05	0.00	0.15

Table 22: Arms (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.02	0.00	0.05
28-37	0.04	0.02	0.07
38-47	0.06	0.02	0.09
48-57	0.05	0.00	0.12
58-67	0.00	0.00	0.00

Table 23: Elbows (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.00	0.00	0.00
28-37	0.03	0.01	0.06
38-47	0.04	0.01	0.07
48-57	0.05	0.00	0.10
58-67	0.15	0.00	0.30

Table 24: Wrists & Hands (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.02	0.00	0.05
28-37	0.07	0.04	0.10
38-47	0.04	0.01	0.07
48-57	0.07	0.02	0.14
58-67	0.15	0.00	0.30

Table 25: Chest (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.17	0.08	0.26
28-37	0.20	0.15	0.25
38-47	0.20	0.14	0.26
48-57	0.19	0.09	0.29
58-67	0.20	0.05	0.40

Table 26: Lower back (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.09	0.03	0.17
28-37	0.07	0.04	0.10
38-47	0.05	0.02	0.09
48-57	0.10	0.03	0.19
58-67	0.10	0.00	0.25

Table 27: Abdomen (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.23	0.14	0.34
28-37	0.31	0.26	0.37
38-47	0.29	0.22	0.37
48-57	0.10	0.03	0.19
58-67	0.05	0.00	0.15

Table 28: Cervical spine (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.12	0.05	0.20
28-37	0.05	0.02	0.08
38-47	0.04	0.02	0.08
48-57	0.00	0.00	0.00
58-67	0.00	0.00	0.00

Table 29: Thoracic spine (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.18	0.09	0.28
28-37	0.14	0.11	0.19
38-47	0.07	0.03	0.11
48-57	0.09	0.02	0.17
58-67	0.25	0.05	0.45

Table 30: Lumbosacral spine (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.18	0.09	0.29
28-37	0.15	0.11	0.19
38-47	0.12	0.07	0.17
48-57	0.16	0.07	0.26
58-67	0.40	0.20	0.60

Table 31: Groin (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.09	0.03	0.17
28-37	0.10	0.07	0.14
38-47	0.09	0.05	0.13
48-57	0.05	0.00	0.12

age_group	point_est	lower_ci	upper_ci
58-67	0.05	0.00	0.15

Table 32: Hips (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.03	0.00	0.08
28-37	0.05	0.02	0.07
38-47	0.09	0.05	0.14
48-57	0.14	0.05	0.24
58-67	0.10	0.00	0.25

Table 33: Legs (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.12	0.05	0.22
28-37	0.13	0.09	0.17
38-47	0.21	0.15	0.28
48-57	0.29	0.17	0.41
58-67	0.20	0.05	0.40

Table 34: Knees (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.11	0.05	0.18
28-37	0.10	0.07	0.14
38-47	0.17	0.11	0.23
48-57	0.22	0.12	0.34
58-67	0.30	0.10	0.50

Table 35: Ankles & Feet (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.22	0.12	0.32
28-37	0.24	0.19	0.29
38-47	0.37	0.29	0.44
48-57	0.52	0.40	0.64
58-67	0.55	0.35	0.75

Table 36: Buttocks (Age group 68-77 years removed because n = 2)

age_group	point_est	lower_ci	upper_ci
18-27	0.03	0.00	0.08

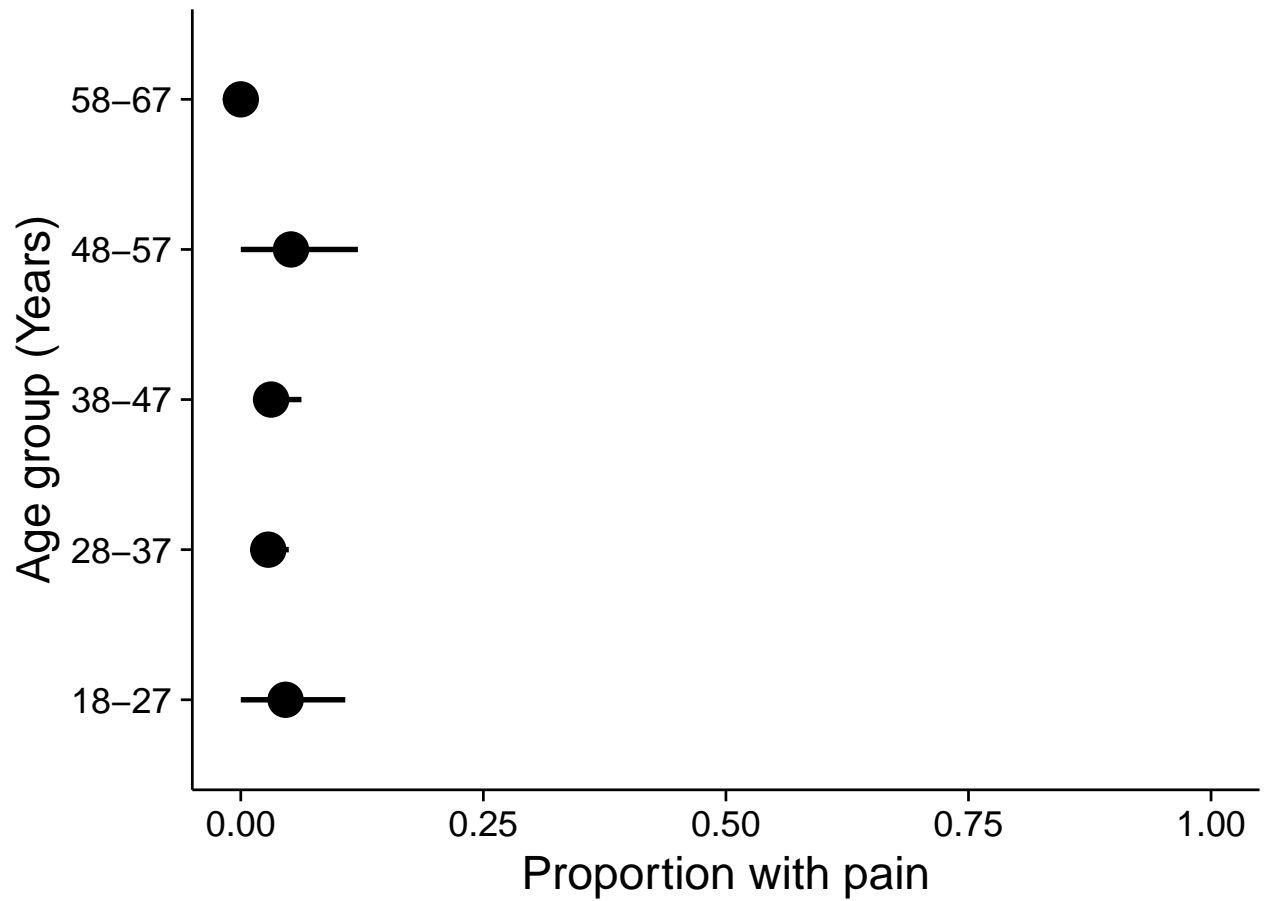| age_group	point_est	lower_ci	upper_ci
28-37	0.04	0.02	0.06
38-47	0.02	0.01	0.05
48-57	0.02	0.00	0.05
58-67	0.05	0.00	0.15

5.3 Plotted proportions (with 95% CIs), by age group and body site

```
walk(age_boot2$plots, ~ print(.x))
```
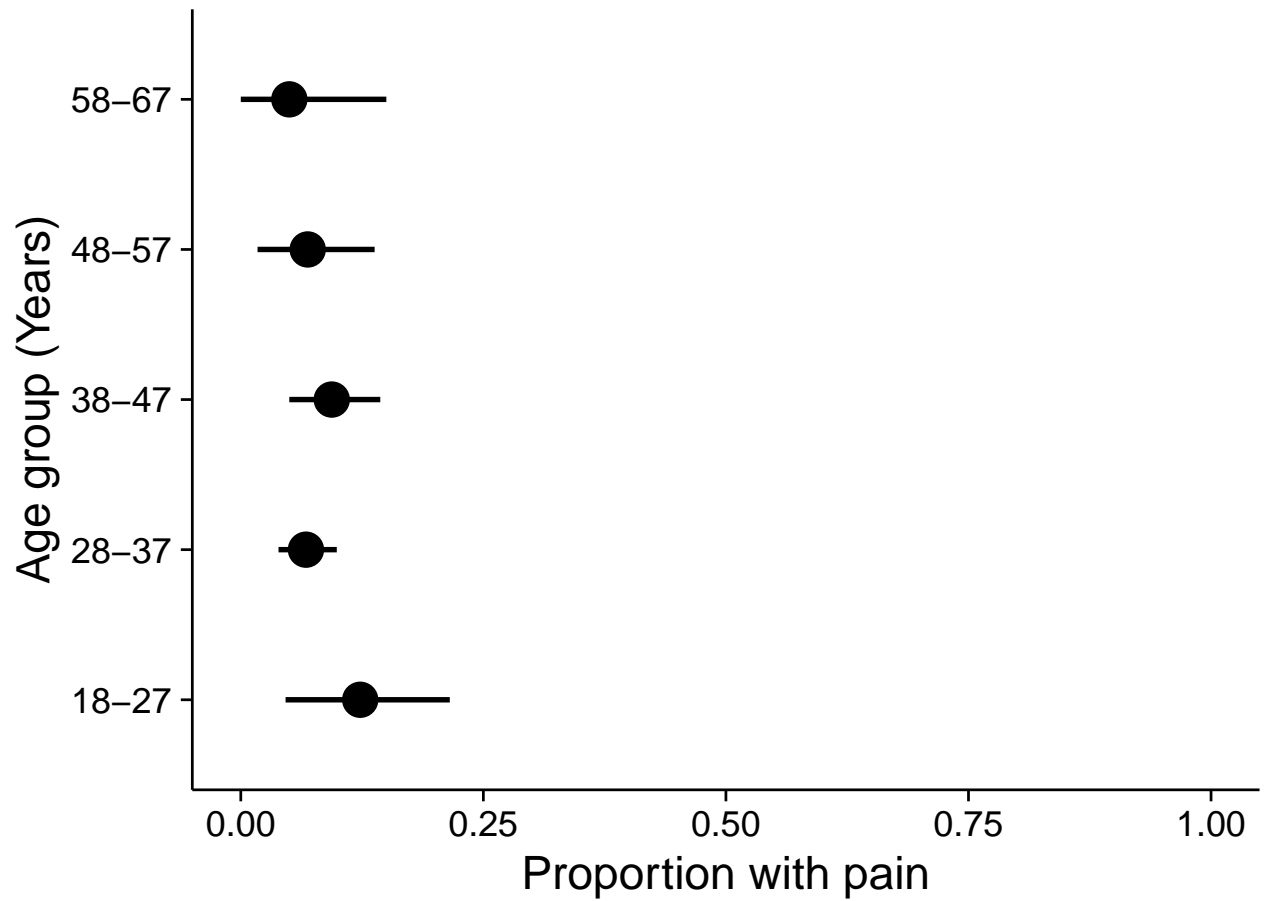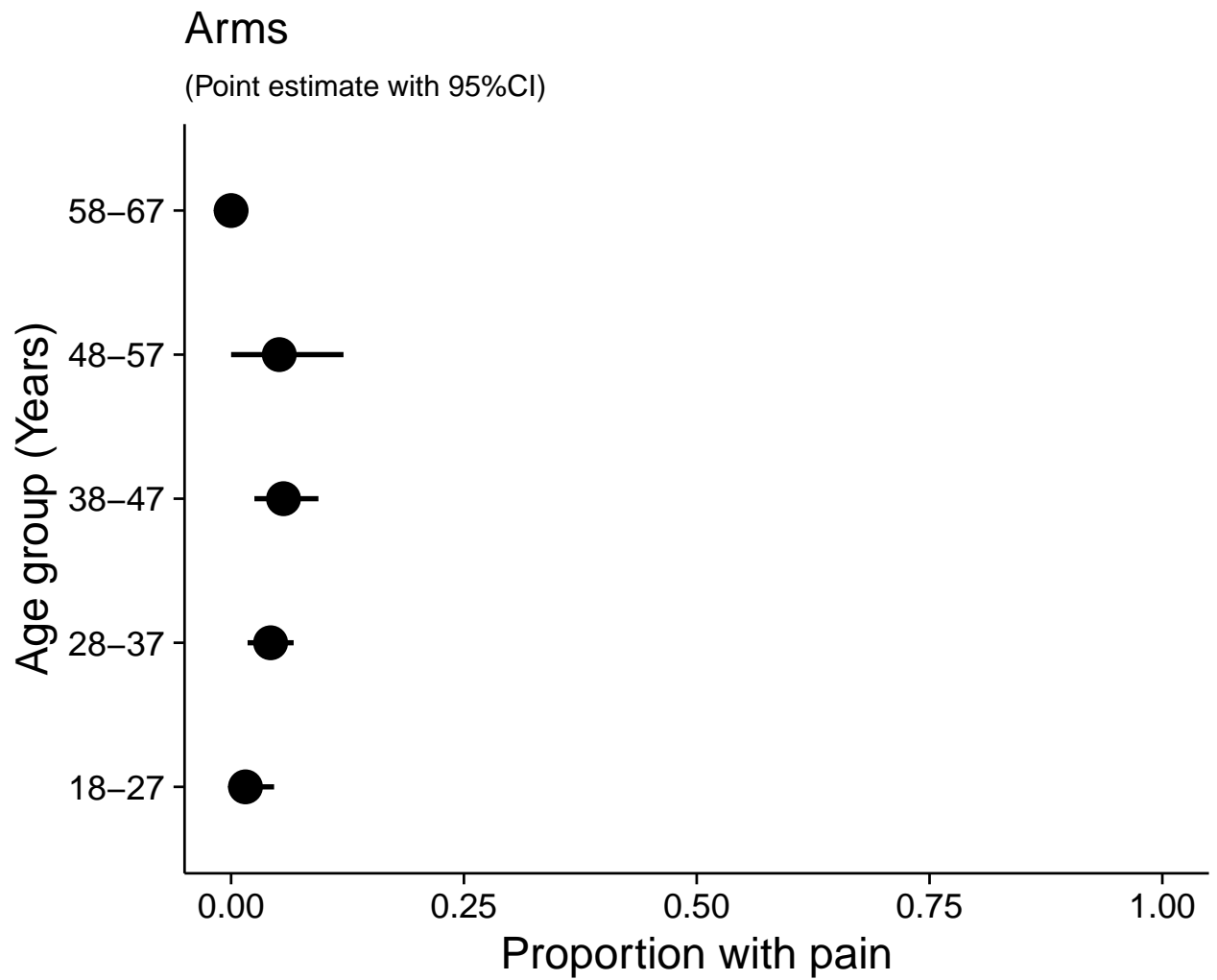
Head

(Point estimate with 95%CI)

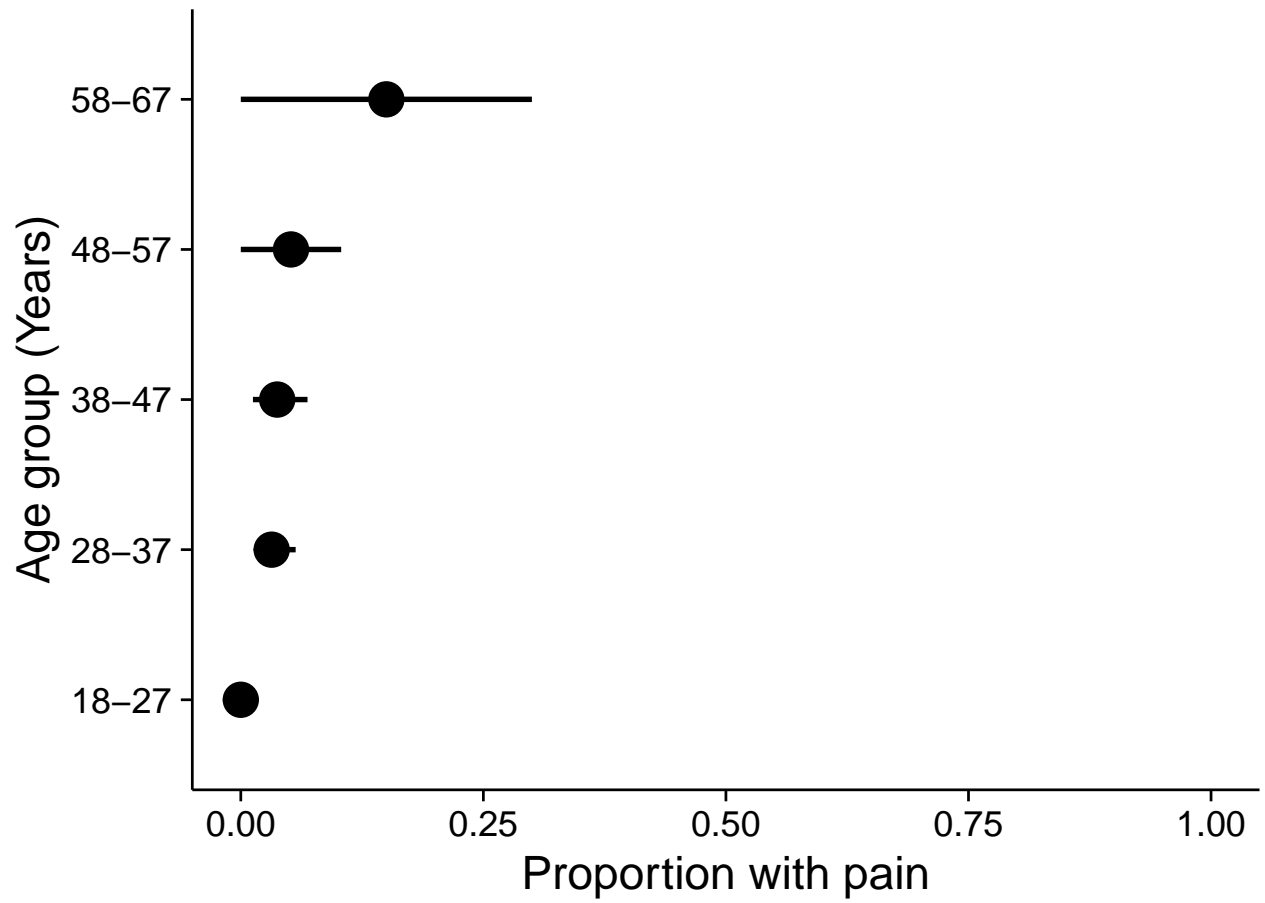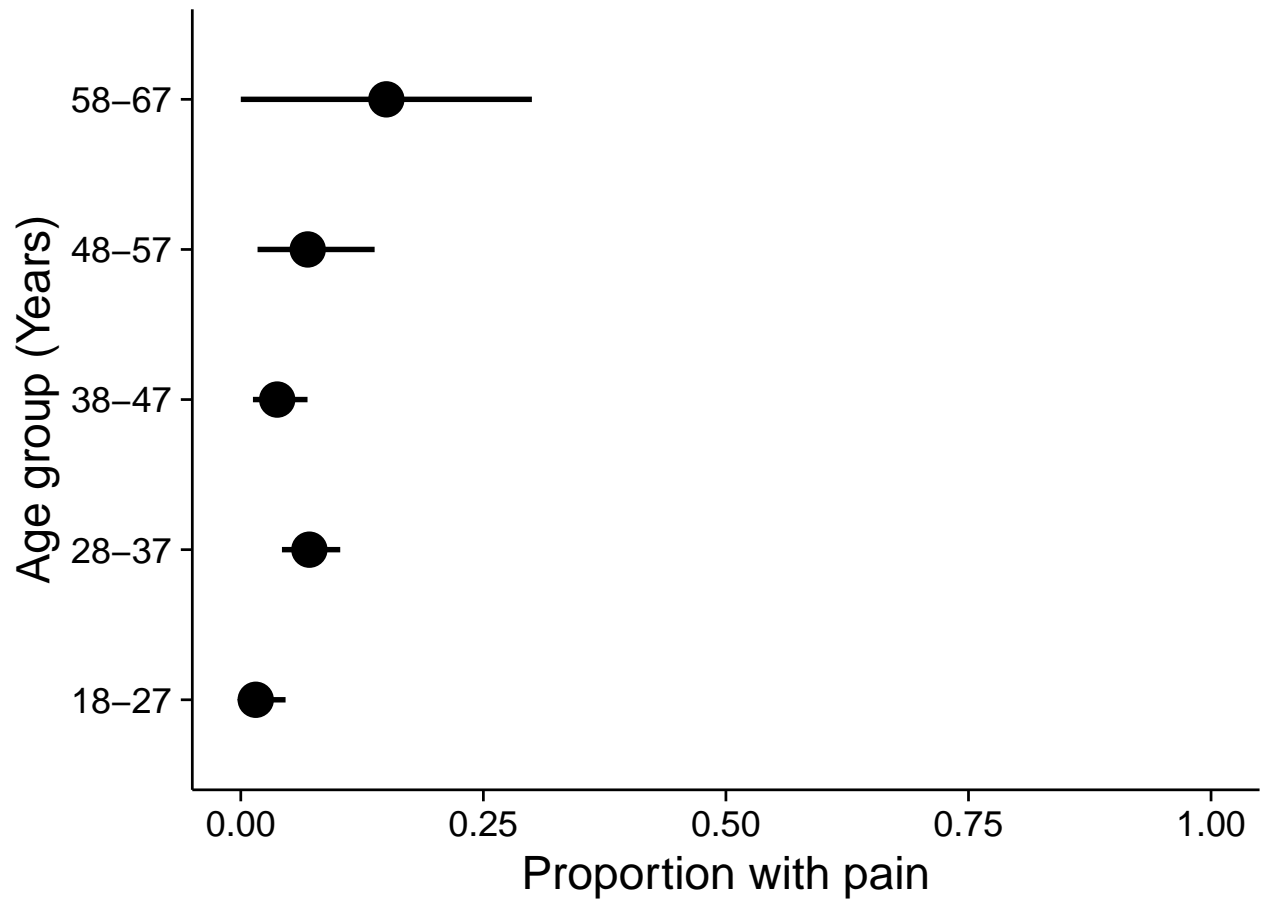Age group 68–77 years removed because n = 2
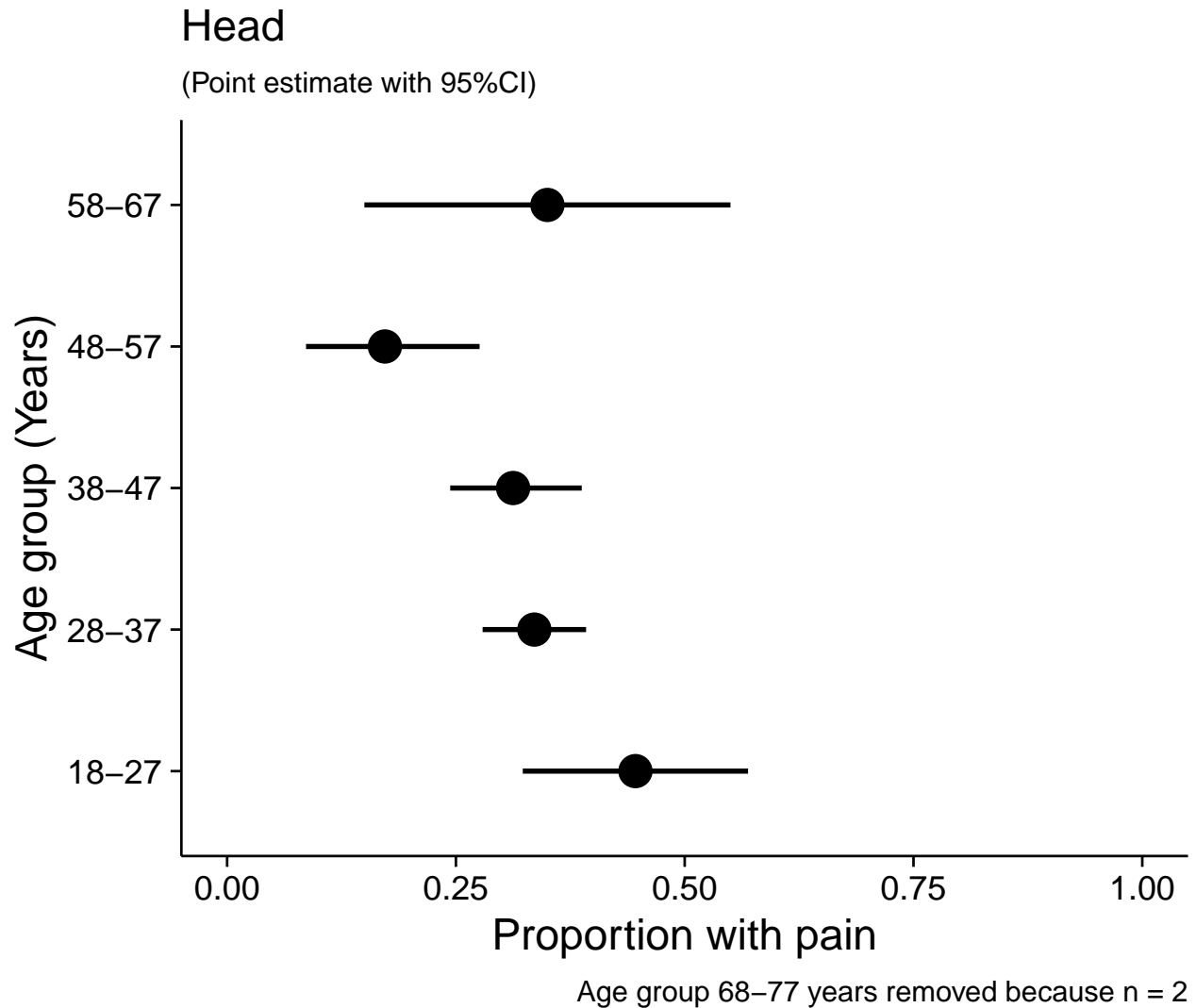
Throat

(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Shoulder

(Point estimate with 95%CI)

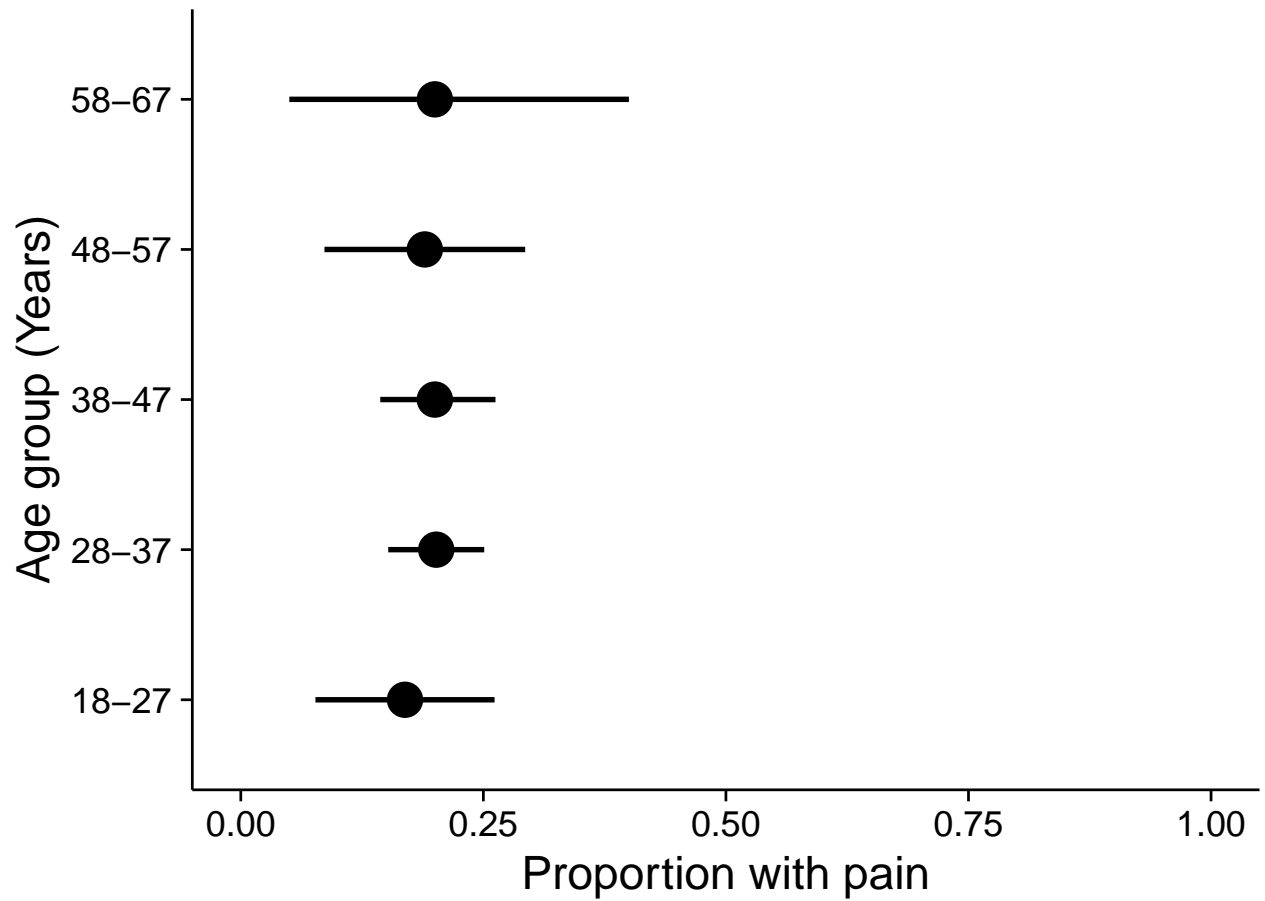Age group 68–77 years removed because n = 2

Arms

(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Elbows
(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Wrists & Hands
(Point estimate with 95%CI)
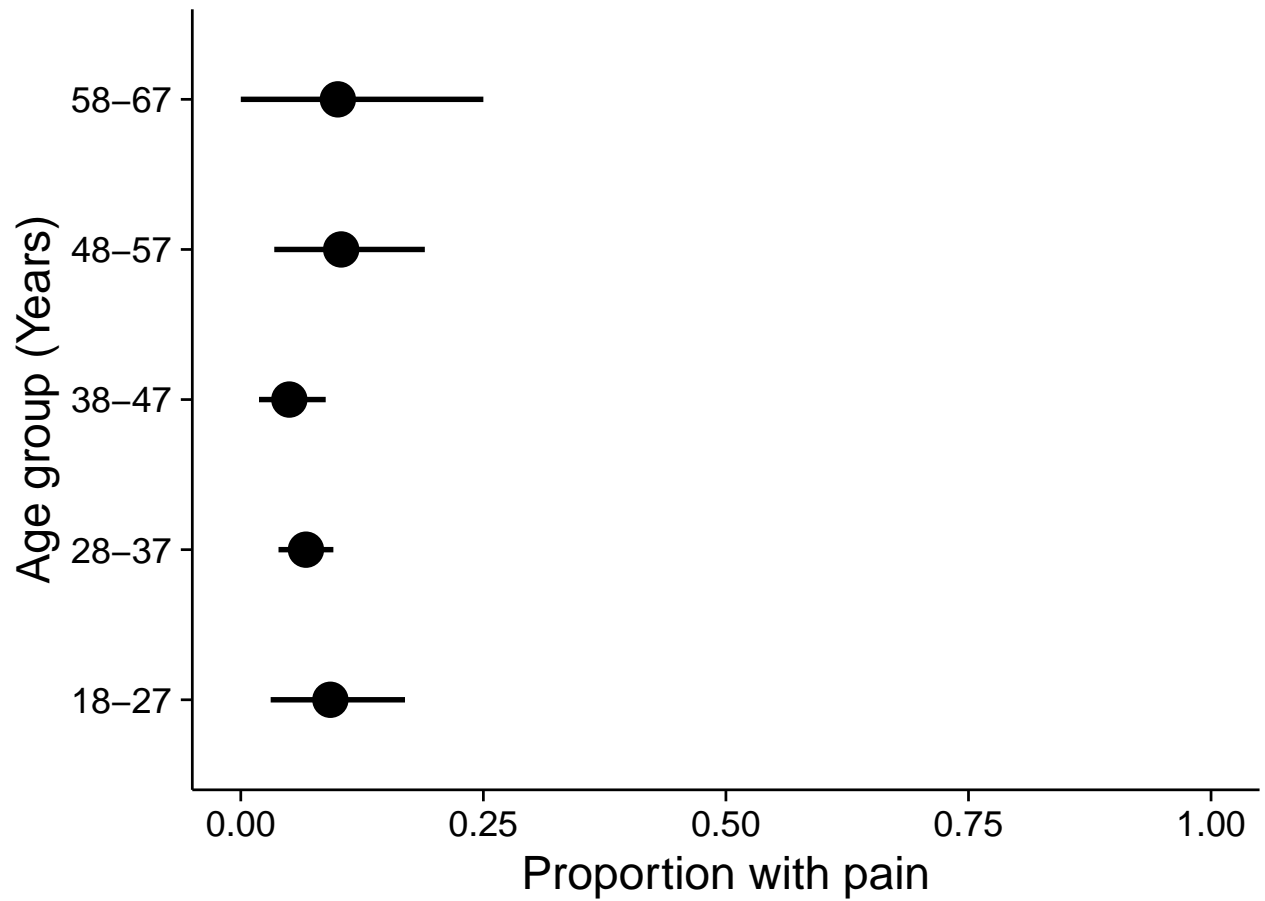
Age group 68–77 years removed because n = 2
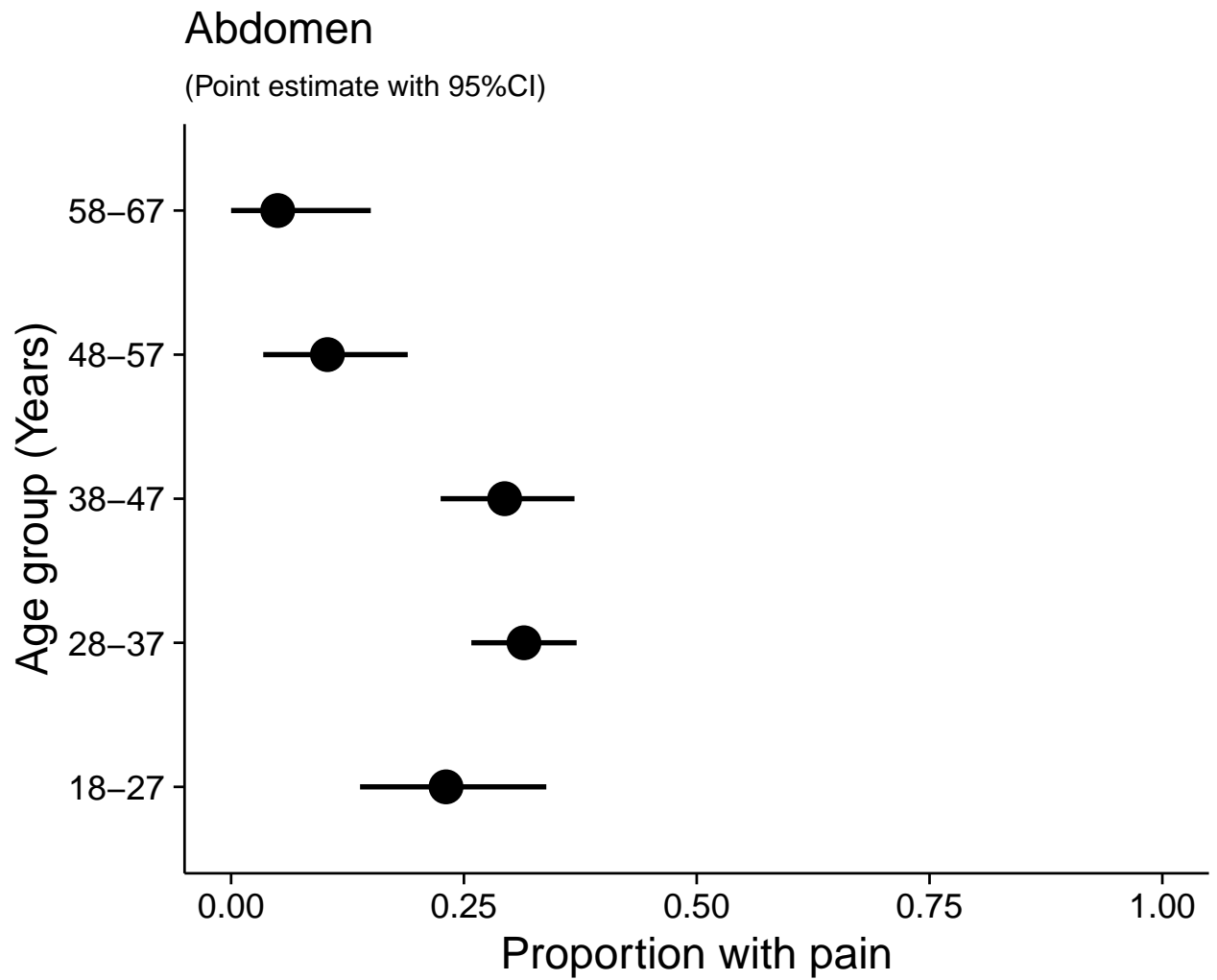
Chest
(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2
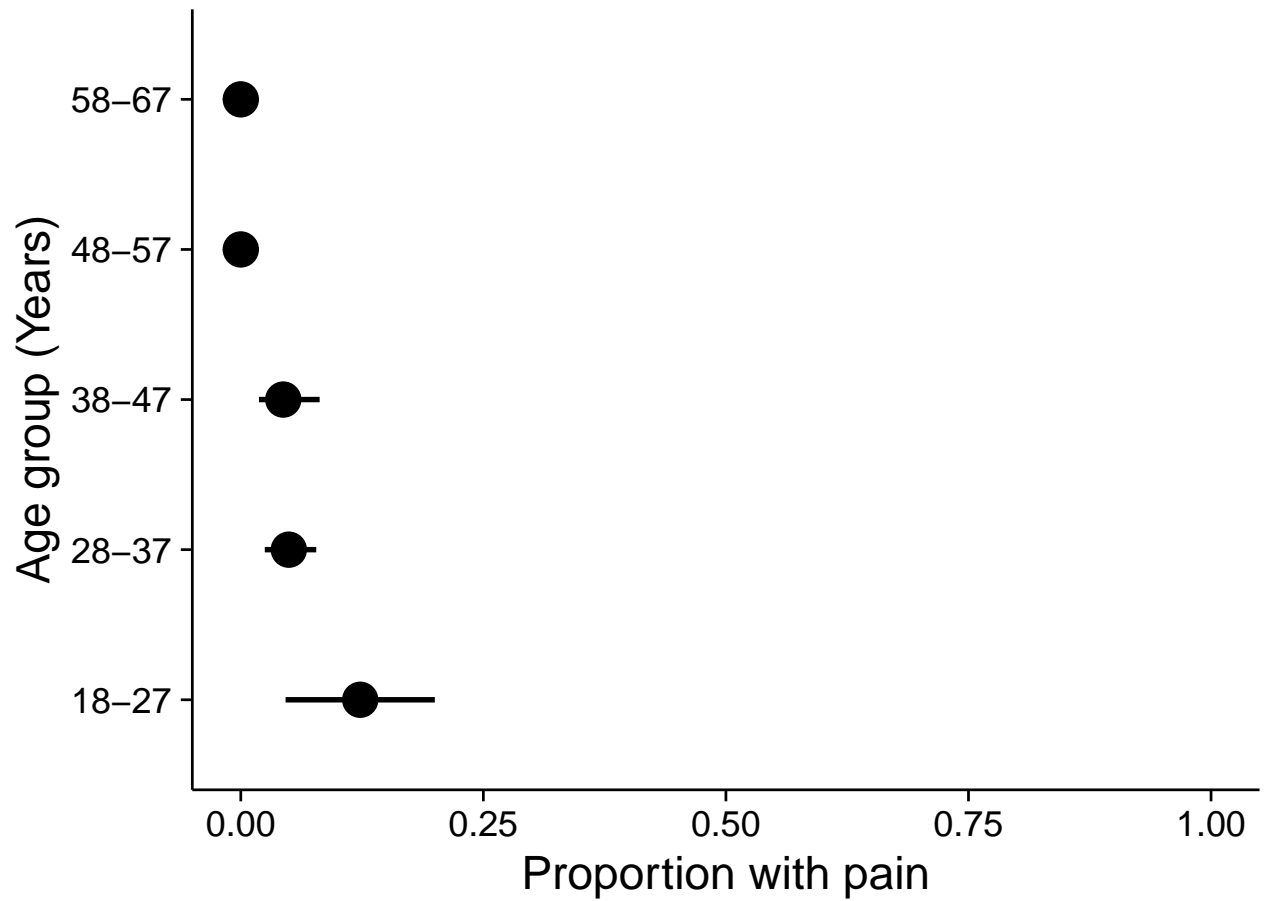
Lower back
(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Abdomen

(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2
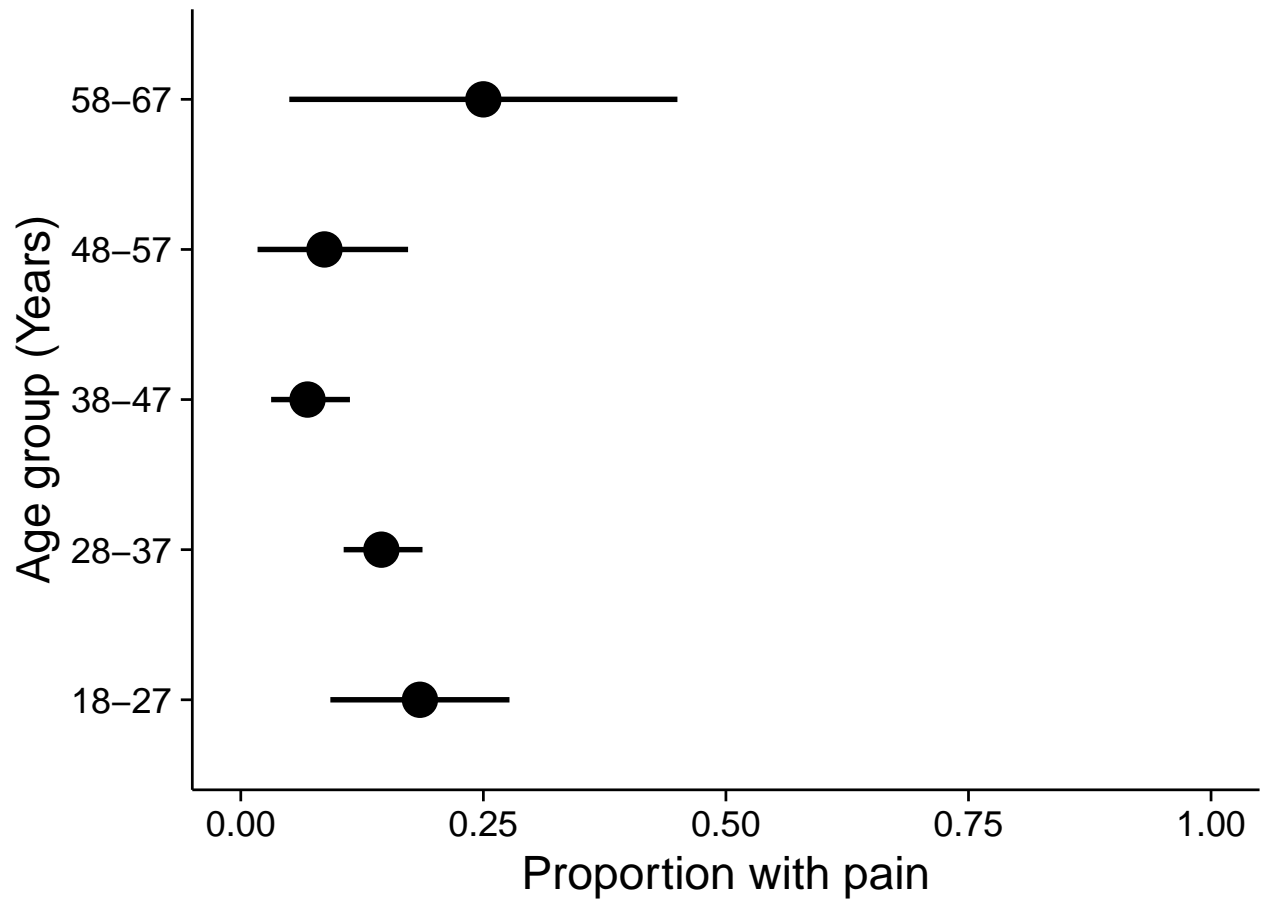
Cervical spine

(Point estimate with 95%CI)

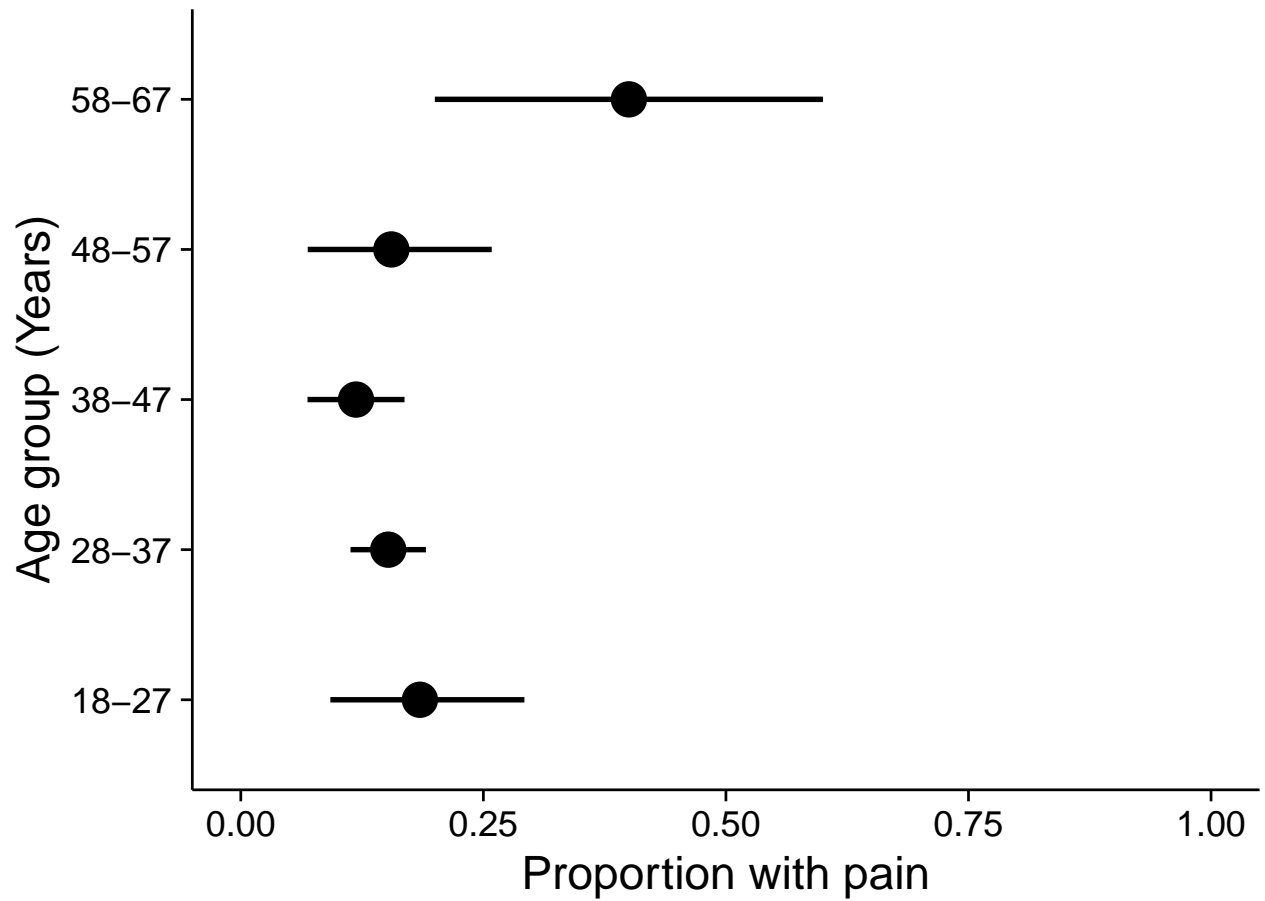Age group 68–77 years removed because n = 2

Thoracic spine

(Point estimate with 95%CI)
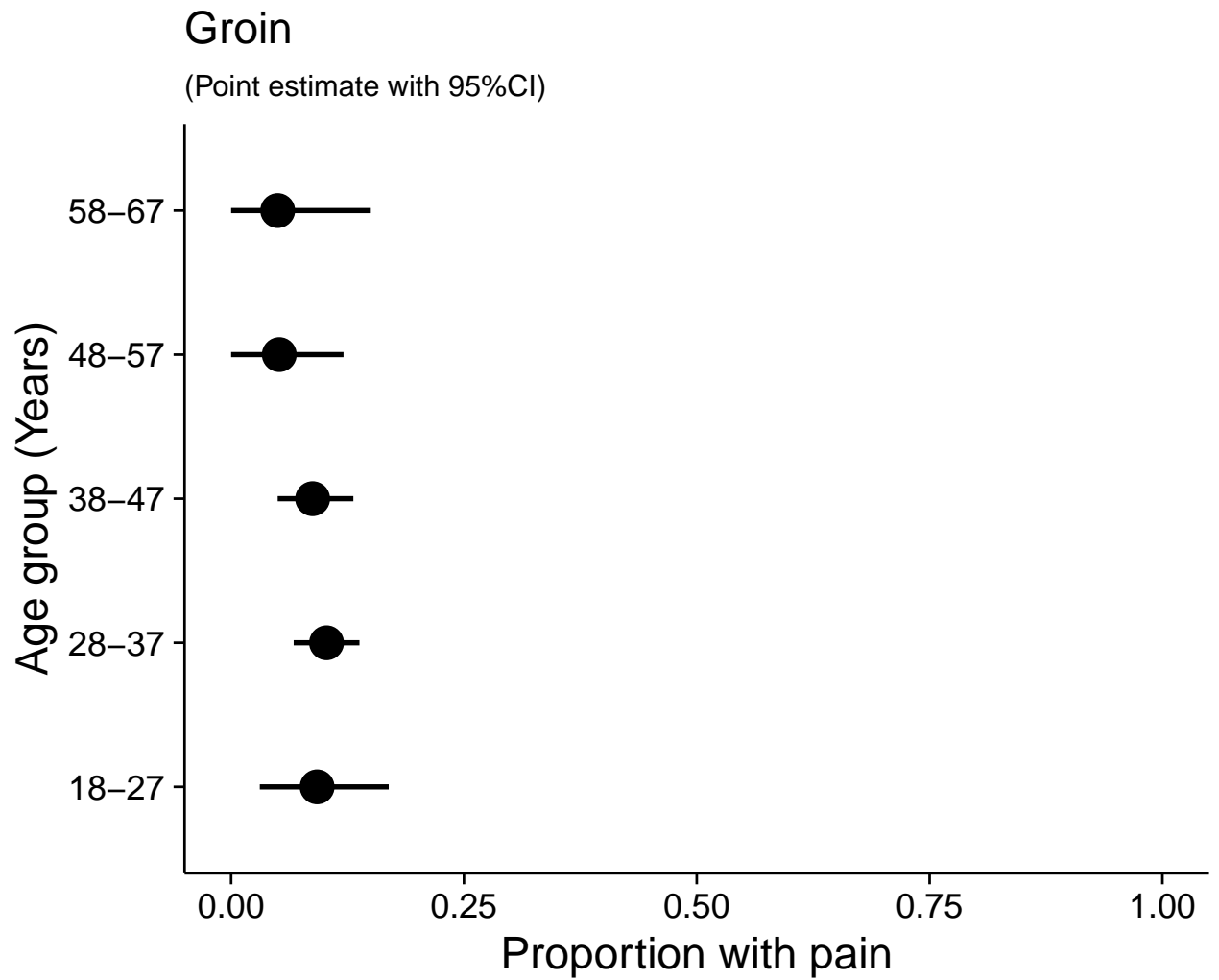
Age group 68–77 years removed because n = 2

Lumbosacral spine

(Point estimate with 95%CI)

Age group 68−77 years removed because n = 2

Groin
(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Hips

(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Legs
(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Knees

(Point estimate with 95%CI)
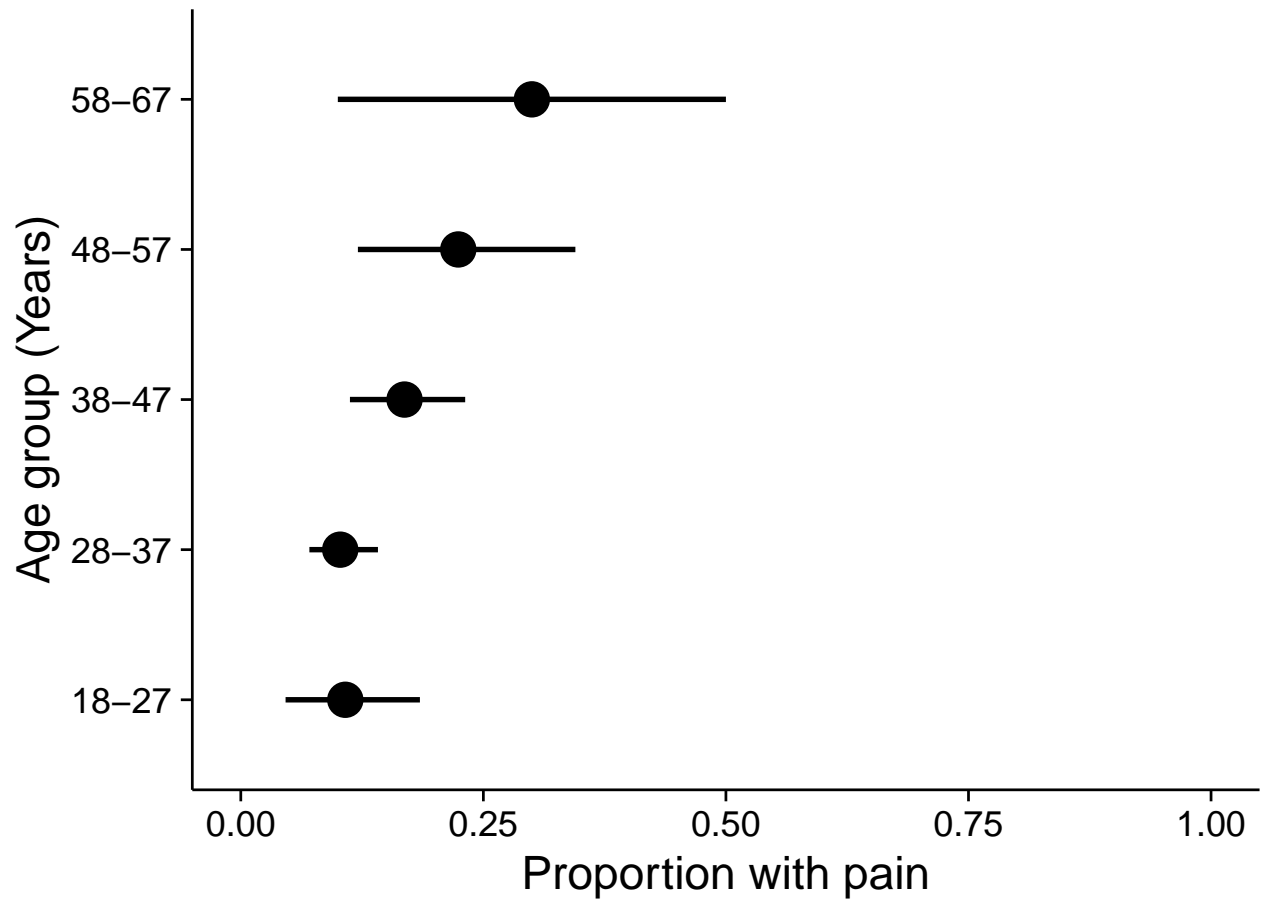
Age group 68–77 years removed because n = 2

Ankles & Feet

(Point estimate with 95%CI)

Age group 68–77 years removed because n = 2

Buttocks
(Point estimate with 95%CI)

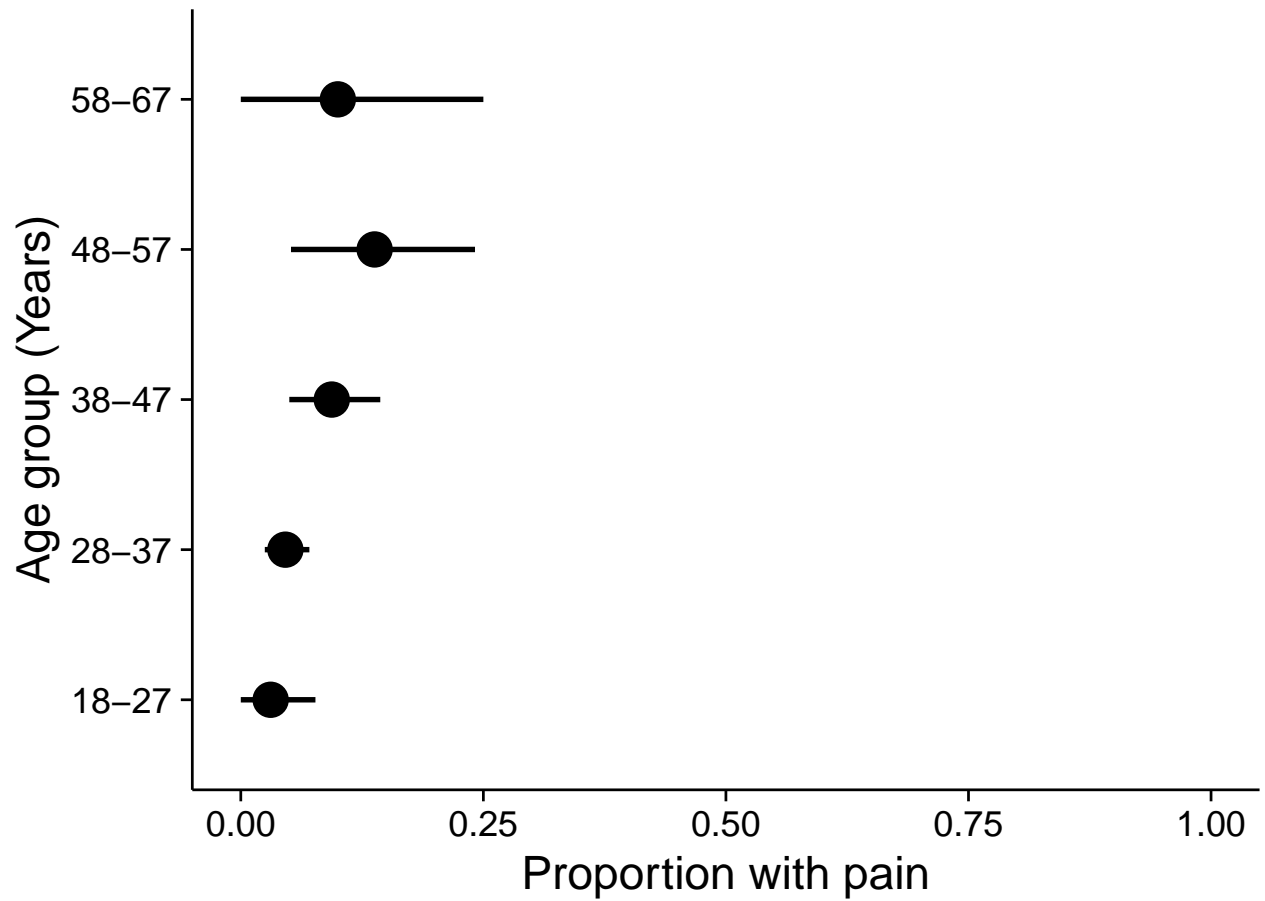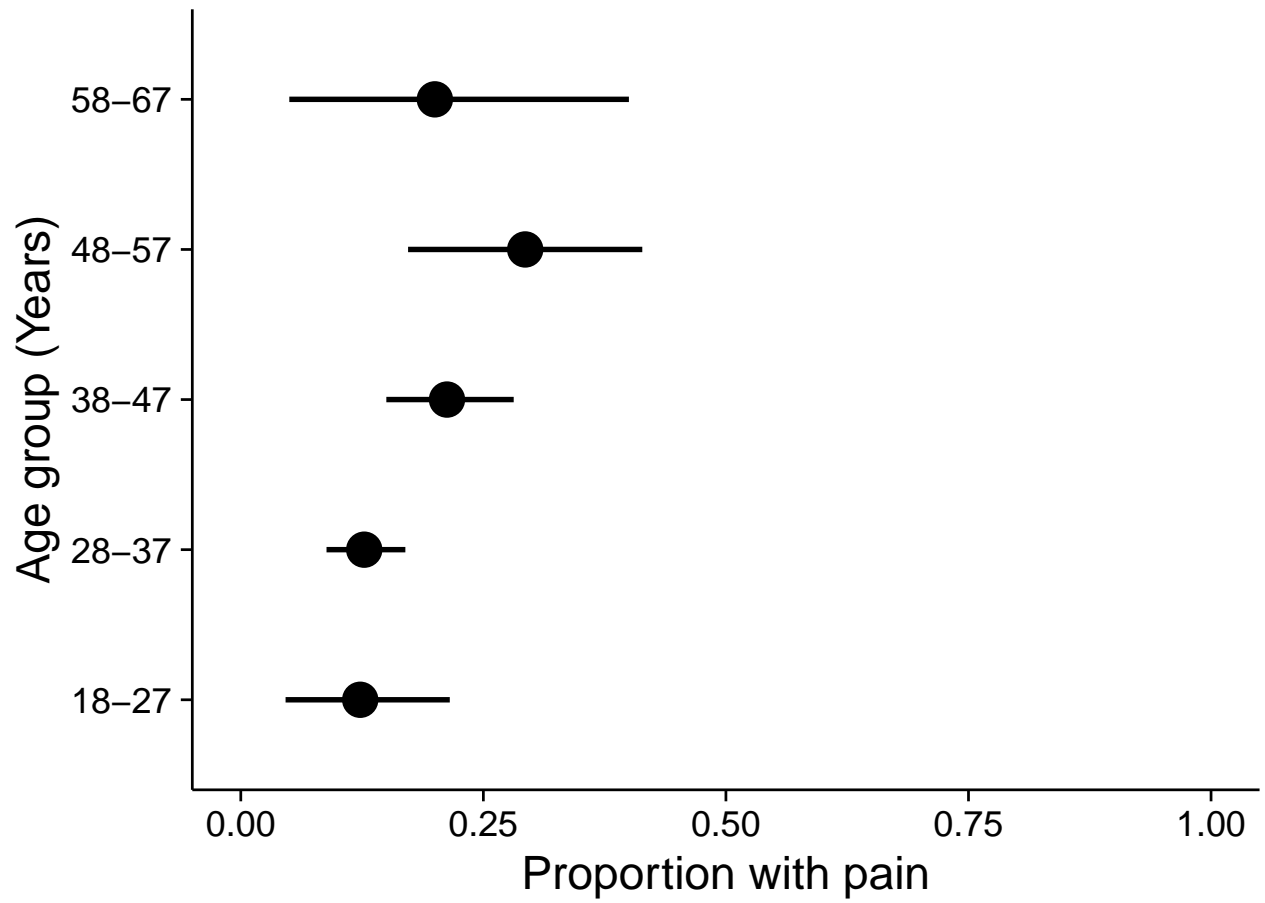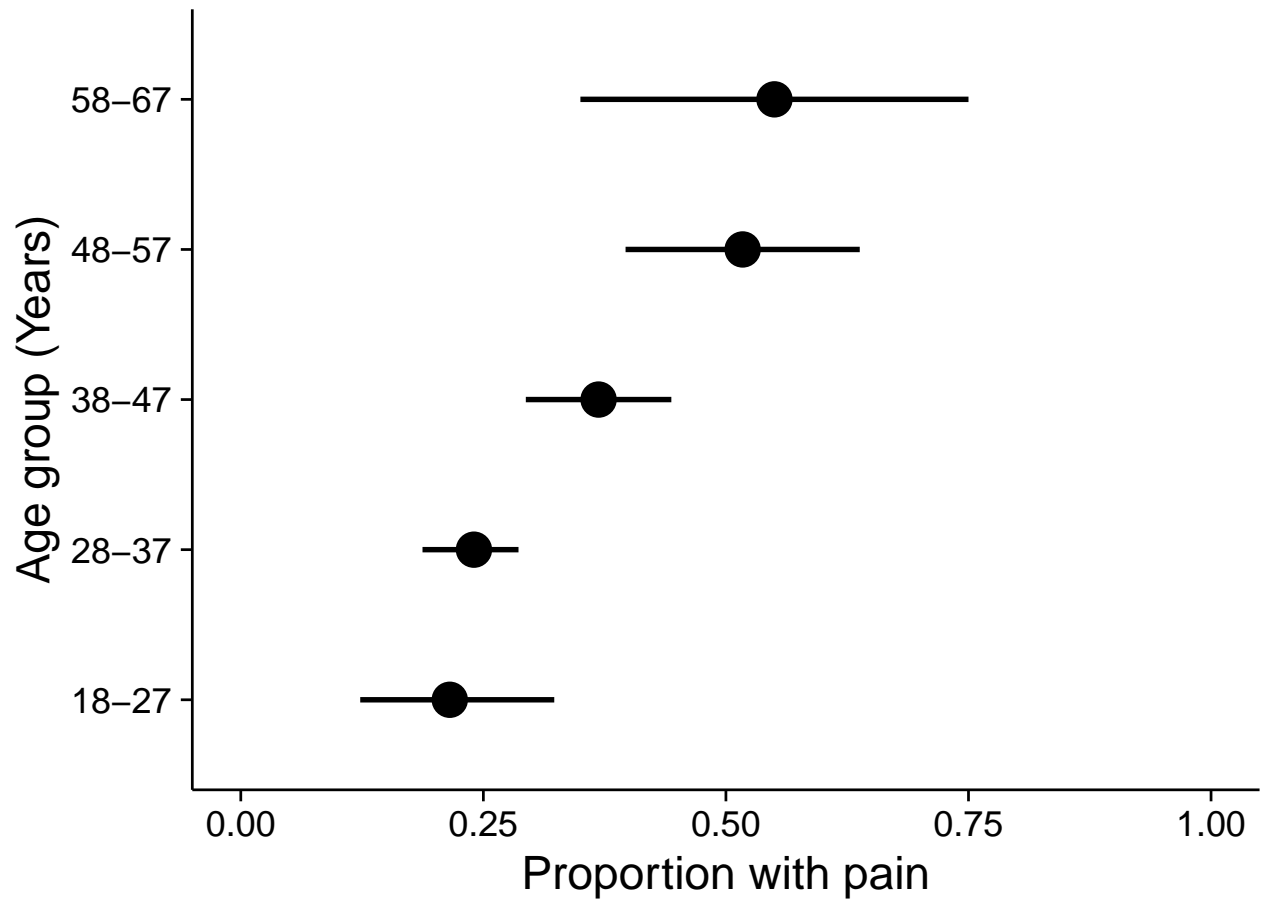Age group 68–77 years removed because n = 2

---
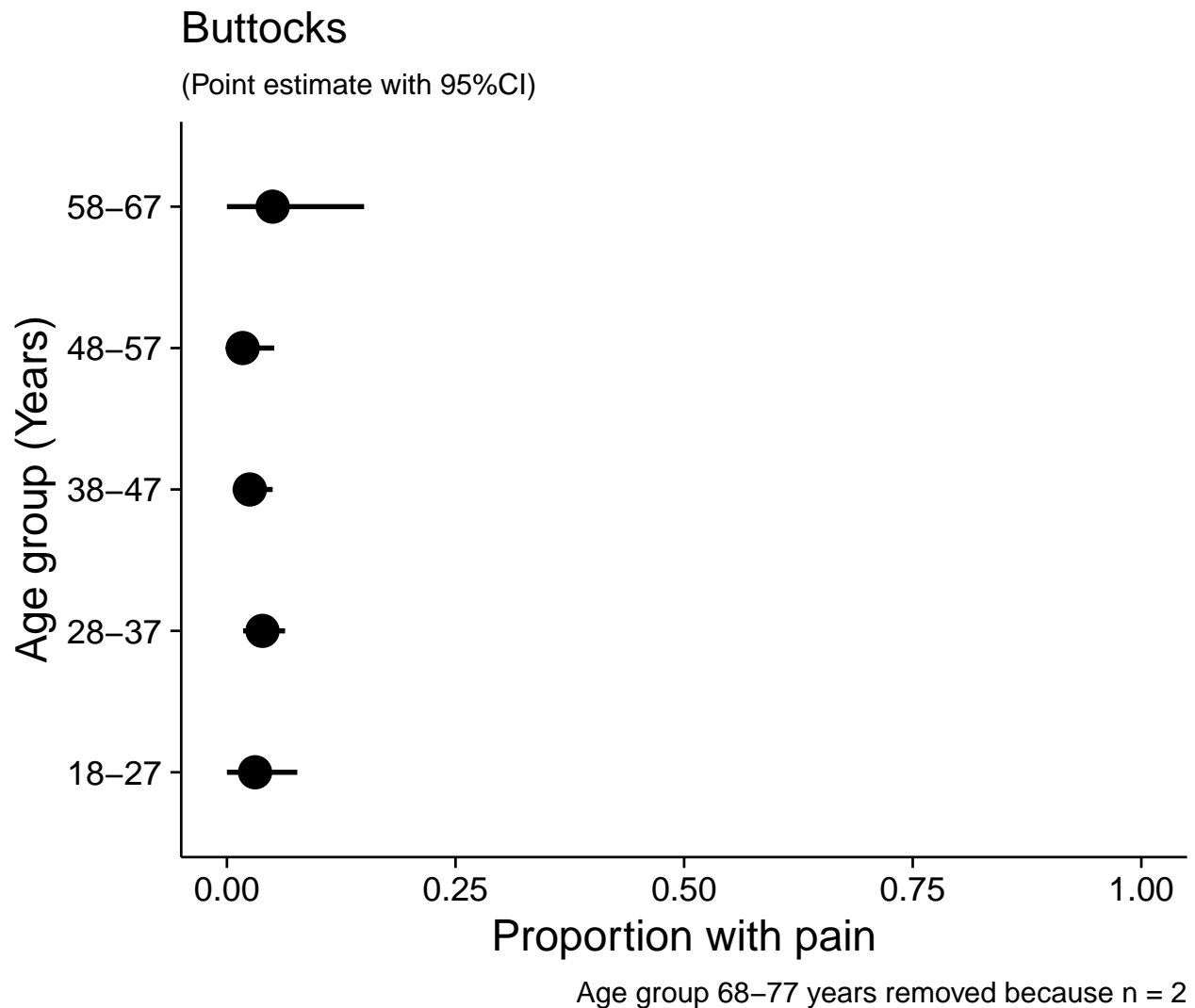
6 By most recent CD4 T-cell count

For ease of tabulation and plotting in this section of the supplement, I divided the most recent CD4 T-cell count into six categories. However for data analysis (see: Logistic regression, section 7), I analysed CD4 T-cell count as a continuous variable.

6.1 Process data

```
# Set seed
set.seed(2020)

# Select CD4 recent data
cd4 <- demo[, c('ID', 'CD4_recent')]

# Join to boot_data & remove ID, study site, and upper_back
# (only one outcome -- no pain)
cd4 <- left_join(data, cd4) %>%
```

```r
    select(-ID, -Site, -Upper_back)

# Get complete cases
cd4 <- cd4[complete.cases(cd4), ]

# Pivot and add CD4 recent group categories (counts of 100)
cd4_boot <- cd4 %>%
    # Pivot to long format
    pivot_longer(cols = -CD4_recent,
                 names_to = 'body_site',
                 values_to = 'pain_present') %>%
    # Add CD4 recent categories
    mutate(cd4_group = case_when(
        CD4_recent < 100 ~ '0-99',
        CD4_recent >= 100 & CD4_recent < 200 ~ '100-199',
        CD4_recent >= 200 & CD4_recent < 300 ~ '200-299',
        CD4_recent >= 300 & CD4_recent < 400 ~ '300-399',
        CD4_recent >= 400 & CD4_recent < 500 ~ '400-499',
        CD4_recent >= 500 ~ '500+'
    ))

# Print count per CD4 recent group
cd4_boot %>%
    group_by(body_site, cd4_group) %>%
    summarise(count = n()) %>%
    filter(body_site == 'Abdomen') %>%
    ungroup() %>%
    select(-body_site) %>%
    kable(caption = 'Participant count per CD4 group')
```

Table 37: Participant count per CD4 group

cd4_group	count
0-99	68
100-199	106
200-299	108
300-399	77
400-499	46
500+	92

```r
# Generate CIs
cd4_boot2 <- cd4_boot %>%
    # Remove CD4 recent
    select(-CD4_recent) %>%
    # Nest by CD4 recent group and body site
    group_by(cd4_group, body_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = prop_func,
                             R = 999,
```

```r
                                  stype = 'i',
                                  parallel = 'multicore',
                                  ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-data, -boot, -ci) %>%
    # Unnest
    unnest(cols = c(point_est, lower_ci, upper_ci)) %>%
    ungroup()

# Re-nest by body region and generate figures and tables
cd4_boot2 <- cd4_boot2 %>%
    # Fix site labels
    mutate(body_site = str_replace_all(body_site,
                                       pattern = '_',
                                       replacement = ' '),
           body_site = str_replace_all(body_site,
                                       pattern = '\\.',
                                       replacement = ' & ')) %>%
    # Group and nest
    group_by(body_site) %>%
    nest() %>%
    # Arrange CD4 recent groups
    # Plot data
    mutate(plots = map2(.x = data,
                        .y = body_site,
                        ~ .x %>%
                            ggplot(data = .) +
                            aes(x = cd4_group,
                                y = point_est,
                                ymin = lower_ci,
                                ymax = upper_ci) +
                            geom_linerange(size = 1,
                                           colour = '#000000') +
                            geom_point(colour = '#000000',
                                       size = 6) +
                            labs(title = .y,
                                 subtitle = '(Point estimate with 95%CI)',
                                 x = expression('CD4 group (cells.mm'^-3*')'),
                                 y = 'Proportion with pain') +
                            scale_y_continuous(limits = c(0, 1)) +
                            coord_flip() +
                            theme_minimal(base_size = 18) +
                            theme(plot.title = element_text(size = 18),
```

```
                              plot.subtitle = element_text(size = 12),
                              panel.grid = element_blank(),
                              axis.text = element_text(colour = '#000000'),
                              axis.line = element_line(size = 0.5),
                              axis.ticks = element_line(size = 0.5)))) %>%
    # Tabulate data
    mutate(tables = map2(.x = data,
                         .y = body_site,
                         ~ .x %>%
                             arrange(cd4_group) %>%
                             kable(caption = .y,
                                   digits = 2)))
```

6.2 Tabulated proportions (with 95% CIs), by CD4 (recent) group and body site

```
walk(cd4_boot2$tables, ~ print(.x))
```

Table 38: Head

cd4_group	point_est	lower_ci	upper_ci
0-99	0.26	0.16	0.37
100-199	0.40	0.30	0.49
200-299	0.32	0.24	0.42
300-399	0.36	0.25	0.48
400-499	0.30	0.17	0.43
500+	0.25	0.16	0.34

Table 39: Throat

cd4_group	point_est	lower_ci	upper_ci
0-99	0.07	0.01	0.15
100-199	0.02	0.00	0.05
200-299	0.03	0.00	0.06
300-399	0.01	0.00	0.04
400-499	0.04	0.00	0.11
500+	0.00	0.00	0.00

Table 40: Shoulder

cd4_group	point_est	lower_ci	upper_ci
0-99	0.07	0.01	0.15
100-199	0.10	0.05	0.17
200-299	0.06	0.02	0.11
300-399	0.05	0.01	0.10
400-499	0.15	0.04	0.26
500+	0.08	0.02	0.13

Table 41: Arms

cd4_group	point_est	lower_ci	upper_ci
0-99	0.06	0.01	0.12
100-199	0.08	0.03	0.14
200-299	0.00	0.00	0.00
300-399	0.04	0.00	0.09
400-499	0.02	0.00	0.07
500+	0.05	0.01	0.11

Table 42: Elbows

cd4_group	point_est	lower_ci	upper_ci
0-99	0.01	0.00	0.04
100-199	0.05	0.01	0.09
200-299	0.02	0.00	0.06
300-399	0.04	0.00	0.08
400-499	0.07	0.00	0.15
500+	0.07	0.02	0.12

Table 43: Wrists & Hands

cd4_group	point_est	lower_ci	upper_ci
0-99	0.01	0.00	0.04
100-199	0.06	0.02	0.10
200-299	0.06	0.02	0.10
300-399	0.06	0.01	0.12
400-499	0.07	0.00	0.13
500+	0.14	0.08	0.22

Table 44: Chest

cd4_group	point_est	lower_ci	upper_ci
0-99	0.34	0.22	0.46
100-199	0.23	0.15	0.31
200-299	0.17	0.10	0.23
300-399	0.17	0.09	0.26
400-499	0.17	0.09	0.28
500+	0.11	0.05	0.17

Table 45: Lower back

cd4_group	point_est	lower_ci	upper_ci
0-99	0.06	0.01	0.12
100-199	0.08	0.04	0.13
200-299	0.06	0.02	0.10
300-399	0.06	0.01	0.13
400-499	0.11	0.02	0.20

cd4_group	point_est	lower_ci	upper_ci
500+	0.10	0.04	0.16

Table 46: Abdomen

cd4_group	point_est	lower_ci	upper_ci
0-99	0.34	0.22	0.46
100-199	0.29	0.21	0.38
200-299	0.27	0.19	0.36
300-399	0.23	0.14	0.32
400-499	0.30	0.17	0.46
500+	0.22	0.13	0.30

Table 47: Cervical spine

cd4_group	point_est	lower_ci	upper_ci
0-99	0.04	0.00	0.10
100-199	0.09	0.04	0.15
200-299	0.02	0.00	0.05
300-399	0.06	0.01	0.13
400-499	0.09	0.02	0.17
500+	0.03	0.00	0.08

Table 48: Thoracic spine

cd4_group	point_est	lower_ci	upper_ci
0-99	0.13	0.06	0.22
100-199	0.11	0.06	0.18
200-299	0.10	0.05	0.16
300-399	0.16	0.08	0.25
400-499	0.22	0.11	0.35
500+	0.14	0.08	0.22

Table 49: Lumbosacral spine

cd4_group	point_est	lower_ci	upper_ci
0-99	0.09	0.03	0.16
100-199	0.14	0.08	0.22
200-299	0.15	0.08	0.22
300-399	0.18	0.10	0.29
400-499	0.30	0.17	0.43
500+	0.21	0.13	0.28

Table 50: Groin

cd4_group	point_est	lower_ci	upper_ci
0-99	0.10	0.03	0.18
100-199	0.12	0.07	0.19
200-299	0.09	0.05	0.16
300-399	0.12	0.05	0.19
400-499	0.11	0.02	0.20
500+	0.08	0.02	0.13

Table 51: Hips

cd4_group	point_est	lower_ci	upper_ci
0-99	0.01	0.00	0.06
100-199	0.06	0.02	0.10
200-299	0.05	0.01	0.09
300-399	0.05	0.01	0.10
400-499	0.11	0.02	0.20
500+	0.14	0.08	0.22

Table 52: Legs

cd4_group	point_est	lower_ci	upper_ci
0-99	0.13	0.06	0.22
100-199	0.11	0.06	0.18
200-299	0.15	0.08	0.21
300-399	0.18	0.10	0.27
400-499	0.20	0.09	0.30
500+	0.23	0.15	0.32

Table 53: Knees

cd4_group	point_est	lower_ci	upper_ci
0-99	0.12	0.04	0.19
100-199	0.11	0.06	0.18
200-299	0.15	0.08	0.21
300-399	0.12	0.05	0.19
400-499	0.17	0.07	0.28
500+	0.22	0.14	0.30

Table 54: Ankles & Feet

cd4_group	point_est	lower_ci	upper_ci
0-99	0.26	0.16	0.37
100-199	0.30	0.22	0.40
200-299	0.31	0.23	0.41
300-399	0.25	0.16	0.34

cd4_group	point_est	lower_ci	upper_ci
400-499	0.26	0.15	0.39
500+	0.35	0.25	0.45

Table 55: Buttocks

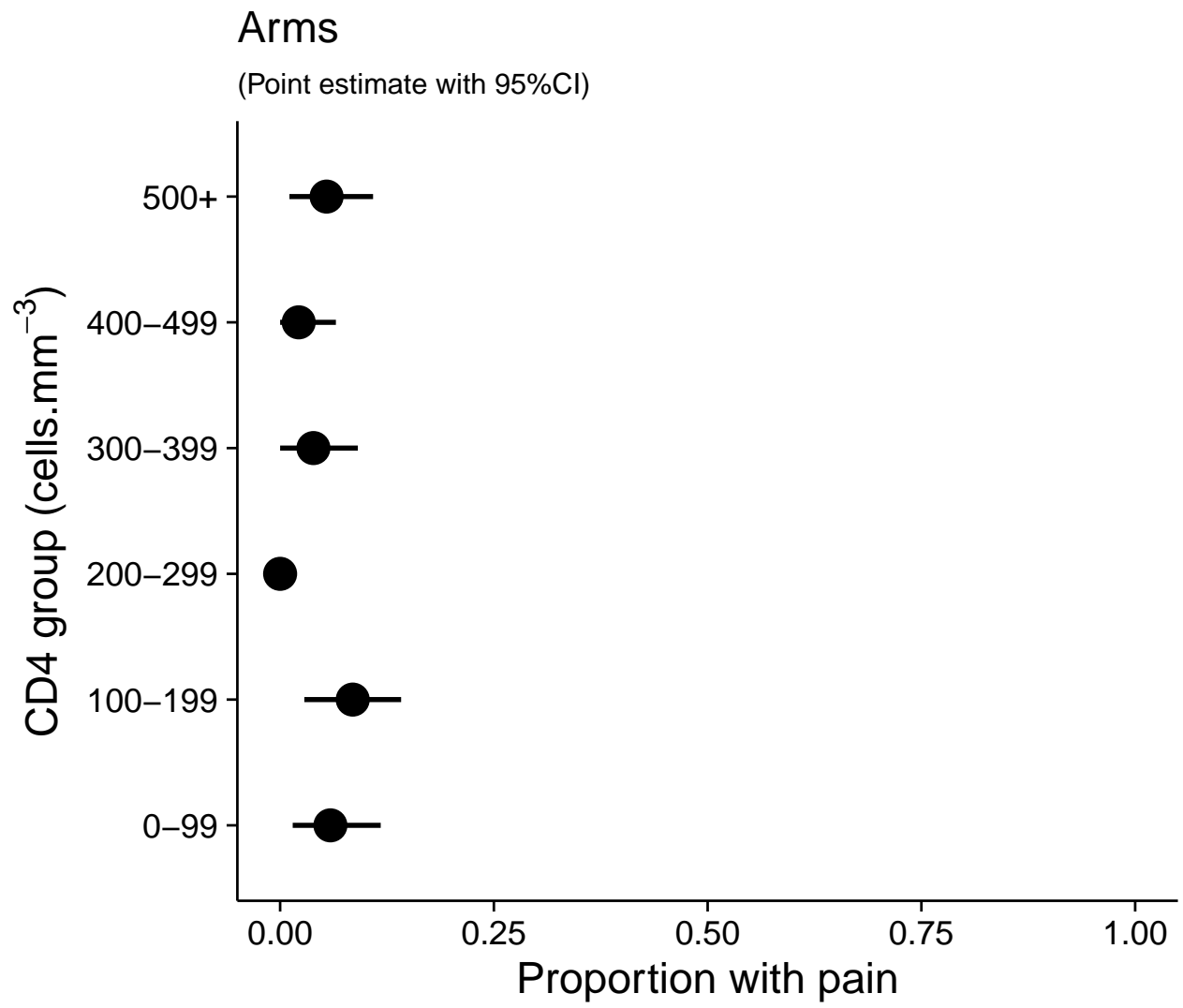| cd4_group	point_est	lower_ci	upper_ci
0-99	0.04	0.00	0.10
100-199	0.01	0.00	0.03
200-299	0.03	0.00	0.06
300-399	0.05	0.01	0.10
400-499	0.04	0.00	0.11
500+	0.07	0.02	0.12

6.3 Plotted proportions (with 95% CIs), by CD4 (recent) group and body site

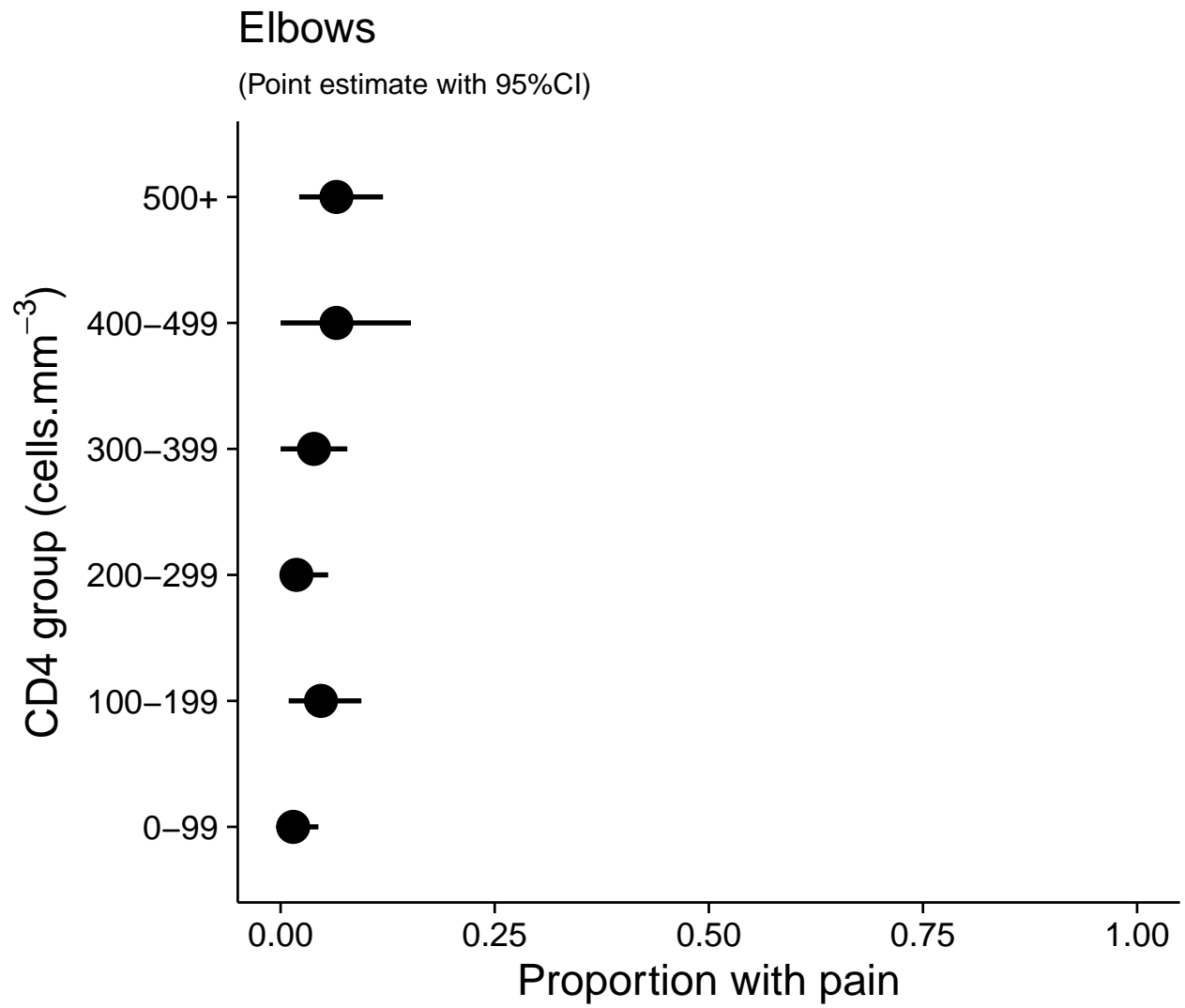```
walk(cd4_boot2$plots, ~ print(.x))
```



Head
(Point estimate with 95%CI)

**Throat**
(Point estimate with 95%CI)

Shoulder
(Point estimate with 95%CI)

## Arms
(Point estimate with 95%CI)

*Y-axis:* CD4 group (cells.mm$^{-3}$)

| | |
|---|---|
| 500+ | |
| 400–499 | |
| 300–399 | |
| 200–299 | |
| 100–199 | |
| 0–99 | |

*X-axis:* Proportion with pain

0.00    0.25    0.50    0.75    1.00

Elbows

(Point estimate with 95%CI)

# Wrists & Hands

(Point estimate with 95%CI)



CD4 group (cells.mm$^{-3}$)

Proportion with pain

Chest

(Point estimate with 95%CI)

Lower back
(Point estimate with 95%CI)

Abdomen
(Point estimate with 95%CI)
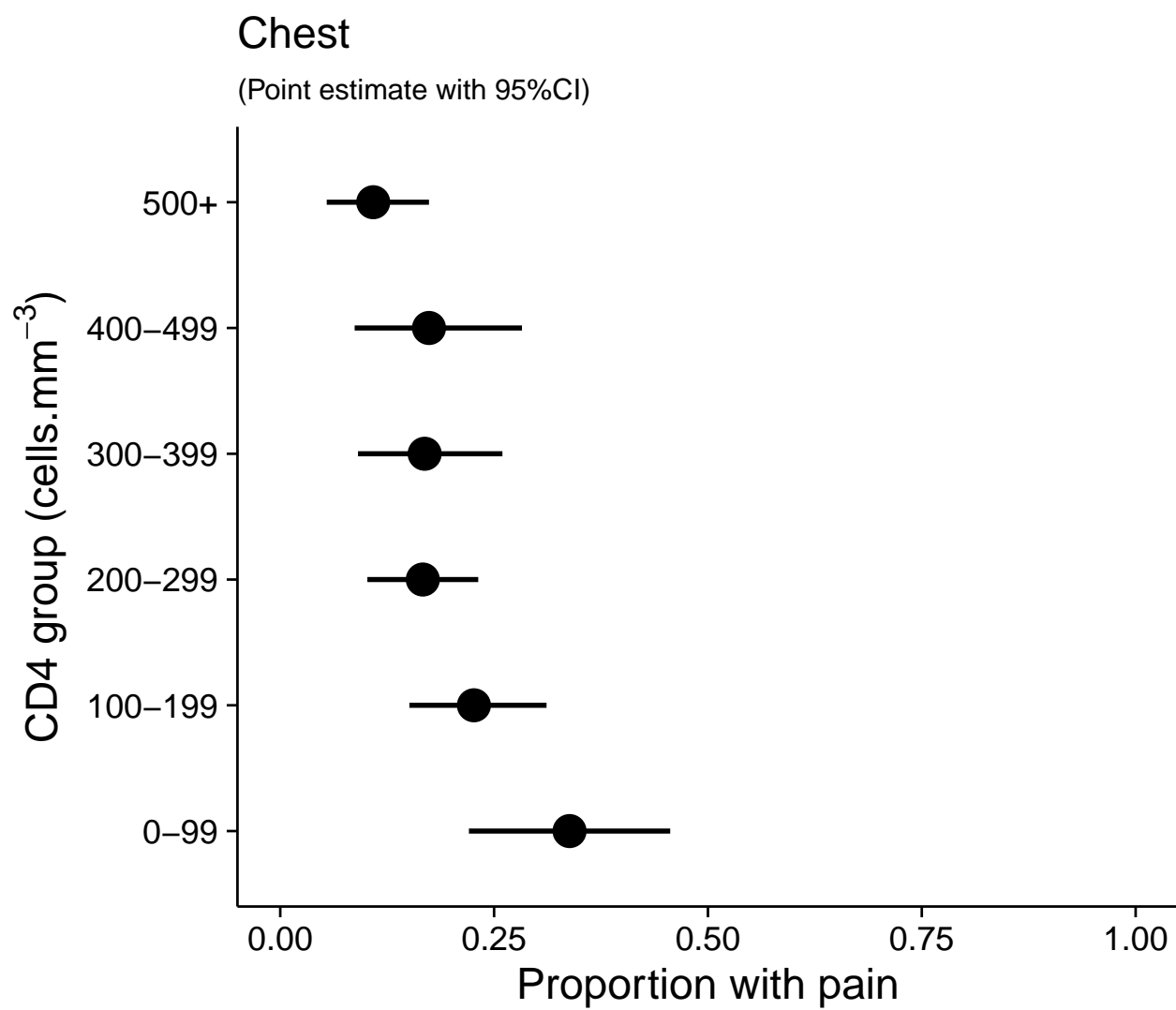
## Cervical spine
(Point estimate with 95%CI)

Thoracic spine
(Point estimate with 95%CI)

Lumbosacral spine
(Point estimate with 95%CI)

Groin

(Point estimate with 95%CI)

Hips
(Point estimate with 95%CI)

Legs
(Point estimate with 95%CI)

Knees

(Point estimate with 95%CI)

# Ankles & Feet
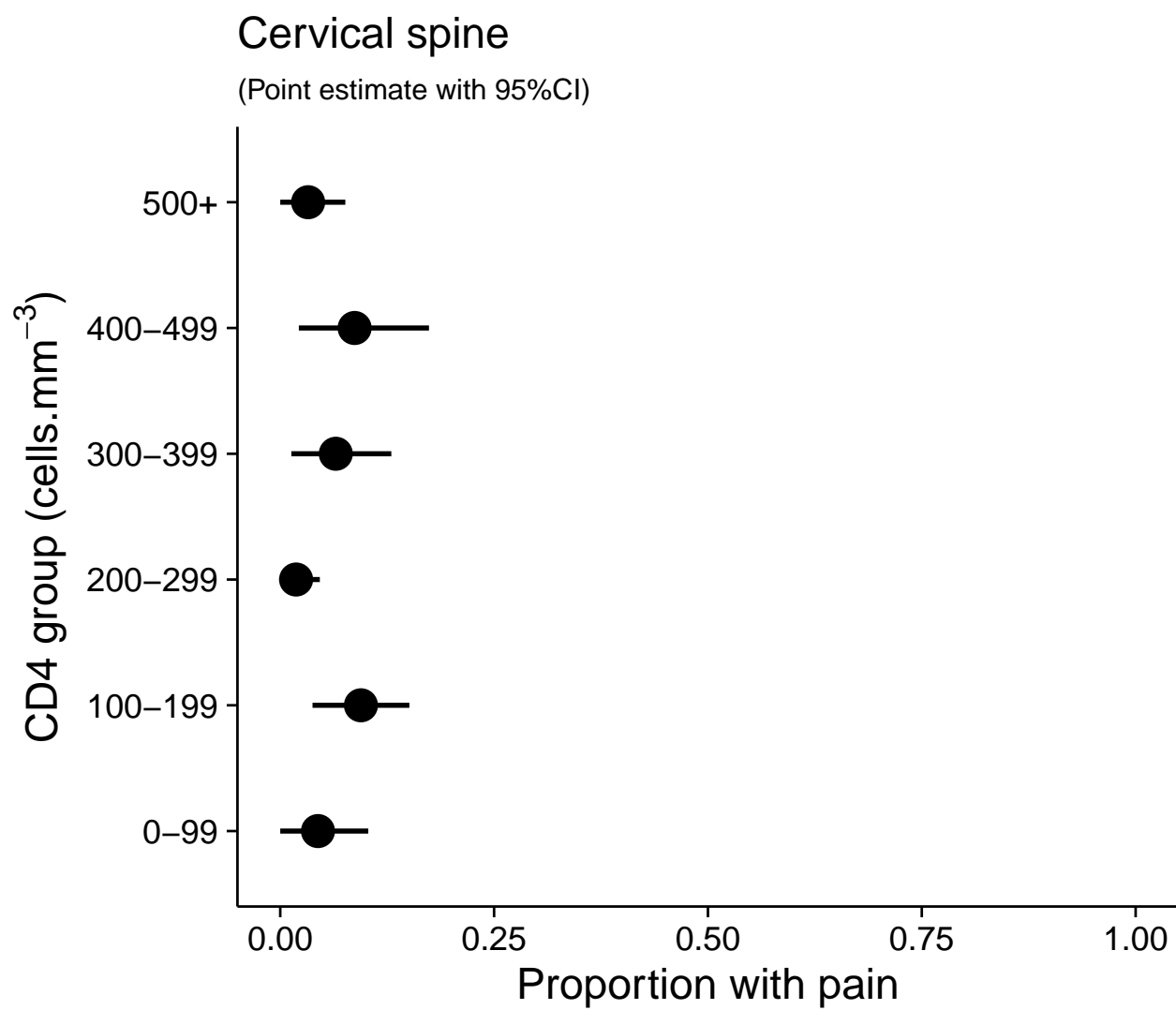
(Point estimate with 95%CI)



CD4 group (cells.mm$^{-3}$)

Proportion with pain

## Buttocks
(Point estimate with 95%CI)
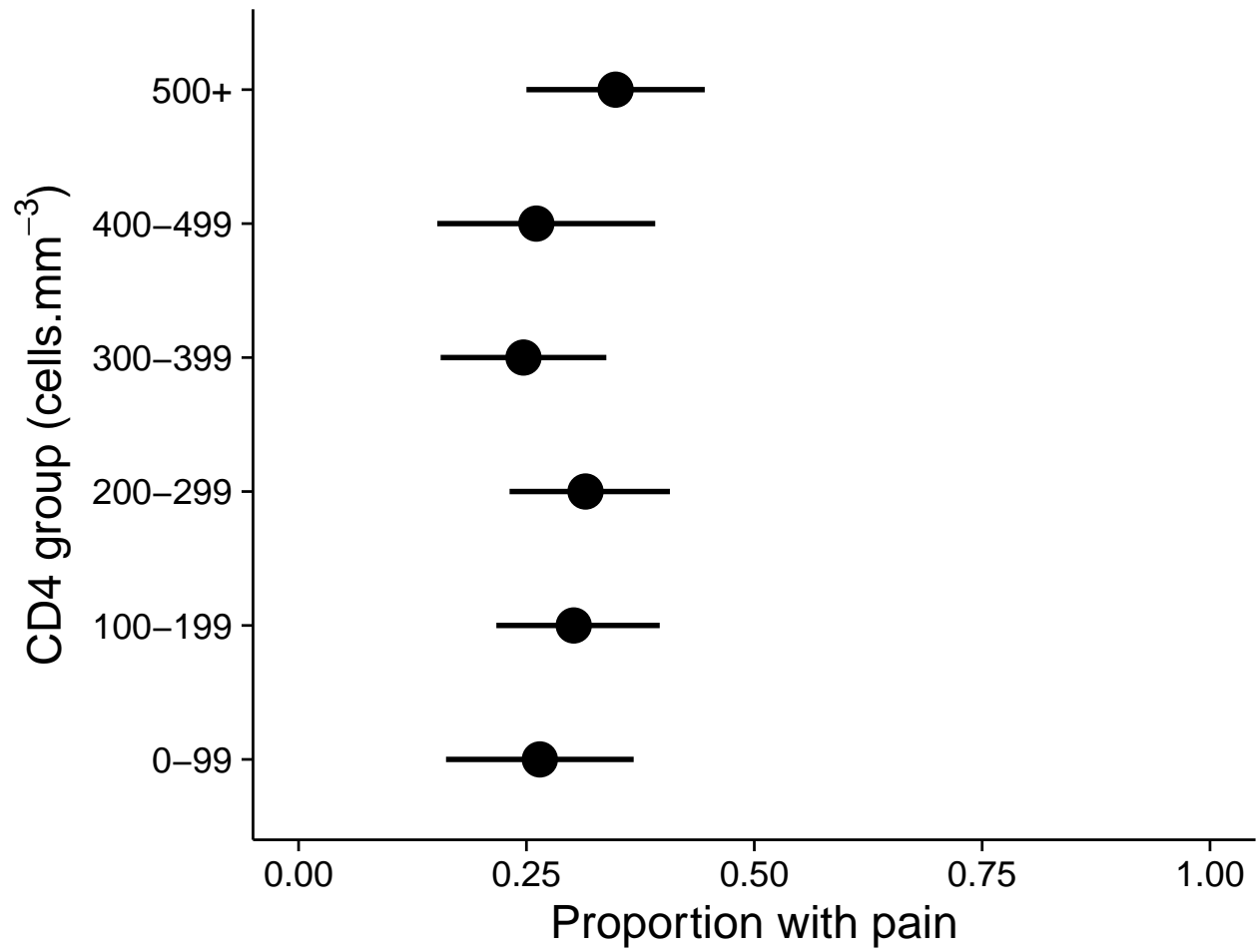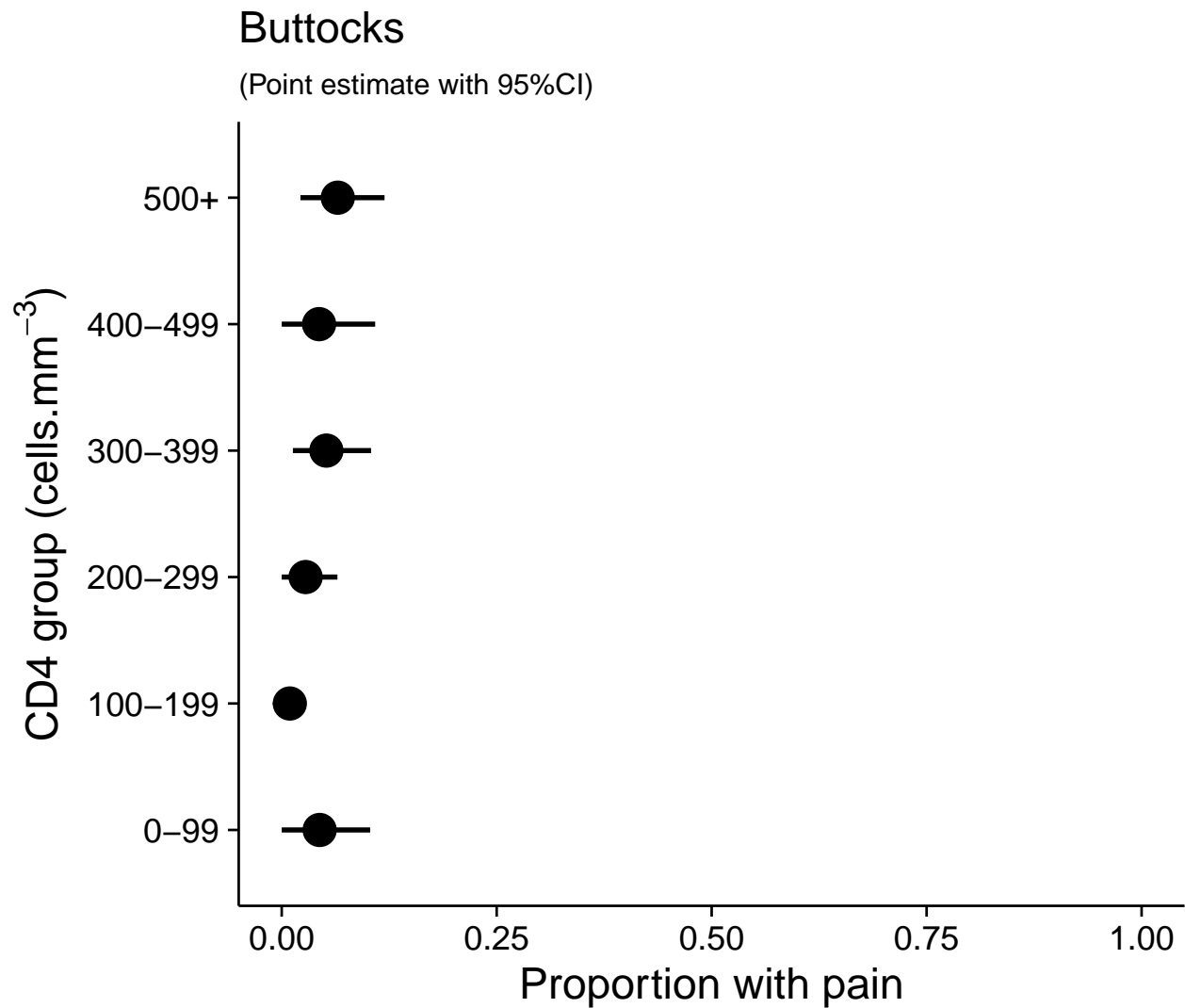
7 By pain definition

Pain in the last week or month (acute) versus persistent pain for the last 3 months (chronic)

## 7.1 Process data

```
# Set seed
set.seed(2020)

# Select pain data
pain <- demo[, c('ID', 'Pain_def')]

# Join to boot_data & remove ID, site, and upper_back (only one outcome -- no pain)
pain <- left_join(data, pain) %>%
    select(-ID, -Site, -Upper_back)

# Bootstrap CIs
```

```r
pain_boot <- pain %>%
    # Pivot to long format
    pivot_longer(cols = -Pain_def,
                 names_to = 'body_site',
                 values_to = 'pain_present') %>%
    # Add body regions
    mutate(region = case_when(
        body_site == 'Chest' |
            body_site == 'Head' |
            body_site == 'Throat' |
            body_site == 'Shoulder' ~ 'Head and upper torso',
        body_site == 'Lower_back' |
            body_site == 'Abdomen' |
            body_site == 'Hips' |
            body_site == 'Buttocks' |
            body_site == 'Groin' ~ 'Lower torso',
        body_site == 'Legs' |
            body_site == 'Knees' |
            body_site == 'Ankles.Feet' ~ 'Lower limbs',
        body_site == 'Arms' |
            body_site == 'Elbows' |
            body_site == 'Wrists.Hands' ~ 'Upper limbs',
        body_site == 'Cervical_spine' |
            body_site == 'Thoracic_spine' |
            body_site == 'Lumbosacral_spine' ~ 'Spinal column',
        TRUE ~ 'other'
    )) %>%
    # Nest by body region and body site
    group_by(Pain_def, region, body_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-data, -boot, -ci) %>%
    # Unnest
    unnest(cols = c(point_est, lower_ci, upper_ci))
```

```r
# Re-nest by body region and generate figures and tables
pain_boot2 <- pain_boot %>%
    group_by(region) %>%
    nest() %>%
    # Fix site labels
    mutate(data = map(.x = data,
                      ~ .x %>%
                        mutate(body_site = str_replace_all(body_site,
                                                           pattern = '_',
                                                           replacement = ' '),
                               body_site = str_replace_all(body_site,
                                                           pattern = '\\.',
                                                           replacement = ' & ')))) %>%
    # Re-order sites by point_est
    mutate(data = map(.x = data,
                      ~ .x %>%
                        mutate(body_site = fct_reorder(body_site,
                                                       point_est)))) %>%
    # Plot data
    mutate(plots = map2(.x = data,
                        .y = region,
                        ~ .x %>%
                          ggplot(data = .) +
                          aes(x = body_site,
                              y = point_est,
                              ymin = lower_ci,
                              ymax = upper_ci,
                              fill = Pain_def) +
                          geom_linerange(position = position_dodge2(width = 0.6),
                                         size = 1,
                                         colour = '#000000') +
                          geom_point(shape = 21,
                                     colour = '#000000',
                                     position = position_dodge2(width = 0.6),
                                     size = 6,
                                     stroke = 1) +
                          coord_flip() +
                          labs(title = .y,
                               subtitle = '(Point estimate with 95%CI)',
                               y = 'Proportion with pain') +
                          scale_y_continuous(limits = c(0, 1)) +
                          scale_fill_manual(values = c('#000000', '#FFFFFF')) +
                          theme_minimal(base_size = 18) +
                          theme(plot.title = element_text(size = 18),
                                plot.subtitle = element_text(size = 12),
                                legend.title = element_blank(),
                                legend.position = 'top',
                                axis.title.y = element_blank(),
                                panel.grid = element_blank(),
                                axis.text = element_text(colour = '#000000'),
                                axis.line = element_line(size = 0.5),
                                axis.ticks = element_line(size = 0.5)))) %>%
    # Tabulate data
```

```
    mutate(tables = map2(.x = data,
                         .y = region,
                         ~ .x %>%
                             kable(caption = .y,
                                   digits = 2)))
```

7.2 Tabulated proportions (with 95% CIs), by age and body region

```
walk(pain_boot2$tables, ~ print(.x))
```

Table 56: Head and upper torso

Pain_def	body_site	point_est	lower_ci	upper_ci
Acute	Head	0.38	0.33	0.43
Acute	Throat	0.03	0.01	0.05
Acute	Shoulder	0.08	0.05	0.11
Acute	Chest	0.24	0.20	0.28
Chronic	Head	0.23	0.18	0.30
Chronic	Throat	0.03	0.01	0.06
Chronic	Shoulder	0.08	0.04	0.11
Chronic	Chest	0.12	0.08	0.17

Table 57: Upper limbs

Pain_def	body_site	point_est	lower_ci	upper_ci
Acute	Arms	0.05	0.03	0.07
Acute	Elbows	0.03	0.02	0.05
Acute	Wrists & Hands	0.05	0.03	0.07
Chronic	Arms	0.02	0.00	0.05
Chronic	Elbows	0.05	0.02	0.08
Chronic	Wrists & Hands	0.08	0.04	0.11

Table 58: Lower torso

Pain_def	body_site	point_est	lower_ci	upper_ci
Acute	Lower back	0.07	0.04	0.09
Acute	Abdomen	0.30	0.26	0.35
Acute	Groin	0.11	0.08	0.15
Acute	Hips	0.02	0.01	0.04
Acute	Buttocks	0.03	0.02	0.05
Chronic	Lower back	0.08	0.04	0.11
Chronic	Abdomen	0.20	0.15	0.26
Chronic	Groin	0.05	0.02	0.08
Chronic	Hips	0.15	0.11	0.20
Chronic	Buttocks	0.03	0.01	0.06

Table 59: Spinal column

Pain_def	body_site	point_est	lower_ci	upper_ci
Acute	Cervical spine	0.07	0.05	0.10
Acute	Thoracic spine	0.14	0.11	0.18
Acute	Lumbosacral spine	0.14	0.11	0.18
Chronic	Cervical spine	0.01	0.00	0.02
Chronic	Thoracic spine	0.10	0.06	0.13
Chronic	Lumbosacral spine	0.18	0.13	0.23

Table 60: Lower limbs

Pain_def	body_site	point_est	lower_ci	upper_ci
Acute	Legs	0.12	0.09	0.15
Acute	Knees	0.10	0.07	0.13
Acute	Ankles & Feet	0.23	0.19	0.27
Chronic	Legs	0.25	0.19	0.32
Chronic	Knees	0.22	0.16	0.27
Chronic	Ankles & Feet	0.46	0.39	0.53

7.3 Plotted proportions (with 95% CIs), by age and body region

```
walk(pain_boot2$plots, ~ print(.x))
```
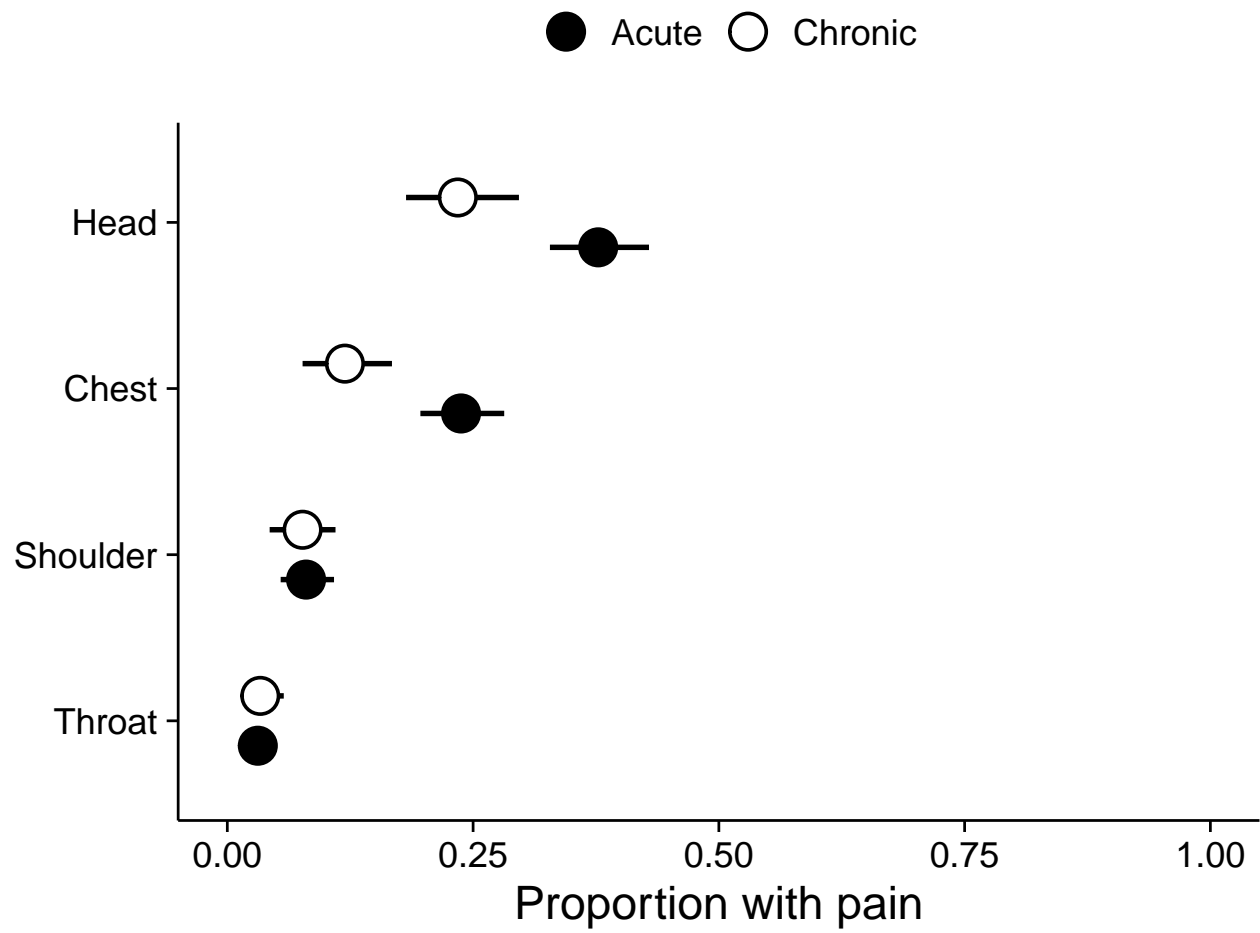
Head and upper torso

(Point estimate with 95%CI)

Upper limbs

(Point estimate with 95%CI)

Lower torso

(Point estimate with 95%CI)

Spinal column

(Point estimate with 95%CI)

● Acute ○ Chronic

Proportion with pain

Lower limbs
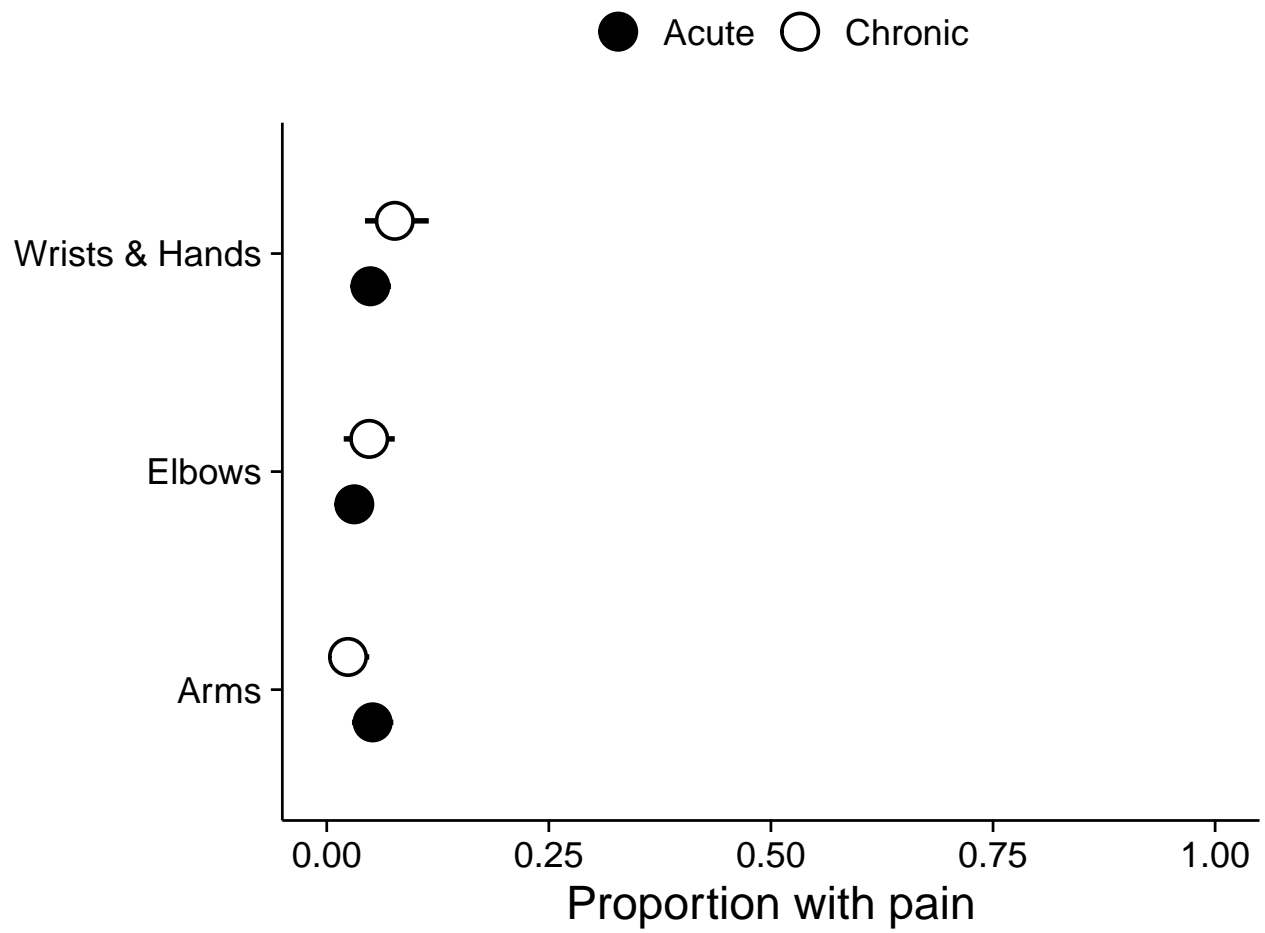
(Point estimate with 95%CI)

● Acute ○ Chronic
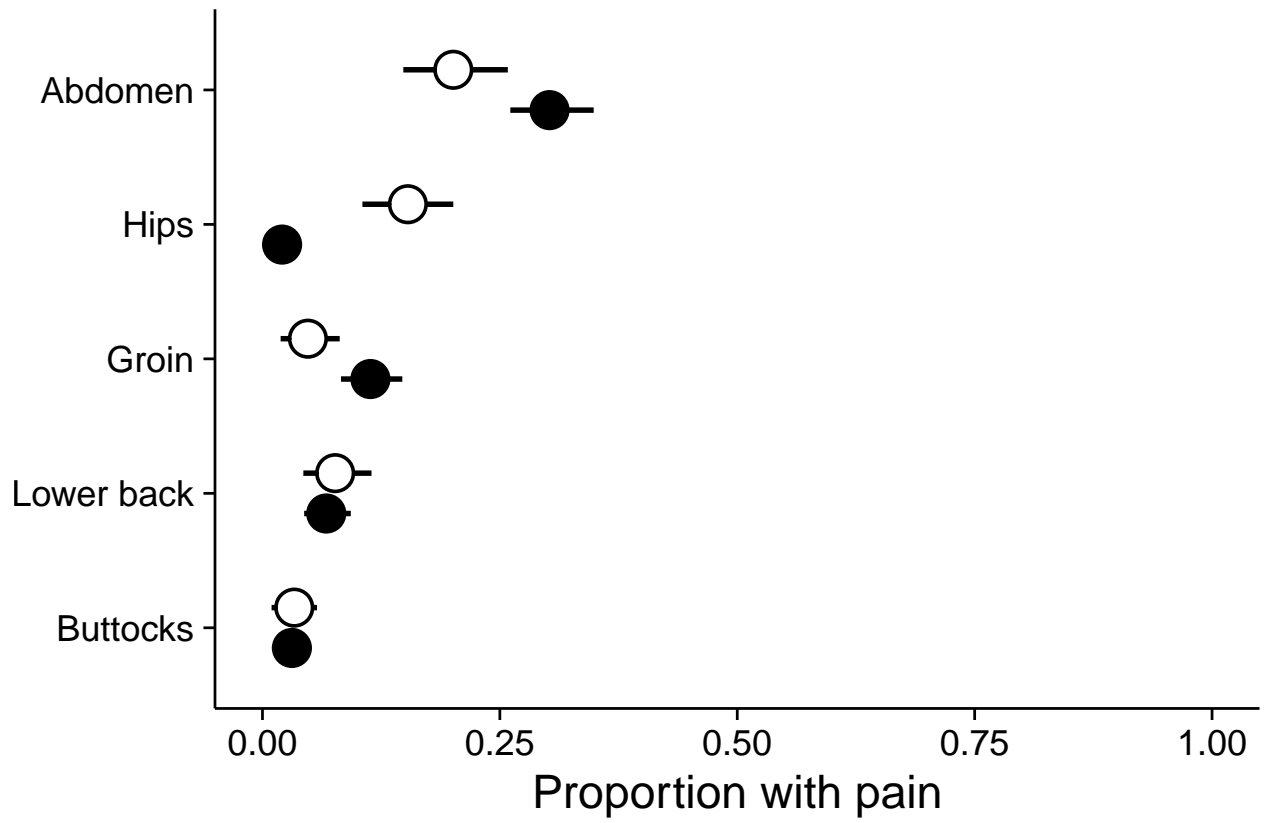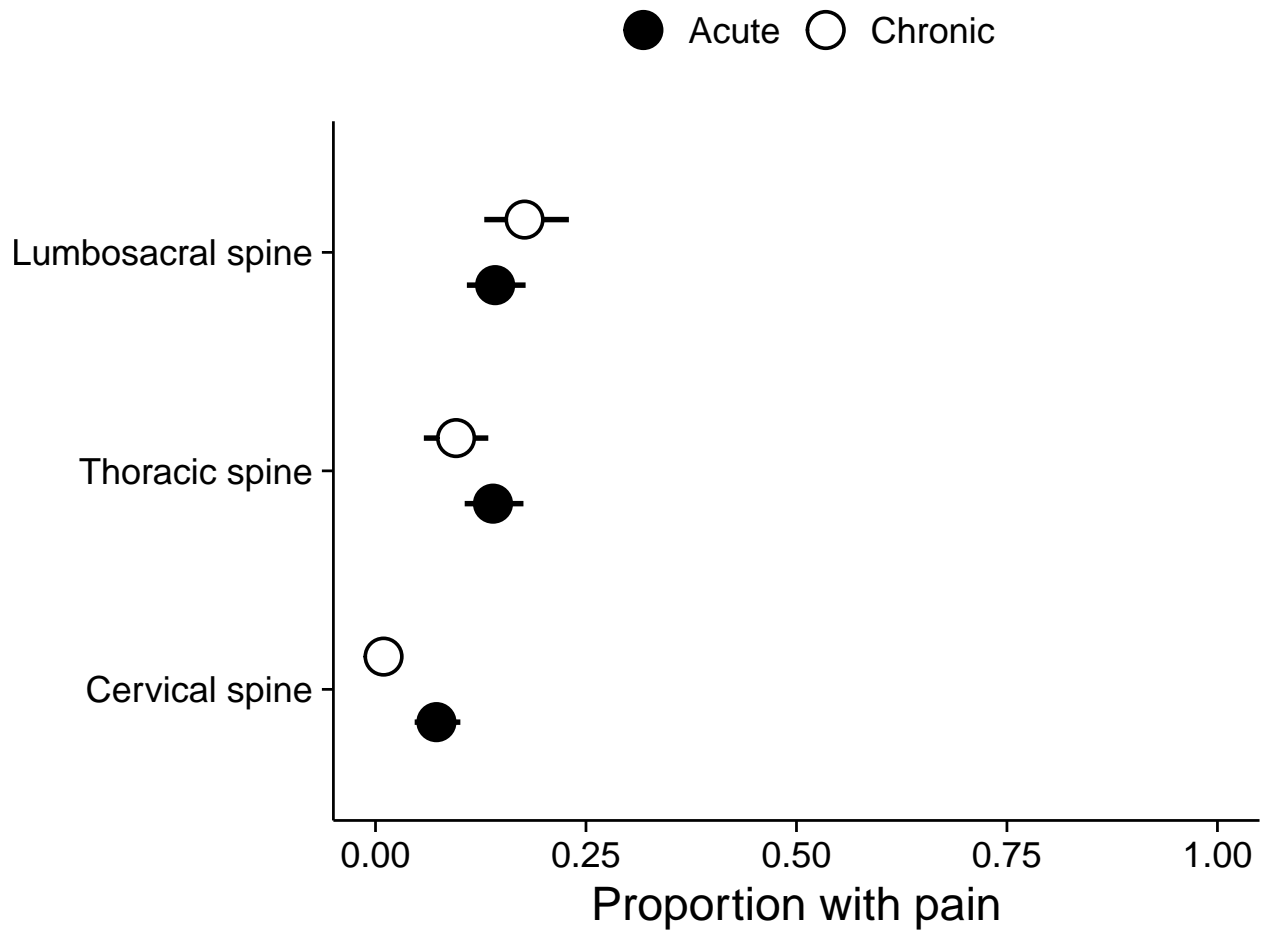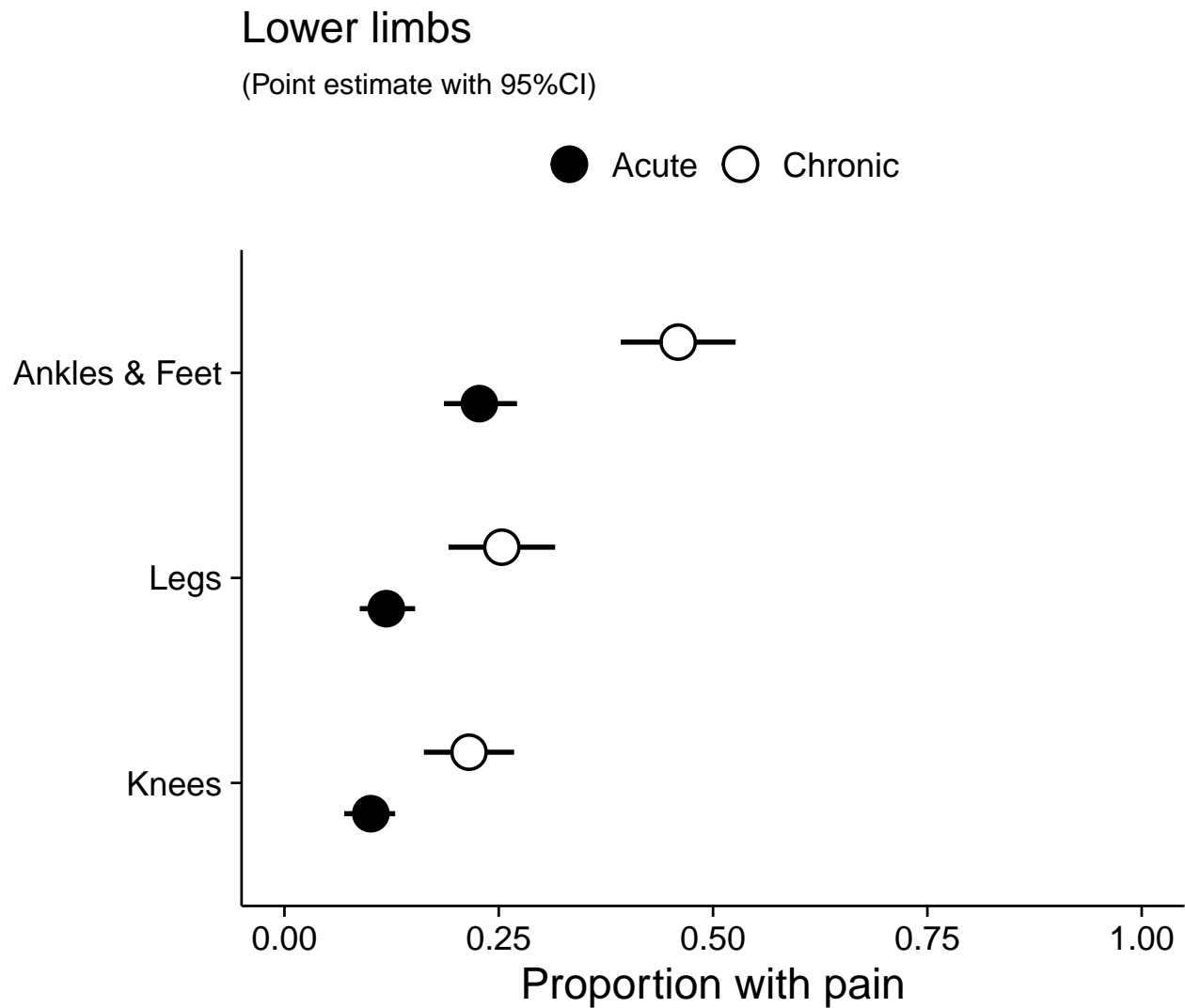
8 Logistic regression

Note: Age and CD4 T-cell count treated as a continuous variables.

Note: Age and CD-4 T-cell count were scaled and centered.

8.1 Process data

Extract and process data on study site, participant ID, CD4 T-cell count, age, sex, and pain definition.

```
# Select demographic data
demo_logit <- demo[, c('ID', 'Site', 'CD4_recent', 'Sex', 'Age', 'Pain_def')]

# Process data
data_logit <- data %>%
    # Join with extracted demographics data (demo_reduced)
    left_join(demo_logit) %>%
    # Remove upper back (only one outcome -- no pain) and ID
```

```
    select(-ID, -Upper_back)

# Convert data_glm to long format
data_logit.long <- data_logit %>%
    pivot_longer(cols = -c(CD4_recent, Age, Sex, Site, Pain_def),
                 names_to = 'Pain_site',
                 values_to = 'Pain_present')

# Scale Age and CD4_recent
data_logit.long <- data_logit.long %>%
    mutate_if(is.numeric, scale)

# Dummy code outcome variable
data_logit.long <- data_logit.long %>%
    mutate(Pain_present = ifelse(Pain_present == 'No',
                                 yes = 0,
                                 no = 1))
```

8.2   Family-wise error correction

Alpha threshold for significance was corrected to control the family-wise error rate. The correction was applied to the results of the likelihood ratio tests comparing null models to full models. We chose the conservative Dunn-Bonferroni correction.

8.3   Run `glmer` model for every body site

Note: p-values reported for the likelihood ratio tests below are uncorrected for multiple comparisons.

```
#-- Head --#
head <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Head') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
                  data = head,
                  family = binomial())

## Full model
head_mod <- glmer(Pain_present ~
                      Age +
                      Sex +
                      CD4_recent +
                      Pain_def +
                      (1|Site),
                  data = head,
                  family = binomial())

## Compare models
anova(null_mod, head_mod, test = 'LRT')

## Data: head
## Models:
```

```
## null_mod: Pain_present ~ 1 + (1 | Site)
## head_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## null_mod    2 599.13 607.53 -297.56   595.13
## head_mod    6 589.86 615.06 -288.93   577.86 17.27  4   0.001713 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Significant
## (family-wise error correction, alpha = 0.0024)
## Print model
summary(head_mod)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: head
##
##      AIC      BIC   logLik deviance df.resid
##    589.9    615.1   -288.9    577.9      487
##
## Scaled residuals:
##     Min     1Q  Median     3Q     Max
## -1.0014 -0.7480 -0.4873  1.0605  2.6634
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.1263   0.3553
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.39255    0.28339  -1.385  0.16599
## Age             0.03543    0.12234   0.290  0.77213
## SexMale        -1.34687    0.38448  -3.503  0.00046 ***
## CD4_recent     -0.11133    0.11511  -0.967  0.33344
## Pain_defChronic -0.54907   0.42960  -1.278  0.20121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal CD4_rc
## Age         0.186
## SexMale    -0.094 -0.103
## CD4_recent  0.068 -0.071  0.081
## Pan_dfChrnc -0.685 -0.238 -0.042 -0.203
#-- Throat --#
throat <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Throat') %>%
    filter(complete.cases(.))

## Null
```

```
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
              data = throat,
              family = binomial())

## Full model
thrt_mod <- glmer(Pain_present ~
              Age +
              Sex +
              CD4_recent +
              Pain_def +
              (1|Site),
            data = throat,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(thrt_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##    Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: throat
##
##      AIC      BIC   logLik deviance df.resid
##    125.0    150.2    -56.5    113.0      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.3660 -0.1799 -0.1466 -0.1094  8.5792
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value        Pr(>|z|)
## (Intercept)     -4.12952    0.46370  -8.905 <0.0000000000000002 ***
## Age              0.07201    0.33111   0.217          0.8278
## SexMale          0.91911    0.62751   1.465          0.1430
## CD4_recent      -0.78730    0.45238  -1.740          0.0818 .
## Pain_defChronic  0.19658    0.68881   0.285          0.7753
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.160
## SexMale     -0.330 -0.224
## CD4_recent   0.512 -0.132  0.168
## Pan_dfChrnc -0.475 -0.240 -0.211 -0.268
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
#-- Shoulder --#
shoulder <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Shoulder') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = shoulder,
            family = binomial())

## Full model
shdr_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = shoulder,
            family = binomial())

## Compare models
anova(null_mod, shdr_mod, test = 'LRT')
```

```
## Data: shoulder
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## shdr_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod    2 286.42 294.82 -141.21   282.42
## shdr_mod    6 292.25 317.45 -140.12   280.25 2.1705  4     0.7044
```

```
## No significant difference between null and full model
## Print model
## Print model
summary(shdr_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: shoulder
##
##      AIC      BIC   logLik deviance df.resid
##    292.2    317.5   -140.1    280.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.4264 -0.3111 -0.2893 -0.2690  4.2089
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.01967  0.1403
## Number of obs: 493, groups:  Site, 5
##
```

```
## Fixed effects:
##               Estimate Std. Error z value         Pr(>|z|)
## (Intercept)   -2.56646    0.25193 -10.187 <0.0000000000000002 ***
## Age           -0.14999    0.20336  -0.738             0.461
## SexMale        0.53246    0.43017   1.238             0.216
## CD4_recent    -0.06105    0.18801  -0.325             0.745
## Pain_defChronic 0.10528   0.46037   0.229             0.819
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal CD4_rc
## Age         0.362
## SexMale    -0.293 -0.137
## CD4_recent  0.111 -0.131  0.122
## Pan_dfChrnc -0.529 -0.368 -0.252 -0.199
```

```r
#-- Arm --#
arms <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Arms') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = arms,
            family = binomial())

## Full model
arms_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = arms,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(arms_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: arms
##
##      AIC      BIC   logLik deviance df.resid
##    189.1    214.3    -88.6    177.1      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.4341 -0.2334 -0.2115 -0.1835  5.5338
##
```

```
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value          Pr(>|z|)
## (Intercept)     -2.89348    0.27466 -10.535 <0.0000000000000002 ***
## Age              0.30866    0.21783   1.417             0.156
## SexMale         -0.05672    0.59049  -0.096             0.923
## CD4_recent      -0.10059    0.25597  -0.393             0.694
## Pain_defChronic -0.62336    0.59010  -1.056             0.291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal CD4_rc
## Age         0.144
## SexMale    -0.310 -0.207
## CD4_recent  0.222 -0.093  0.123
## Pan_dfChrnc -0.468 -0.297 -0.101 -0.317
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Elbow --#
elbow <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Elbows') %>%
    filter(complete.cases(.))


## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = elbow,
            family = binomial())


## Full model
elbw_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = elbow,
            family = binomial())


## Fit is singular, print model to check SD of random effect
## Print model
summary(elbw_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##     Data: elbow
##
```

```
##      AIC      BIC   logLik deviance df.resid
##    172.1    197.3    -80.1    160.1      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.4574 -0.2127 -0.1764 -0.1533  6.7009
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value           Pr(>|z|)
## (Intercept)     -3.38921    0.33556 -10.100 <0.0000000000000002 ***
## Age              0.42521    0.21018   2.023             0.0431 *
## SexMale         -0.16645    0.59159  -0.281             0.7784
## CD4_recent       0.04229    0.23571   0.179             0.8576
## Pain_defChronic  0.45074    0.53643   0.840             0.4008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.023
## SexMale     -0.244 -0.138
## CD4_recent   0.103 -0.100  0.138
## Pan_dfChrnc -0.595 -0.316 -0.115 -0.352
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Wrist & hand --#
hand <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Wrists.Hands') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = hand,
            family = binomial())

## Full model
hand_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = hand,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
```

```
summary(hand_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: hand
##
##      AIC      BIC   logLik deviance df.resid
##    243.3    268.5   -115.6    231.3      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.6343 -0.2735 -0.2284 -0.2038  6.8403
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                  Estimate Std. Error z value        Pr(>|z|)
## (Intercept)      -2.7754     0.2588 -10.722 <0.0000000000000002 ***
## Age               0.2316     0.1805   1.283           0.199
## SexMale          -0.6507     0.5663  -1.149           0.251
## CD4_recent        0.2766     0.1711   1.616           0.106
## Pain_defChronic   0.4408     0.4376   1.007           0.314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.191
## SexMale     -0.208 -0.145
## CD4_recent  -0.005 -0.020  0.126
## Pan_dfChrnc -0.601 -0.377 -0.100 -0.374
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
#-- Chest --#
chest <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Chest') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = chest,
            family = binomial())

## Full model
chst_mod <- glmer(Pain_present ~
                Age +
                Sex +
```

```
                    CD4_recent +
                    Pain_def +
                    (1|Site),
             data = chest,
             family = binomial())

## Compare models
anova(null_mod, chst_mod, test = 'LRT')
```

```
## Data: chest
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## chst_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod    2 479.88 488.28 -237.94   475.88
## chst_mod    6 480.58 505.78 -234.29   468.58 7.3024  4     0.1207
```

```
## No significant difference in null and full models
## Print model
## Print model
summary(chst_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##     Data: chest
##
##      AIC      BIC   logLik deviance df.resid
##    480.6    505.8   -234.3    468.6      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7467 -0.5449 -0.4222 -0.3084  4.7140
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.04797  0.219
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value      Pr(>|z|)
## (Intercept)     -1.30612    0.22142  -5.899 0.00000000366 ***
## Age              0.09422    0.13852   0.680        0.4964
## SexMale          0.04311    0.32631   0.132        0.8949
## CD4_recent      -0.27859    0.14976  -1.860        0.0628 .
## Pain_defChronic -0.75437    0.38896  -1.939        0.0524 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.254
## SexMale     -0.150 -0.107
## CD4_recent   0.132 -0.151  0.064
```

```
## Pan_dfChrnc -0.562 -0.253 -0.118 -0.213
```

```r
#-- Abdomen --#
abdomen <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Abdomen') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = abdomen,
            family = binomial())

## Full model
abdm_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = abdomen,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(abdm_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: abdomen
##
##      AIC      BIC   logLik deviance df.resid
##    581.2    606.4   -284.6    569.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8003 -0.6690 -0.5480  1.3841  2.3391
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value      Pr(>|z|)
## (Intercept)      -0.8813     0.1365  -6.458 0.000000000106 ***
## Age              -0.1613     0.1219  -1.323          0.186
## SexMale          -0.3072     0.3015  -1.019          0.308
## CD4_recent       -0.1145     0.1154  -0.993          0.321
## Pain_defChronic  -0.2542     0.2706  -0.940          0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.403
## SexMale     -0.275 -0.183
## CD4_recent   0.170 -0.038  0.157
## Pan_dfChrnc -0.537 -0.333 -0.143 -0.324
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Lower back (flank) --#
low_back <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Lower_back') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
                data = low_back,
                family = binomial())

## Full model
back_mod <- glmer(Pain_present ~
                    Age +
                    Sex +
                    CD4_recent +
                    Pain_def +
                    (1|Site),
                data = low_back,
                family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(back_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
## Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: low_back
##
##      AIC      BIC   logLik deviance df.resid
##    272.1    297.4   -130.1    260.1      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.4014 -0.2965 -0.2655 -0.2528  4.3683
##
## Random effects:
##  Groups Name        Variance                Std.Dev.
##  Site   (Intercept) 0.0000000000000001446 0.00000001202
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##               Estimate Std. Error z value          Pr(>|z|)
## (Intercept)   -2.59625    0.23601 -11.001 <0.0000000000000002 ***
```

87

```
## Age              0.10656    0.18068   0.590              0.555
## SexMale         -0.23426    0.48221  -0.486              0.627
## CD4_recent       0.08667    0.17452   0.497              0.619
## Pain_defChronic  0.33885    0.41624   0.814              0.416
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.259
## SexMale     -0.246 -0.160
## CD4_recent   0.080 -0.060  0.161
## Pan_dfChrnc -0.591 -0.357 -0.141 -0.355
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Groin --#
groin <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Groin') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
               data = groin,
               family = binomial())

## Full model
groin_mod <- glmer(Pain_present ~
                   Age +
                   Sex +
                   CD4_recent +
                   Pain_def +
                   (1|Site),
               data = groin,
               family = binomial())

## Compare models
anova(null_mod, groin_mod, test = 'LRT')
```

```
## Data: groin
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## groin_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##           npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod     2 304.54 312.94 -150.27   300.54
## groin_mod    6 308.23 333.43 -148.12   296.23 4.3039  4     0.3664
```

```r
## No significant difference between null and full model
## Print model
summary(groin_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
```

```
##    Data: groin
##
##      AIC      BIC   logLik deviance df.resid
##    308.2    333.4   -148.1    296.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7049 -0.4182 -0.2169 -0.1479  7.0102
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.9946   0.9973
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.57300    0.77458  -3.322 0.000894 ***
## Age             -0.05428    0.16923  -0.321 0.748400
## SexMale         -0.63053    0.44135  -1.429 0.153105
## CD4_recent       0.20593    0.16740   1.230 0.218616
## Pain_defChronic -0.41363    1.05390  -0.392 0.694708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.108
## SexMale     -0.053 -0.162
## CD4_recent   0.001 -0.081  0.066
## Pan_dfChrnc -0.727 -0.126 -0.019 -0.141
```

```r
#-- Buttocks --#
buttocks <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Buttocks') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = buttocks,
            family = binomial())

## Full model
butt_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = buttocks,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
```

```
summary(butt_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##    Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: buttocks
##
##      AIC       BIC   logLik deviance df.resid
##    170.7     195.9    -79.4    158.7      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.3403 -0.2151 -0.1860 -0.1710  6.1831
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                  Estimate Std. Error z value         Pr(>|z|)
## (Intercept)      -3.42677    0.33725 -10.161 <0.0000000000000002 ***
## Age              -0.12939    0.26949  -0.480            0.631
## SexMale           0.65435    0.57735   1.133            0.257
## CD4_recent        0.23689    0.22789   1.039            0.299
## Pain_defChronic   0.09589    0.57052   0.168            0.867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.379
## SexMale     -0.386 -0.244
## CD4_recent  -0.038 -0.077  0.219
## Pan_dfChrnc -0.512 -0.318 -0.169 -0.344
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
#-- Hips --#
hips <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Hips') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = hips,
            family = binomial())

## Full model
hips_mod <- glmer(Pain_present ~
                Age +
                Sex +
```

```
                CD4_recent +
                Pain_def +
                (1|Site),
          data = hips,
          family = binomial())

## Compare models
anova(null_mod, hips_mod, test = 'LRT')

## Data: hips
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## hips_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod    2 224.25 232.65 -110.12   220.25
## hips_mod    6 225.28 250.49 -106.64   213.28 6.9633  4     0.1378

## No significant difference between null and full model
## Print model
summary(hips_mod)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: hips
##
##      AIC      BIC   logLik deviance df.resid
##    225.3    250.5   -106.6    213.3      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.6294 -0.3457 -0.1553 -0.1426  7.1926
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.06768  0.2601
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value           Pr(>|z|)
## (Intercept)     -3.74770    0.41409  -9.050 < 0.0000000000000002 ***
## Age              0.05564    0.19046   0.292           0.770190
## SexMale          0.31715    0.43874   0.723           0.469760
## CD4_recent       0.12278    0.17504   0.701           0.483039
## Pain_defChronic  1.88246    0.55730   3.378           0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.155
## SexMale     -0.157 -0.117
## CD4_recent   0.006 -0.110  0.243
## Pan_dfChrnc -0.717 -0.284 -0.115 -0.217
```

```
#-- Legs --#
legs <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Legs') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = legs,
            family = binomial())

## Full model
legs_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = legs,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(legs_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: legs
##
##      AIC      BIC   logLik deviance df.resid
##    431.0    456.2   -209.5    419.0      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8089 -0.4651 -0.3498 -0.3251  3.2001
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value         Pr(>|z|)
## (Intercept)      -2.0238     0.1812 -11.170 <0.0000000000000002 ***
## Age               0.1117     0.1305   0.856          0.3920
## SexMale           0.6088     0.3013   2.020          0.0433 *
## CD4_recent        0.1644     0.1252   1.313          0.1892
## Pain_defChronic   0.6567     0.2903   2.262          0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##              (Intr) Age    SexMal CD4_rc
## Age          0.228
## SexMale     -0.325 -0.184
## CD4_recent   0.029 -0.088  0.209
## Pan_dfChrnc -0.581 -0.323 -0.148 -0.332
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Knees --#
knees <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Knees') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = knees,
            family = binomial())

## Full model
knee_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = knees,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(knee_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
## Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: knees
##
##     AIC      BIC   logLik deviance df.resid
##    400.2    425.4   -194.1    388.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8381 -0.4380 -0.3233 -0.2914  3.8153
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0        0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)   -2.1860    0.1932 -11.314 < 0.0000000000000002 ***
## Age            0.2308    0.1323   1.745              0.08100 .
```

```
## SexMale             0.4132     0.3147    1.313              0.18928
## CD4_recent          0.0948     0.1320    0.718              0.47273
## Pain_defChronic     0.8155     0.3015    2.705              0.00683 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age     SexMal CD4_rc
## Age          0.160
## SexMale     -0.295 -0.163
## CD4_recent   0.059 -0.096  0.197
## Pan_dfChrnc -0.604 -0.309 -0.150 -0.336
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
#-- Ankles.Feet --#
ankles <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Ankles.Feet') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = ankles,
            family = binomial())

## Full model
ankl_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
            data = ankles,
            family = binomial())

## Fit is singular, print model to check SD of random effect
## Print model
summary(ankl_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: ankles
##
##      AIC      BIC   logLik deviance df.resid
##    570.2    595.4   -279.1    558.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4370 -0.5803 -0.4904  0.9236  2.3929
##
## Random effects:
##  Groups Name        Variance Std.Dev.
```

94

```
## Site   (Intercept) 0       0
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                Estimate Std. Error z value         Pr(>|z|)
## (Intercept)    -1.26753    0.14377  -8.816 < 0.0000000000000002 ***
## Age             0.27385    0.11072   2.473           0.0134 *
## SexMale         0.38509    0.26289   1.465           0.1430
## CD4_recent     -0.08338    0.11100  -0.751           0.4526
## Pain_defChronic 0.96276    0.24465   3.935           0.0000831 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal CD4_rc
## Age         0.216
## SexMale    -0.294 -0.177
## CD4_recent  0.142 -0.093  0.177
## Pan_dfChrnc -0.584 -0.295 -0.135 -0.356
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#-- Cervical spine --#
cervical <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Cervical_spine') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
             data = cervical,
             family = binomial())

## Full model
neck_mod <- glmer(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def +
                (1|Site),
             data = cervical,
             family = binomial())

## Compare models
anova(null_mod, neck_mod, test = 'LRT')
```

```
## Data: cervical
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## neck_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod    2 205.43 213.83 -100.716   201.43
## neck_mod    6 206.19 231.40  -97.096   194.19 7.2402  4     0.1237
```

```
## No significant difference between null and full models
## Print model
summary(neck_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: cervical
##
##      AIC      BIC   logLik deviance df.resid
##    206.2    231.4    -97.1    194.2      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.3705 -0.3024 -0.2546 -0.0844 11.3198
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.08489  0.2914
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value        Pr(>|z|)
## (Intercept)     -2.37663    0.31368  -7.577 0.0000000000000354 ***
## Age              0.14746    0.26428   0.558           0.577
## SexMale         -0.60510    0.80413  -0.752           0.452
## CD4_recent       0.06598    0.24090   0.274           0.784
## Pain_defChronic -2.65214    1.11087  -2.387           0.017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal CD4_rc
## Age         0.267
## SexMale    -0.243 -0.106
## CD4_recent  0.102 -0.145 -0.016
## Pan_dfChrnc -0.310 -0.224 -0.046 -0.143
```

```
#-- Thoracic spine --#
thoracic <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Thoracic_spine') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
            data = thoracic,
            family = binomial())

## Full model
thrx_mod <- glmer(Pain_present ~
                Age +
                Sex +
```

```
                CD4_recent +
                Pain_def +
                (1|Site),
            data = thoracic,
            family = binomial())

## Compare models
anova(null_mod, thrx_mod, test = 'LRT')
```

```
## Data: thoracic
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## thrx_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## null_mod    2 392.73 401.13 -194.36   388.73
## thrx_mod    6 396.75 421.95 -192.38   384.75 3.9765  4     0.4092
```

```
## No significant difference between null and full models
## Print model
summary(thrx_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##     Data: thoracic
##
##      AIC      BIC   logLik deviance df.resid
##    396.7    422.0   -192.4    384.7      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.5846 -0.4324 -0.3798 -0.2759  3.8233
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.05601  0.2367
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##                 Estimate Std. Error z value        Pr(>|z|)
## (Intercept)     -1.67590    0.24111  -6.951 0.00000000000363 ***
## Age             -0.01341    0.15894  -0.084           0.933
## SexMale         -0.66287    0.47408  -1.398           0.162
## CD4_recent       0.14524    0.14100   1.030           0.303
## Pain_defChronic -0.34346    0.42404  -0.810           0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    SexMal CD4_rc
## Age          0.281
## SexMale     -0.172 -0.101
## CD4_recent   0.055 -0.079  0.067
## Pan_dfChrnc -0.602 -0.335 -0.108 -0.264
```

```r
# Lumbosacral spine
lumbar <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Lumbosacral_spine') %>%
    filter(complete.cases(.))

## Null
null_mod <- glmer(Pain_present ~ 1 + (1|Site),
             data = lumbar,
             family = binomial())

## Full model
lmbr_mod <- glmer(Pain_present ~
               Age +
               Sex +
               CD4_recent +
               Pain_def +
               (1|Site),
             data = lumbar,
             family = binomial())

## Compare models
anova(null_mod, lmbr_mod, test = 'LRT')
```

```
## Data: lumbar
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## lmbr_mod: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##          npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## null_mod    2 451.68 460.08 -223.84   447.68
## lmbr_mod    6 456.11 481.32 -222.06   444.11 3.567  4     0.4678
```

```r
## No significant difference between null and full models
## Print model
summary(lmbr_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + Pain_def + (1 | Site)
##    Data: lumbar
##
##      AIC      BIC   logLik deviance df.resid
##    456.1    481.3   -222.1    444.1      487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7060 -0.4701 -0.4167 -0.3660  2.8857
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.04987  0.2233
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
```

```
##                  Estimate Std. Error z value          Pr(>|z|)
## (Intercept)      -1.59777    0.22763  -7.019 0.00000000000223 ***
## Age               0.02967    0.14127   0.210           0.8336
## SexMale           0.03293    0.34633   0.095           0.9242
## CD4_recent        0.22649    0.12388   1.828           0.0675 .
## Pain_defChronic  -0.05838    0.37842  -0.154           0.8774
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Age     SexMal CD4_rc
## Age           0.250
## SexMale      -0.199 -0.130
## CD4_recent    0.009 -0.108  0.134
## Pan_dfChrnc  -0.609 -0.335 -0.147 -0.232
```

Body sites with models that generated an *"is singular"* error included:

- Throat

- Arms

- Elbows

- Wrists & hands

- Abdomen

- Lower back (flanks)

- Buttocks

- Legs

- Knees

- Ankles & feet

All these models had a random error variance of 0 (or tending to 0), indicating that **site** was not contributing to the model and could be removed. That is, there is not enough additional site-level variation to warrant adding a random effect to explain all the observed variation. Analyzed these using `glm`, not `glmm`, and check for issues.

8.4 Run `glm` for selected body sites

Note: p-values reported for the likelihood ratio tests below are uncorrected for multiple comparisons.

```
#-- Throat --#
throat <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Throat') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = throat,
                family = binomial())

## Full model
thrt_mod <- glm(Pain_present ~
                Age +
```

```
                Sex +
                CD4_recent  +
                Pain_def,
            data = throat,
            family = binomial())

## Compare models
anova(null_mod, thrt_mod, test = 'LRT')

## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492    120.18
## 2       488    112.98  4   7.1991   0.1257

## No significant difference between the null and full models
## Print model
summary(thrt_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = throat)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -0.5014  -0.2524  -0.2063  -0.1542   2.9367
##
## Coefficients:
##                 Estimate Std. Error z value          Pr(>|z|)
## (Intercept)     -4.12952    0.46370  -8.906 <0.0000000000000002 ***
## Age              0.07201    0.33111   0.217            0.8278
## SexMale          0.91911    0.62751   1.465            0.1430
## CD4_recent      -0.78730    0.45238  -1.740            0.0818 .
## Pain_defChronic  0.19658    0.68880   0.285            0.7753
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 120.18  on 492  degrees of freedom
## Residual deviance: 112.98  on 488  degrees of freedom
## AIC: 122.98
##
## Number of Fisher Scoring iterations: 7

#-- Arms --#
arms <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Arms') %>%
    filter(complete.cases(.))

## Null
```

```
null_mod <- glm(Pain_present ~ 1,
                data = arms,
                family = binomial())

## Full model
arms_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent  +
                Pain_def,
            data = arms,
            family = binomial())

## Compare models
anova(null_mod, arms_mod, test = 'LRT')

## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     179.82
## 2       488     177.13  4    2.686   0.6117

## No significant difference between the null and full models
## Print model
summary(arms_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = arms)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -0.5876  -0.3257  -0.2959  -0.2574   2.6283
##
## Coefficients:
##                 Estimate Std. Error z value        Pr(>|z|)
## (Intercept)     -2.89348    0.27466 -10.535 <0.0000000000000002 ***
## Age              0.30866    0.21783   1.417            0.156
## SexMale         -0.05672    0.59049  -0.096            0.923
## CD4_recent      -0.10059    0.25597  -0.393            0.694
## Pain_defChronic -0.62336    0.59010  -1.056            0.291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 179.82  on 492  degrees of freedom
## Residual deviance: 177.13  on 488  degrees of freedom
## AIC: 187.13
##
## Number of Fisher Scoring iterations: 6
```

```
#-- Elbow --#
elbows <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Elbows') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = elbows,
                family = binomial())

## Full model
elbw_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent  +
                Pain_def,
            data = elbows,
            family = binomial())

## Compare models
anova(null_mod, elbw_mod, test = 'LRT')

## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     167.37
## 2       488     160.13  4   7.2361   0.1239
## No significant difference between the null and full models
## Print model
summary(elbw_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = elbows)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6164  -0.2974  -0.2476  -0.2156   2.7664
##
## Coefficients:
##                 Estimate Std. Error z value         Pr(>|z|)
## (Intercept)     -3.38921    0.33556 -10.100 <0.0000000000000002 ***
## Age              0.42521    0.21018   2.023           0.0431 *
## SexMale         -0.16645    0.59159  -0.281           0.7784
## CD4_recent       0.04229    0.23571   0.179           0.8576
## Pain_defChronic  0.45074    0.53643   0.840           0.4008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##      Null deviance: 167.37  on 492  degrees of freedom
## Residual deviance: 160.13  on 488  degrees of freedom
## AIC: 170.13
##
## Number of Fisher Scoring iterations: 6
#-- Wrist & hand --#
wrists <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Wrists.Hands') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = wrists,
                family = binomial())

## Full model
wrst_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent  +
                Pain_def,
              data = wrists,
              family = binomial())

## Compare models
anova(null_mod, wrst_mod, test = 'LRT')

## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492      242.2
## 2       488      231.3  4   10.908  0.02762 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## No significant difference between the null and full models
## (family-wise error correction, alpha = 0.0024)
## Print model
summary(wrst_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = wrists)
##
## Deviance Residuals:
##     Min        1Q   Median        3Q       Max
## -0.8224   -0.3799  -0.3189  -0.2853    2.7809
##
## Coefficients:
##                Estimate Std. Error z value            Pr(>|z|)
```

```
## (Intercept)      -2.7754      0.2588 -10.723 <0.0000000000000002 ***
## Age               0.2316      0.1804   1.284                 0.199
## SexMale          -0.6507      0.5662  -1.149                 0.251
## CD4_recent        0.2766      0.1711   1.616                 0.106
## Pain_defChronic   0.4408      0.4376   1.007                 0.314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 242.2  on 492  degrees of freedom
## Residual deviance: 231.3  on 488  degrees of freedom
## AIC: 241.3
##
## Number of Fisher Scoring iterations: 5
```

```r
#-- Abdomen --#
abdomen <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Abdomen') %>%
    filter(complete.cases(.))


## Null
null_mod <- glm(Pain_present ~ 1,
               data = abdomen,
               family = binomial())

## Full model
abdm_mod <- glm(Pain_present ~
               Age +
               Sex +
               CD4_recent  +
               Pain_def,
            data = abdomen,
            family = binomial())

## Compare models
anova(null_mod, abdm_mod, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     578.82
## 2       488     569.15  4   9.6645  0.04648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## No significant difference between the null and full models
## (family-wise error correction, alpha = 0.0024)
## Print model
summary(abdm_mod)
```

```
##
## Call:
```

```
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = abdomen)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9950  -0.8601  -0.7247   1.4630   1.9325
##
## Coefficients:
##                 Estimate Std. Error z value     Pr(>|z|)
## (Intercept)      -0.8813     0.1365  -6.458 0.000000000106 ***
## Age              -0.1613     0.1219  -1.323          0.186
## SexMale          -0.3072     0.3015  -1.019          0.308
## CD4_recent       -0.1145     0.1154  -0.993          0.321
## Pain_defChronic  -0.2542     0.2706  -0.940          0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 578.82  on 492  degrees of freedom
## Residual deviance: 569.15  on 488  degrees of freedom
## AIC: 579.15
##
## Number of Fisher Scoring iterations: 4
```

```r
#-- Lower back (flank) --#
low_back <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Lower_back') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = low_back,
                family = binomial())

## Full model
back_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent  +
                Pain_def,
            data = low_back,
            family = binomial())

## Compare models
anova(null_mod, back_mod, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     262.78
## 2       488     260.15  4   2.6328    0.621
```

```r
## No significant difference between the null and full models
## Print model
summary(back_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = low_back)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5466  -0.4105  -0.3690  -0.3520   2.4494
##
## Coefficients:
##                 Estimate Std. Error z value            Pr(>|z|)
## (Intercept)     -2.59625    0.23601 -11.001 <0.0000000000000002 ***
## Age              0.10656    0.18068   0.590               0.555
## SexMale         -0.23426    0.48221  -0.486               0.627
## CD4_recent       0.08667    0.17452   0.497               0.619
## Pain_defChronic  0.33885    0.41624   0.814               0.416
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 262.78  on 492  degrees of freedom
## Residual deviance: 260.15  on 488  degrees of freedom
## AIC: 270.15
##
## Number of Fisher Scoring iterations: 5
```

```r
#-- Buttocks --#
buttocks <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Buttocks') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = buttocks,
                family = binomial())

## Full model
butt_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent  +
                Pain_def,
            data = buttocks,
            family = binomial())

## Compare models
anova(null_mod, butt_mod, test = 'LRT')
```

```
## Analysis of Deviance Table
```

```
## 
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492      160.99
## 2       488      158.73  4   2.2631   0.6875
```

```
## No significant difference between the null and full models
## Print model
summary(butt_mod)
```

```
## 
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = buttocks)
## 
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4682  -0.3007  -0.2608  -0.2401   2.7090
## 
## Coefficients:
##                 Estimate Std. Error z value            Pr(>|z|)
## (Intercept)     -3.42677    0.33725 -10.161 <0.0000000000000002 ***
## Age             -0.12939    0.26949  -0.480              0.631
## SexMale          0.65435    0.57735   1.133              0.257
## CD4_recent       0.23689    0.22789   1.039              0.299
## Pain_defChronic  0.09589    0.57052   0.168              0.867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 160.99  on 492  degrees of freedom
## Residual deviance: 158.73  on 488  degrees of freedom
## AIC: 168.73
## 
## Number of Fisher Scoring iterations: 6
```

```
#-- Legs --#
legs <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Legs') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
            data = legs,
            family = binomial())

## Full model
legs_mod <- glm(Pain_present ~
            Age +
            Sex +
            CD4_recent +
            Pain_def,
```

```
              data = legs,
              family = binomial())

## Compare models
anova(null_mod, legs_mod, test = 'LRT')

## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       492     440.48
## 2       488     419.02  4   21.454 0.0002574 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Significant difference between the null and full models
## (family-wise error correction, alpha = 0.0024)
summary(legs_mod)

##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = legs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0034  -0.6258  -0.4805  -0.4483   2.1998
##
## Coefficients:
##                 Estimate Std. Error z value         Pr(>|z|)
## (Intercept)      -2.0238     0.1812 -11.170 <0.0000000000000002 ***
## Age               0.1117     0.1305   0.856          0.3920
## SexMale           0.6088     0.3013   2.020          0.0433 *
## CD4_recent        0.1644     0.1252   1.313          0.1892
## Pain_defChronic   0.6567     0.2903   2.262          0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 440.48  on 492  degrees of freedom
## Residual deviance: 419.02  on 488  degrees of freedom
## AIC: 429.02
##
## Number of Fisher Scoring iterations: 4
#-- Knees --#
knees <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Knees') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
```

```
                  data = knees,
                  family = binomial())

## Full model
knee_mod <- glm(Pain_present ~
                  Age +
                  Sex +
                  CD4_recent +
                  Pain_def,
                data = knees,
                family = binomial())

## Compare models
anova(null_mod, knee_mod, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       492     413.48
## 2       488     388.18  4   25.298 0.00004382 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Significant difference between the null and full models
## (family-wise error correction, alpha = 0.0024)
summary(knee_mod)
```

```
##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = knees)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0315  -0.5925  -0.4460  -0.4038   2.3429
##
## Coefficients:
##                 Estimate Std. Error z value          Pr(>|z|)
## (Intercept)      -2.1860     0.1932 -11.314 < 0.0000000000000002 ***
## Age               0.2308     0.1323   1.745           0.08100 .
## SexMale           0.4132     0.3147   1.313           0.18928
## CD4_recent        0.0948     0.1320   0.718           0.47273
## Pain_defChronic   0.8155     0.3015   2.705           0.00683 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 413.48  on 492  degrees of freedom
## Residual deviance: 388.18  on 488  degrees of freedom
## AIC: 398.18
##
## Number of Fisher Scoring iterations: 5
```

```r
#-- Ankles/feet --#
ankles <- data_logit.long %>%
    # Filter by pain site
    filter(Pain_site == 'Ankles.Feet') %>%
    filter(complete.cases(.))

## Null
null_mod <- glm(Pain_present ~ 1,
                data = ankles,
                family = binomial())

## Full model
ankl_mod <- glm(Pain_present ~
                Age +
                Sex +
                CD4_recent +
                Pain_def,
            data = ankles,
            family = binomial())

## Compare models
anova(null_mod, ankl_mod, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent + Pain_def
##   Resid. Df Resid. Dev Df Deviance      Pr(>Chi)
## 1       492     600.78
## 2       488     558.18  4   42.596 0.00000001255 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Significant difference between the null and full models
## (family-wise error correction, alpha = 0.0024)
summary(ankl_mod)
```

```
##
## Call:
## glm(formula = Pain_present ~ Age + Sex + CD4_recent + Pain_def,
##     family = binomial(), data = ankles)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4967  -0.7619  -0.6565   1.1107   1.9524
##
## Coefficients:
##                 Estimate Std. Error z value            Pr(>|z|)
## (Intercept)     -1.26753    0.14377  -8.816 < 0.0000000000000002 ***
## Age              0.27385    0.11072   2.473              0.0134 *
## SexMale          0.38509    0.26289   1.465              0.1430
## CD4_recent      -0.08338    0.11100  -0.751              0.4526
## Pain_defChronic  0.96276    0.24465   3.935           0.0000831 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 600.78  on 492  degrees of freedom
## Residual deviance: 558.18  on 488  degrees of freedom
## AIC: 568.18
##
## Number of Fisher Scoring iterations: 4
```

8.5   GLMM data for tabulation and plotting

Perform GLMM on selected body sites (head, shoulders, chest, hips, cervical spine, thoracic spine, and lumbosacral spine) to extract fixed effect odds ratios (with 95% CIs) to tabulate and to produce forest plots.

```r
# Perform GLMM on each selected body sites
glmm_mods <- data_logit.long %>%
    filter(complete.cases(.)) %>%
    # remove body sites with glmm singular fits
    filter(!Pain_site %in% c("Throat", "Arms", "Elbows", "Lower_back",
                             "Wrists.Hands", "Abdomen", "Buttocks",
                             "Legs", "Knees", "Ankles.Feet")) %>%
    # Group and then nest by Pain_site
    group_by(Pain_site) %>%
    nest() %>%
    # Perform logistic regression
    mutate(logistic = map(.x = data,
                          ~ glmer(Pain_present ~
                                      Age +
                                      Sex +
                                      CD4_recent +
                                      Pain_def +
                                      (1|Site),
                                  data = .x,
                                  family = binomial())),
           null = map(.x = data,
                      ~glmer(Pain_present ~ 1 +
                                 (1|Site),
                             data = .x,
                             family = binomial()))) %>%
    # Perform likelihood ratio test
    mutate(LRT = map2(.x = logistic,
                      .y = null,
                      ~ anova(.y, .x, test = 'LRT'))) %>%
    # Extract LRT p-values and correct for multiple comparisons
    mutate(LRT_p = map(.x = LRT,
                       ~ .x[, 8][[2]]),
           LRT_p.corrected = map(.x = LRT_p,
                                 ~ p.adjust(p = .x,
                                            method = 'bonferroni',
                                            n = 21))) %>%
    # Extract effect sizes
    mutate(OR = map(.x = logistic,
                    ~ exp(fixef(.x))),
           CI = map(.x = logistic,
```

```r
                          ~ exp(confint.merMod(.x,
                                        method = 'Wald')[2:6, ]))) %>%
ungroup() %>%
mutate(Pain_site = str_replace(Pain_site,
                          pattern = '_',
                          replacement = ' ')) %>%
mutate(Pain_site = str_replace(Pain_site,
                          pattern = '\\.',
                          replacement = ' & ')) %>%
# Bind data into a dataframe and tabulate,
mutate(df = pmap(.l = list(OR, CI, LRT_p.corrected, Pain_site),
                ~ cbind(..1, ..2) %>%
                    .[2:5, ] %>%
                    kable(caption = str_glue('{..4} (GLMM; LRT corrected p-value = {round(..3, 3)}]
                            digits = 3,
                            col.names = c('OR', 'Wald lower 95%CI',
                                        'Wald upper 95%CI')))) %>%
# Generate forest plot data
mutate(forest_data = map2(.x = OR,
                          .y = CI,
                          # Create the dataframe
                          ~ as.data.frame(cbind(.x, .y)) %>%
                              set_names(nm = c('OR', 'lower_ci',
                                            'upper_ci')) %>%
                              .[-1, ] %>%
                              rownames_to_column() %>%
                              mutate(rowname = case_when(
                                  rowname == 'Age' ~ 'Age',
                                  rowname == 'SexMale' ~ 'Sex (male)',
                                  rowname == 'CD4_recent' ~ 'CD4 T-cells',
                                  rowname == 'Pain_defChronic' ~ 'Pain (chronic)'
                                  )) %>%
                              # Round digits to 2 decimal places
                              mutate_if(is.numeric, round, 2) %>%
                              # Paste CI range
                              mutate(ci_range = paste0('(', lower_ci, ' to ',
                                                upper_ci, ')')) %>%
                              # Cut short confidence interval for plotting
                              mutate(upper_ci = ifelse(upper_ci > 4,
                                                    yes = 4,
                                                    no = upper_ci)))) %>%
# Generate forest plots
mutate(forest_plots = pmap(.l = list(forest_data, Pain_site, LRT_p.corrected),
                        ~ ..1 %>%
                            ggplot(data = .) +
                            aes(x = OR,
                                xmin = lower_ci,
                                xmax = upper_ci,
                                y = rowname,
                                label = ci_range) +
                            geom_pointrange(size = 1) +
                            geom_vline(xintercept = 1,
                                    linetype = 2) +
```

```
                        scale_x_continuous(limits = c(-0.5, 8),
                                            breaks = c(0, 1, 2, 3, 4)) +
                        geom_text(x = 4.5, hjust = 0, size = 5) +
                        annotate(geom = 'text',
                                 label = '95% CI',
                                 size = 5,
                                 fontface = 2,
                                 x = 4.5,
                                 y = 4.4,
                                 hjust = 0) +
                        labs(title = ..2,
                             subtitle = str_glue('(GLMM | LRT corrected p-value =  {round(..3
                             x = 'Odds ratio') +
                        theme_minimal(base_size = 18) +
                        theme(plot.title = element_text(size = 18),
                              plot.subtitle = element_text(size = 14),
                              panel.grid = element_blank(),
                              axis.title.y = element_blank(),
                              axis.text = element_text(colour = '#000000'),
                              axis.line = element_line(size = 0.5),
                              axis.ticks = element_line(size = 0.5))))
```

8.6 GLM data for tabulation and plotting

Perform GLM on selected body sites (throat, arms, elbows, wrists and hands, abdomen, buttocks, legs, knees, and ankles and feet) to extract fixed effect odds ratios (with 95% CIs) to tabulate and to produce forest plots.

```
# Perform GLM on each selected body sites
glm_mods <- data_logit.long %>%
    filter(complete.cases(.)) %>%
    # Retain body sites with glmm singular fits
    filter(Pain_site %in% c("Throat", "Arms", "Elbows", "Lower_back",
                            "Wrists.Hands", "Abdomen", "Buttocks",
                            "Legs", "Knees", "Ankles.Feet")) %>%
    # Change lower_back to flank
    mutate(Pain_site = ifelse(Pain_site == 'Lower_back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    # Group and then nest by Pain_site
    group_by(Pain_site) %>%
    nest() %>%
    # Perform logistic regression
    mutate(logistic = map(.x = data,
                          ~ glm(Pain_present ~
                                Age +
                                Sex +
                                CD4_recent +
                                Pain_def,
                                data = .x,
                                family = binomial())),
           null = map(.x = data,
                      ~glm(Pain_present ~ 1,
                           data = .x,
```

113

```r
                                   family = binomial())))) %>%
    # Perform likelihood ratio test
    mutate(LRT = map2(.x = logistic,
                      .y = null,
                      ~ anova(.y, .x, test = 'LRT'))) %>%
    # Extract LRT p-values and correct for multiple comparisons
    mutate(LRT_p = map(.x = LRT,
                       ~ .x[, 5][[2]]),
           LRT_p.corrected = map(.x = LRT_p,
                                 ~ p.adjust(p = .x,
                                            method = 'bonferroni',
                                            n = 21))) %>%
    # Extract effect sizes
    mutate(OR = map(.x = logistic,
                    ~ exp(coef(.x))),
           CI = map(.x = logistic,
                    ~ exp(confint.default(.x)))) %>%
    ungroup() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    # Bind data into a dataframe and tabulate,
    mutate(df = pmap(.l = list(OR, CI, LRT_p.corrected, Pain_site),
                     ~ cbind(..1, ..2) %>%
                       .[2:5, ] %>%
                       kable(caption = str_glue('{..4} (GLM; LRT corrected p-value = {round(..3, 3)})'),
                             digits = 3,
                             col.names = c('OR', 'Wald lower 95%CI',
                                           'Wald upper 95%CI')))) %>%
    # Generate forest plot data
    mutate(forest_data = map2(.x = OR,
                              .y = CI,
                              # Create the dataframe
                              ~ as.data.frame(cbind(.x, .y)) %>%
                                set_names(nm = c('OR', 'lower_ci',
                                                 'upper_ci')) %>%
                                .[-1, ] %>%
                                rownames_to_column() %>%
                                mutate(rowname = case_when(
                                    rowname == 'Age' ~ 'Age',
                                    rowname == 'SexMale' ~ 'Sex (male)',
                                    rowname == 'CD4_recent' ~ 'CD4 T-cells',
                                    rowname == 'Pain_defChronic' ~ 'Pain (chronic)'
                                    )) %>%
                                # Round digits to 2 decimal places
                                mutate_if(is.numeric, round, 2) %>%
                                # Paste CI range
                                mutate(ci_range = paste0('(', lower_ci, ' to ',
                                                         upper_ci, ')')) %>%
                                # Cut short confidence interval for plotting
```

```
                                mutate(upper_ci = ifelse(upper_ci > 4,
                                                         yes = 4,
                                                         no = upper_ci)))) %>%
        # Generate forest plots
        mutate(forest_plots = pmap(.l = list(forest_data, Pain_site, LRT_p.corrected),
                                ~ ..1 %>%
                                    ggplot(data = .) +
                                    aes(x = OR,
                                        xmin = lower_ci,
                                        xmax = upper_ci,
                                        y = rowname,
                                        label = ci_range) +
                                    geom_pointrange(size = 1) +
                                    geom_vline(xintercept = 1,
                                               linetype = 2) +
                                    scale_x_continuous(limits = c(-0.5, 8),
                                                       breaks = c(0, 1, 2, 3, 4)) +
                                    geom_text(x = 4.5, hjust = 0, size = 5) +
                                    annotate(geom = 'text',
                                             label = '95% CI',
                                             size = 5,
                                             fontface = 2,
                                             x = 4.5,
                                             y = 4.4,
                                             hjust = 0) +
                                    labs(title = ..2,
                                         subtitle = str_glue('(GLM | LRT corrected p-value = {round(..3,
                                         x = 'Odds ratio') +
                                    theme_minimal(base_size = 18) +
                                    theme(plot.title = element_text(size = 18),
                                          plot.subtitle = element_text(size = 14),
                                          panel.grid = element_blank(),
                                          axis.title.y = element_blank(),
                                          axis.text = element_text(colour = '#000000'),
                                          axis.line = element_line(size = 0.5),
                                          axis.ticks = element_line(size = 0.5))))
```

8.7   Tabulated results (fixed effects)

8.7.1   GLMM tabulations

```
# Print OR for fixed effects with 95% CIs and p-values
walk(glmm_mods$df, ~print(.x))
```

Table 61: Head (GLMM; LRT corrected p-value = 0.036)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.036	0.815	1.317
SexMale	0.260	0.122	0.553
CD4_recent	0.895	0.714	1.121
Pain_defChronic	0.577	0.249	1.340

Table 62: Shoulder (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.861	0.578	1.282
SexMale	1.703	0.733	3.957
CD4_recent	0.941	0.651	1.360
Pain_defChronic	1.111	0.451	2.739

Table 63: Chest (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.099	0.838	1.442
SexMale	1.044	0.551	1.979
CD4_recent	0.757	0.564	1.015
Pain_defChronic	0.470	0.219	1.008

Table 64: Cervical spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.159	0.690	1.945
SexMale	0.546	0.113	2.640
CD4_recent	1.068	0.666	1.713
Pain_defChronic	0.071	0.008	0.622

Table 65: Thoracic spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.987	0.723	1.347
SexMale	0.515	0.204	1.305
CD4_recent	1.156	0.877	1.524
Pain_defChronic	0.709	0.309	1.628

Table 66: Lumbosacral spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.030	0.781	1.359
SexMale	1.033	0.524	2.038
CD4_recent	1.254	0.984	1.599
Pain_defChronic	0.943	0.449	1.980

Table 67: Groin (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.947	0.680	1.320
SexMale	0.532	0.224	1.264

	OR	Wald lower 95%CI	Wald upper 95%CI
CD4_recent	1.229	0.885	1.706
Pain_defChronic	0.661	0.084	5.217

Table 68: Hips (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.057	0.728	1.536
SexMale	1.373	0.581	3.245
CD4_recent	1.131	0.802	1.593
Pain_defChronic	6.570	2.204	19.585

8.7.2 GLM tabulations

```
# Print OR for fixed effects with 95% CIs and p-values
walk(glm_mods$df, ~print(.x))
```

Table 69: Throat (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.075	0.562	2.056
SexMale	2.507	0.733	8.576
CD4_recent	0.455	0.188	1.104
Pain_defChronic	1.217	0.316	4.696

Table 70: Arms (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.362	0.888	2.087
SexMale	0.945	0.297	3.006
CD4_recent	0.904	0.548	1.493
Pain_defChronic	0.536	0.169	1.704

Table 71: Elbows (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.530	1.013	2.310
SexMale	0.847	0.266	2.699
CD4_recent	1.043	0.657	1.656
Pain_defChronic	1.569	0.548	4.491

Table 72: Wrists & Hands (GLM; LRT corrected p-value = 0.58)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.261	0.885	1.795

	OR	Wald lower 95%CI	Wald upper 95%CI
SexMale	0.522	0.172	1.583
CD4_recent	1.319	0.943	1.844
Pain_defChronic	1.554	0.659	3.663

Table 73: Lower back/flanks (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.112	0.781	1.585
SexMale	0.791	0.307	2.036
CD4_recent	1.091	0.775	1.535
Pain_defChronic	1.403	0.621	3.173

Table 74: Abdomen (GLM; LRT corrected p-value = 0.976)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.851	0.670	1.081
SexMale	0.736	0.407	1.328
CD4_recent	0.892	0.711	1.118
Pain_defChronic	0.776	0.456	1.318

Table 75: Legs (GLM; LRT corrected p-value = 0.005)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.118	0.866	1.444
SexMale	1.838	1.018	3.318
CD4_recent	1.179	0.922	1.506
Pain_defChronic	1.928	1.092	3.407

Table 76: Knees (GLM; LRT corrected p-value = 0.001)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.260	0.972	1.632
SexMale	1.512	0.816	2.801
CD4_recent	1.099	0.849	1.424
Pain_defChronic	2.260	1.252	4.081

Table 77: Ankles & Feet (GLM; LRT corrected p-value = 0)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.315	1.058	1.634
SexMale	1.470	0.878	2.460
CD4_recent	0.920	0.740	1.144
Pain_defChronic	2.619	1.621	4.230

Table 78: Buttocks (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.879	0.518	1.490
SexMale	1.924	0.620	5.965
CD4_recent	1.267	0.811	1.981
Pain_defChronic	1.101	0.360	3.367

8.8 Plotted results (fixed effects)

Includes both GLM and GLMM results.

```r
# Extract and process sub-plots from GLMM data
head <- glmm_mods$forest_plots[[1]] +
    theme(axis.title.x = element_blank())

shoulder <- glmm_mods$forest_plots[[2]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

chest <- glmm_mods$forest_plots[[3]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

cervical <- glmm_mods$forest_plots[[4]] +
    theme(axis.text.y = element_blank())

thoracic<- glmm_mods$forest_plots[[5]]

lumbar <- glmm_mods$forest_plots[[6]] +
    theme(axis.text.y = element_blank())

groin <- glmm_mods$forest_plots[[7]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

hips <- glmm_mods$forest_plots[[8]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

# Extract and process sub-plots from GLM data
throat <- glm_mods$forest_plots[[1]] +
    theme(axis.title.x = element_blank(),
          axis.text.y = element_blank())

arms <- glm_mods$forest_plots[[2]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

elbows <- glm_mods$forest_plots[[3]] +
    theme(axis.title.x = element_blank())

wrists.hands <- glm_mods$forest_plots[[4]] +
```

```r
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank()))

flank <- glm_mods$forest_plots[[5]] +
    theme(axis.title.x = element_blank())

abdomen <- glm_mods$forest_plots[[6]] +
    theme(axis.title.x = element_blank(),
          axis.text.y = element_blank())

legs <- glm_mods$forest_plots[[7]] +
    theme(axis.title.x = element_blank())

knees <- glm_mods$forest_plots[[8]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

ankles.feet <- glm_mods$forest_plots[[9]] +
    theme(axis.text.y = element_blank())

buttocks <- glm_mods$forest_plots[[10]] +
    theme(axis.text.y = element_blank(),
          axis.title.x = element_blank())

# Patchwork
log_patch <- head + throat + chest + abdomen + flank + groin +
    shoulder + arms + elbows + wrists.hands +
    buttocks + hips + legs + knees + ankles.feet +
    cervical + thoracic + lumbar +
    plot_layout(ncol = 4)

# Save
ggsave(filename = 'figures/figure_2.png',
       plot = log_patch,
       width = 16,
       height = 18)
```
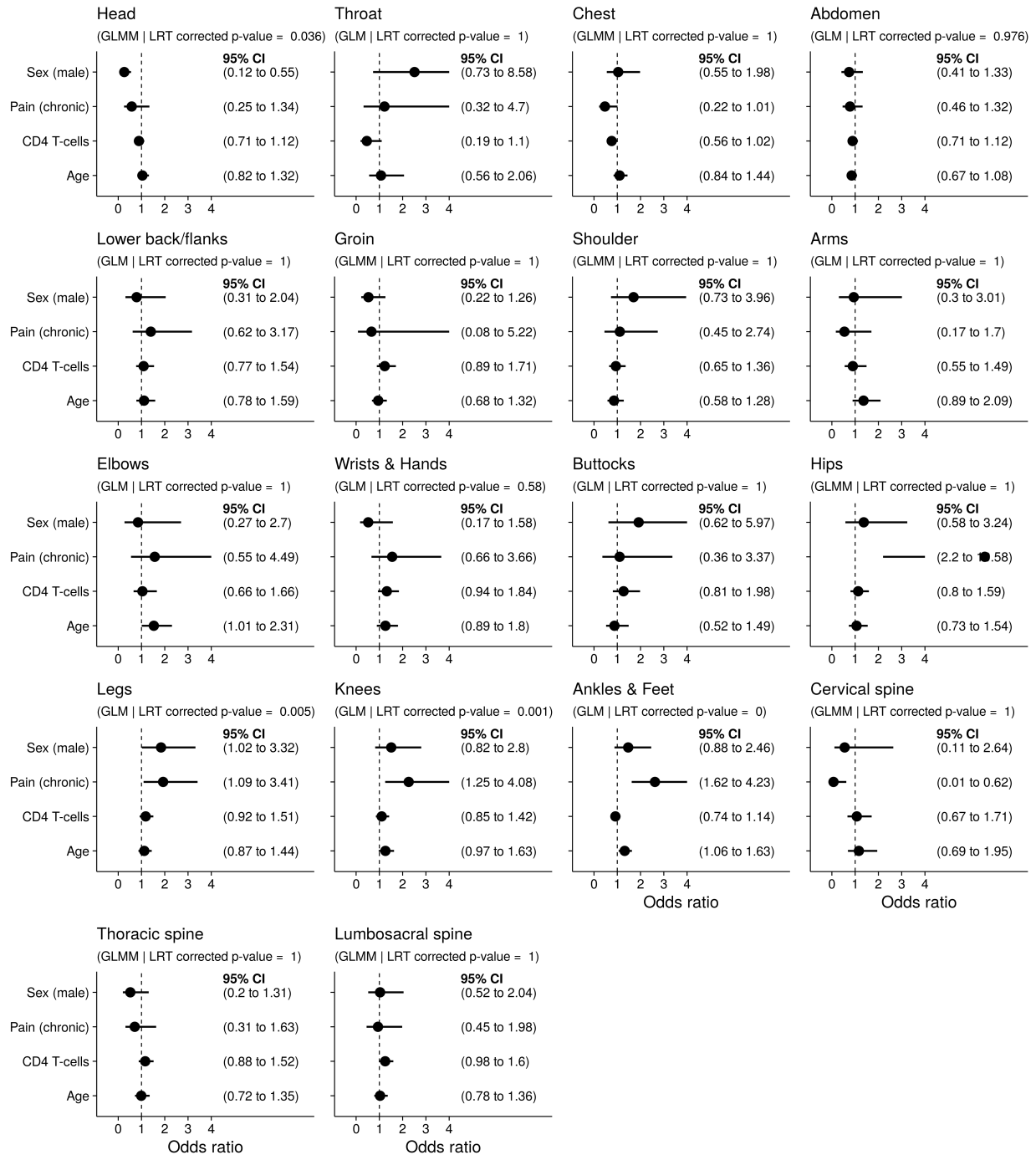
Figure 1: Plotted results (fixed effects)

9 Summary plots

9.1 Body sites with a point estimate pain proportion >10%

```
# Set seed
set.seed(2020)
```

```r
# Generate a filter to extract sites with >10% pain prevalence
filter <- data_logit %>%
    select(-Site, -CD4_recent, -Age, -Sex, -Pain_def) %>%
    pivot_longer(cols = everything(),
                 names_to = 'Site',
                 values_to = 'Response') %>%
    group_by(Site, Response) %>%
    summarise(count = n()) %>%
    mutate(total = sum(count),
           prop = count/total) %>%
    filter(Response == 'Yes')

filter_geq10 <- filter(filter, prop >= 0.1) %>%
    .$Site

# Generate new dataset
data_geq10 <- data_logit %>%
    pivot_longer(cols = -c(Site, CD4_recent, Sex, Age, Pain_def),
                 names_to = 'Pain_site',
                 values_to = 'Pain_present') %>%
    # Filter by filter_geq10
    filter(Pain_site %in% filter_geq10)

pubs_nofacet <- data_geq10 %>%
    group_by(Pain_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet <- pubs_nofacet %>%
    select(-CD4_recent, -Sex, -Age, -Pain_present, -Site, -Pain_def) %>%
    distinct() %>%
```

```r
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Head', 'Chest', 'Abdomen',
                                         'Legs', 'Knees', 'Ankles & Feet',
                                         'Thoracic spine', 'Lumbosacral spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci) +
    geom_linerange(size = 1) +
    geom_point(size = 5) +
    coord_flip() +
    labs(title = 'Body sites',
         subtitle = '(Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.6)) +
    theme_minimal(base_size = 18) +
    theme(plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by sex --#
pubs_nofacet.sex <- data_geq10 %>%
    group_by(Pain_site, Sex) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
```

```
                            ~ .x$percent[[5]]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet.sex <- pubs_nofacet.sex %>%
    select(-CD4_recent, -Age, -Pain_present, -Site, -Pain_def) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Head', 'Chest', 'Abdomen',
                                         'Legs', 'Knees', 'Ankles & Feet',
                                         'Thoracic spine', 'Lumbosacral spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Sex) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by sex',
         subtitle = '(Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.6)) +
    scale_fill_manual(values = c('#FFFFFF', '#000000')) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.1),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))
```

```r
#-- Body sites by pain chronicity --#
pubs_nofacet.pain_def <- data_geq10 %>%
    group_by(Pain_site, Pain_def) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet.pain_def <- pubs_nofacet.pain_def %>%
    select(-CD4_recent, -Age, -Pain_present, -Site, -Sex) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Head', 'Chest', 'Abdomen',
                                         'Legs', 'Knees', 'Ankles & Feet',
                                         'Thoracic spine', 'Lumbosacral spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_def) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
```

```r
                    position = position_dodge2(width = 0.6),
                    size = 4,
                    stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by pain chronicity',
         subtitle = '(Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.6)) +
    scale_fill_manual(values = c('#FFFFFF', '#000000')) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.1),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by age --#
pubs_nofacet.age <- data_geq10 %>%
    select(-CD4_recent, -Site, -Sex, -Pain_def) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    group_by(Pain_site, Pain_present) %>%
    nest() %>%
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = median_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup() %>%
```

```
    mutate(Pain_site = factor(Pain_site,
                                levels = c('Head', 'Chest', 'Abdomen', 'Groin',
                                           'Legs', 'Knees', 'Ankles & Feet',
                                           'Thoracic spine', 'Lumbosacral spine'),
                                ordered = TRUE)) %>%
    select(-Age) %>%
    distinct()

plot_nofacet.age <- pubs_nofacet.age %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_present) +
    geom_linerange(position = position_dodge2(width = 0.6),
                    size = 1,
                    colour = '#000000') +
    geom_point(shape = 21,
                colour = '#000000',
                position = position_dodge2(width = 0.6),
                size = 4,
                stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by median age',
        subtitle = '(Point estimate with 95%CI)',
        y = 'Age (years)') +
    scale_fill_manual(values = c('#FFFFFF', '#000000'),
                       labels = c('No pain', 'Pain present')) +
    scale_y_continuous(limits = c(31, 45),
                        breaks = c(31, 35, 39, 43)) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.95),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by CD4 --#
pubs_nofacet.cd4 <- data_geq10 %>%
    select(-Age, -Site, -Sex, -Pain_def) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                    pattern = '_',
                                    replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                    pattern = '\\.',
                                    replacement = ' & ')) %>%
    group_by(Pain_site, Pain_present) %>%
```

```r
    nest() %>%
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = median_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup() %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Head', 'Chest', 'Abdomen',
                                         'Legs', 'Knees', 'Ankles & Feet',
                                         'Thoracic spine', 'Lumbosacral spine'),
                              ordered = TRUE)) %>%
    select(-CD4_recent) %>%
    distinct()

plot_nofacet.cd4 <- pubs_nofacet.cd4 %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_present) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by median CD4 T-cell count',
         subtitle = '(Point estimate with 95%CI)',
         y = expression('CD4 T-cell count (cell.mm'^-3*')')) +
    scale_fill_manual(values = c('#FFFFFF', '#000000'),
                      labels = c('No pain', 'Pain present')) +
    scale_y_continuous(limits = c(100, 500),
```

```
                          breaks = c(100, 200, 300, 400, 500)) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.95),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

# Patchwork
pub_plot <- plot_nofacet + plot_nofacet.sex + plot_nofacet.pain_def +
    plot_nofacet.age + plot_nofacet.cd4 +
    plot_layout(ncol = 2)

# Save
ggsave(filename = 'figures/figure_supp1.png',
       width = 14,
       height = 18)
```

9.2 Body sites with a point estimate pain proportion <10%

```
# Set seed
set.seed(2020)

# Generate filter
filter_less10 <- filter(filter, prop < 0.1) %>%
    .$Site

# Generate new dataset
data_less10 <- data_logit %>%
    pivot_longer(cols = -c(Site, CD4_recent, Sex, Age, Pain_def),
                 names_to = 'Pain_site',
                 values_to = 'Pain_present') %>%
    # Filter by filter_geq10
    filter(Pain_site %in% filter_less10)

pubs_nofacet2 <- data_less10 %>%
    group_by(Pain_site) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
```
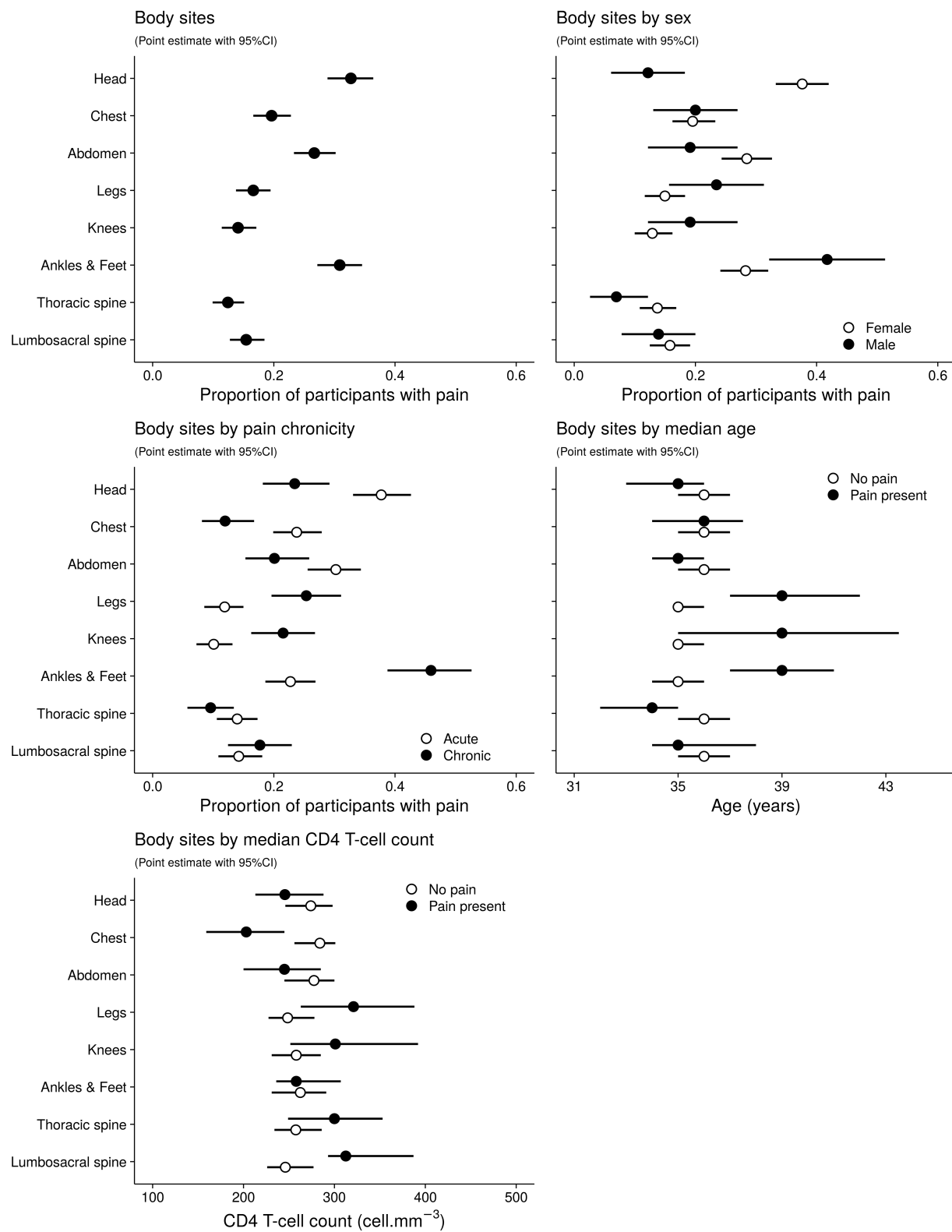
Figure 2: Body sites with a point estimate pain proportion >10%

```r
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet2 <- pubs_nofacet2 %>%
    select(-CD4_recent, -Sex, -Age, -Pain_present, -Site, -Pain_def) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = ifelse(Pain_site == 'Lower back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Throat', 'Shoulder', 'Arms',
                                         'Elbows', 'Wrists & Hands',
                                         'Lower back/flanks', 'Groin', 'Hips',
                                         'Buttocks', 'Cervical spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci) +
    geom_linerange(size = 1) +
    geom_point(size = 5) +
    coord_flip() +
    labs(title = 'Body sites',
         subtitle = '(Body sites with <10% pain; Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.3)) +
    theme_minimal(base_size = 18) +
    theme(plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by sex --#
```

```
pubs_nofacet.sex2 <- data_less10 %>%
    group_by(Pain_site, Sex) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet.sex2 <- pubs_nofacet.sex2 %>%
    select(-CD4_recent, -Age, -Pain_present, -Site, -Pain_def) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = ifelse(Pain_site == 'Lower back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Throat', 'Shoulder', 'Arms',
                                         'Elbows', 'Wrists & Hands',
                                         'Lower back/flanks', 'Groin', 'Hips',
                                         'Buttocks', 'Cervical spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Sex) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
```

```r
                        colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by sex',
         subtitle = '(Body sites with <10% pain; Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.3)) +
    scale_fill_manual(values = c('#FFFFFF', '#000000')) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.1),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by pain chronicity --#
pubs_nofacet.pain2 <- data_less10 %>%
    group_by(Pain_site, Pain_def) %>%
    nest() %>%
    # Boostrap data
    mutate(boot = map(.x = data,
                      ~ boot(data = .x[, 'Pain_present'],
                             statistic = prop_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup()

plot_nofacet.pain2 <- pubs_nofacet.pain2 %>%
```

```r
    select(-CD4_recent, -Age, -Pain_present, -Site, -Sex) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    mutate(Pain_site = ifelse(Pain_site == 'Lower back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Throat', 'Shoulder', 'Arms',
                                         'Elbows', 'Wrists & Hands',
                                         'Lower back/flanks', 'Groin', 'Hips',
                                         'Buttocks', 'Cervical spine'),
                              ordered = TRUE)) %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_def) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by pain chronicity',
         subtitle = '(Body sites with <10% pain; Point estimate with 95%CI)',
         y = 'Proportion of participants with pain') +
    scale_y_continuous(limits = c(0, 0.3)) +
    scale_fill_manual(values = c('#FFFFFF', '#000000')) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.1),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by age --#
pubs_nofacet.age2 <- data_less10 %>%
    select(-CD4_recent, -Site, -Sex, -Pain_def) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
```

```
                                              replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    group_by(Pain_site, Pain_present) %>%
    nest() %>%
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = median_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup() %>%
    mutate(Pain_site = ifelse(Pain_site == 'Lower back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Throat', 'Shoulder', 'Arms',
                                         'Elbows', 'Wrists & Hands',
                                         'Lower back/flanks', 'Groin', 'Hips',
                                         'Buttocks', 'Cervical spine'),
                              ordered = TRUE)) %>%
    select(-Age) %>%
    distinct()

plot_nofacet.age2 <- pubs_nofacet.age2 %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_present) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
```

```
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by median age',
         subtitle = '(Body sites with <10% pain; Point estimate with 95%CI)',
         y = 'Age (years)') +
    scale_fill_manual(values = c('#FFFFFF', '#000000'),
                      labels = c('No pain', 'Pain present')) +
    scale_y_continuous(limits = c(27, 47),
                       breaks = c(27, 31, 35, 39, 43, 47)) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.1),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

#-- Body sites by CD4 --#
pubs_nofacet.cd42 <- data_less10 %>%
    select(-Age, -Site, -Sex, -Pain_def) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
    group_by(Pain_site, Pain_present) %>%
    nest() %>%
    mutate(boot = map(.x = data,
                      ~ boot(data = .x,
                             statistic = median_func,
                             R = 999,
                             stype = 'i',
                             parallel = 'multicore',
                             ncpus = 4))) %>%
    # Get CI
    mutate(ci = map(.x = boot,
                    ~ boot.ci(.x, type = 'perc'))) %>%
    # Extract ci data
    mutate(point_est = map(.x = ci,
                           ~ .x$t0),
           lower_ci = map(.x = ci,
                          ~ .x$percent[[4]]),
           upper_ci = map(.x = ci,
                          ~ .x$percent[[5]])) %>%
    # Remove columns
    select(-boot, -ci) %>%
    # Unnest
```

136

```r
    unnest(cols = c(data, point_est, lower_ci, upper_ci)) %>%
    ungroup() %>%
    mutate(Pain_site = ifelse(Pain_site == 'Lower back',
                              yes = 'Lower back/flanks',
                              no = Pain_site)) %>%
    mutate(Pain_site = factor(Pain_site,
                              levels = c('Throat', 'Shoulder', 'Arms',
                                         'Elbows', 'Wrists & Hands',
                                         'Lower back/flanks', 'Groin', 'Hips',
                                         'Buttocks', 'Cervical spine'),
                              ordered = TRUE)) %>%
    select(-CD4_recent) %>%
    distinct()

plot_nofacet.cd42 <- pubs_nofacet.cd42 %>%
    ggplot(data = .) +
    aes(x = fct_rev(Pain_site),
        y = point_est,
        ymin = lower_ci,
        ymax = upper_ci,
        fill = Pain_present) +
    geom_linerange(position = position_dodge2(width = 0.6),
                   size = 1,
                   colour = '#000000') +
    geom_point(shape = 21,
               colour = '#000000',
               position = position_dodge2(width = 0.6),
               size = 4,
               stroke = 1) +
    coord_flip() +
    labs(title = 'Body sites by median CD4 T-cell count',
         subtitle = '(Body sites with <10% pain; Point estimate with 95%CI)',
         y = expression('CD4 T-cell count (cell.mm'^-3*')')) +
    scale_fill_manual(values = c('#FFFFFF', '#000000'),
                      labels = c('No pain', 'Pain present')) +
    scale_y_continuous(limits = c(50, 600),
                       breaks = c(100, 200, 300, 400, 500, 600)) +
    theme_minimal(base_size = 18) +
    theme(legend.position = c(0.8, 0.95),
          legend.title = element_blank(),
          plot.title = element_text(size = 18),
          plot.subtitle = element_text(size = 12),
          axis.title.y = element_blank(),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'),
          axis.line = element_line(size = 0.5),
          axis.ticks = element_line(size = 0.5))

# Patchwork
pub_plot2 <- plot_nofacet2 + plot_nofacet.sex2 + plot_nofacet.pain2 +
    plot_nofacet.age2 + plot_nofacet.cd42 +
    plot_layout(ncol = 2)

# Save
```

```
ggsave(filename = 'figures/figure_supp2.png',
       width = 14,
       height = 18)
```
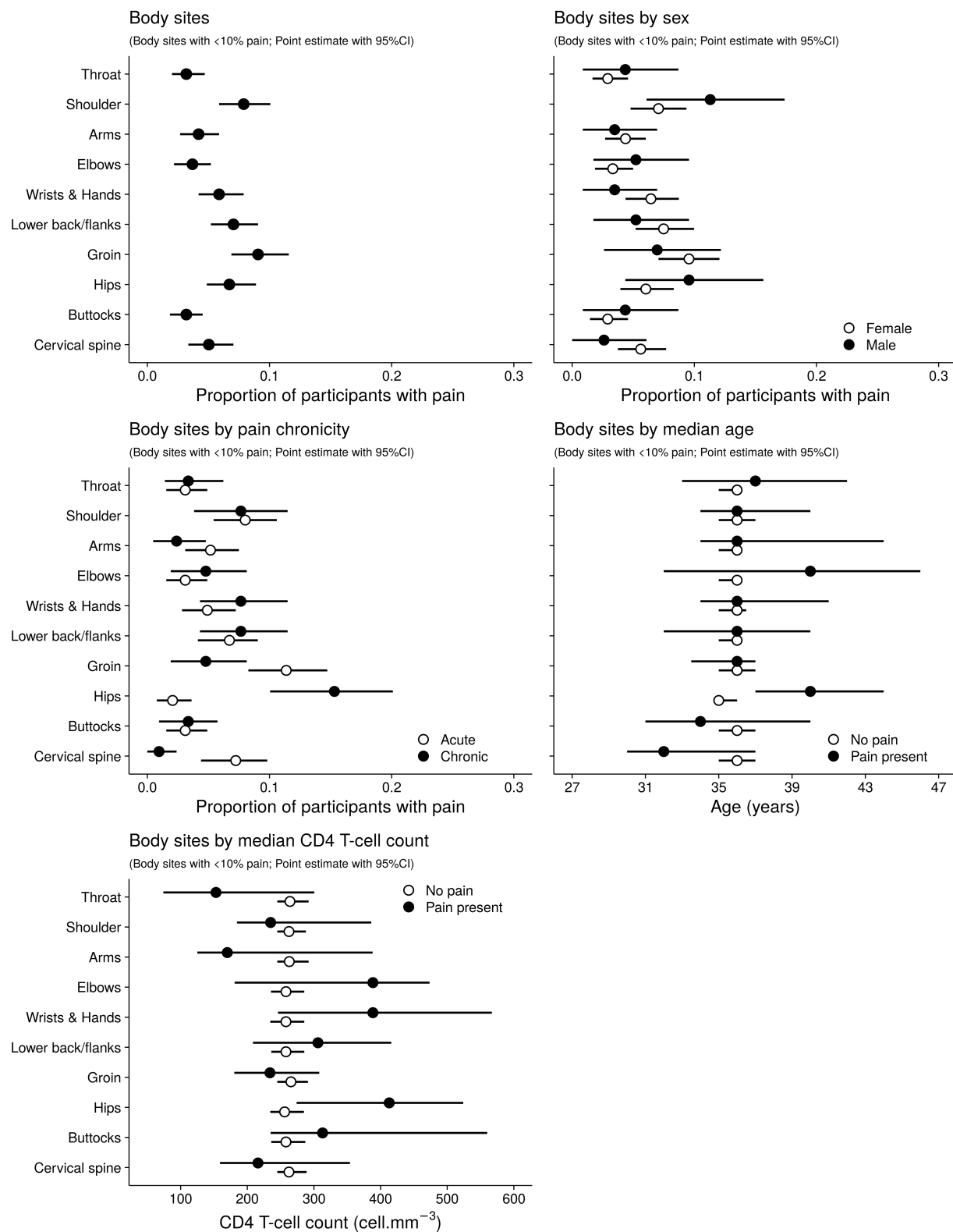
Body sites
(Body sites with <10% pain; Point estimate with 95%CI)

Body sites by sex
(Body sites with <10% pain; Point estimate with 95%CI)

Body sites by pain chronicity
(Body sites with <10% pain; Point estimate with 95%CI)

Body sites by median age
(Body sites with <10% pain; Point estimate with 95%CI)

Body sites by median CD4 T-cell count
(Body sites with <10% pain; Point estimate with 95%CI)

Figure 3: Body sites with a point estimate pain proportion <10%

139

10 Session information

```r
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 10 (buster)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.3.5.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=C
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] patchwork_1.0.0 lme4_1.1-23    Matrix_1.2-18   boot_1.3-24
##  [5] knitr_1.28      skimr_2.1.1    forcats_0.5.0   stringr_1.4.0
##  [9] dplyr_0.8.5     purrr_0.3.4    readr_1.3.1     tidyr_1.0.2
## [13] tibble_3.0.1    ggplot2_3.3.0  tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6     lubridate_1.7.8 lattice_0.20-38 utf8_1.1.4
##  [5] assertthat_0.2.1 digest_0.6.25   R6_2.4.1         cellranger_1.1.0
##  [9] repr_1.1.0       backports_1.1.6 reprex_0.3.0     evaluate_0.14
## [13] highr_0.8        httr_1.4.1      pillar_1.4.3     rlang_0.4.5
## [17] readxl_1.3.1     rstudioapi_0.11 minqa_1.2.4      nloptr_1.2.2.1
## [21] rmarkdown_2.1    labeling_0.3    splines_3.6.3    statmod_1.4.34
## [25] munsell_0.5.0    broom_0.5.6     compiler_3.6.3   modelr_0.1.6
## [29] xfun_0.13        pkgconfig_2.0.3 base64enc_0.1-3 htmltools_0.4.0
## [33] tidyselect_1.0.0 fansi_0.4.1     crayon_1.3.4     dbplyr_1.4.3
## [37] withr_2.2.0      MASS_7.3-51.5   grid_3.6.3       nlme_3.1-144
## [41] jsonlite_1.6.1   gtable_0.3.0    lifecycle_0.2.0 DBI_1.1.0
## [45] magrittr_1.5     scales_1.1.0    cli_2.0.2        stringi_1.4.6
## [49] farver_2.0.3     fs_1.4.1        xml2_1.3.2       ellipsis_0.3.0
## [53] generics_0.0.2   vctrs_0.2.4     tools_3.6.3      glue_1.4.0
## [57] hms_0.5.3        parallel_3.6.3  yaml_2.2.1       colorspace_1.4-1
## [61] rvest_0.3.5      haven_2.2.0
```