Supplement 2

Sites of pain

Peter Kamerman

Last knitted: 10 May 2020

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

skim_variable	n_missing	complete_rate	n_unique	top_counts
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 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

```
# 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 %>%
    # 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,
                      ~ 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 6: 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 7: 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 8: 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 9: 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 10: Lower limbs

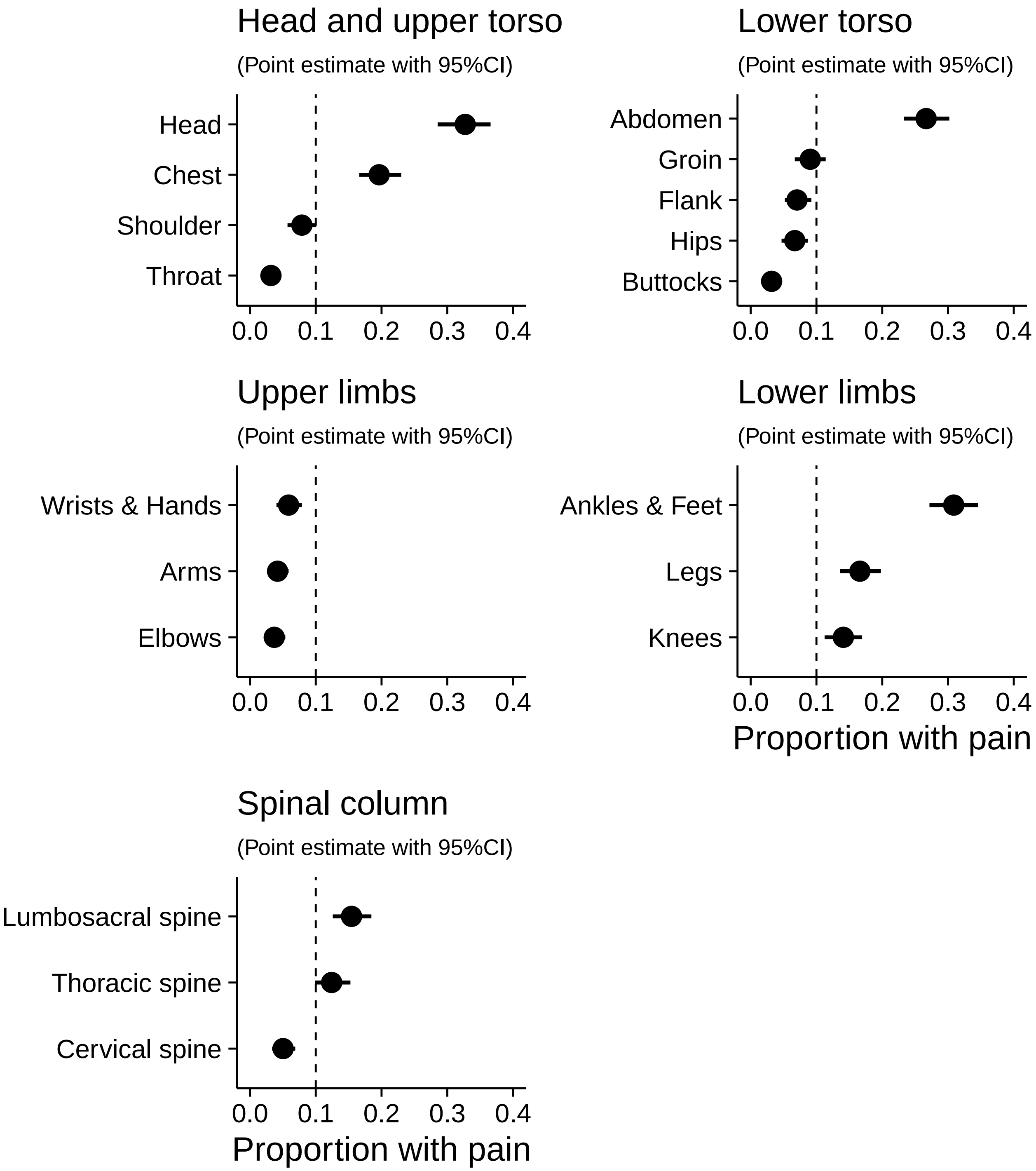
body_site	point_est	lower_ci	upper_ci
Legs	0.17	0.14	0.20
Knees	0.14	0.11	0.17
Ankles & Feet	0.31	0.27	0.35

Table 11: other

body_site	point_est	lower_ci	upper_ci
Site	0	0	0

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

```
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
```



```
# 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
    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 12: 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 13: 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 14: 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 15: Spinal column

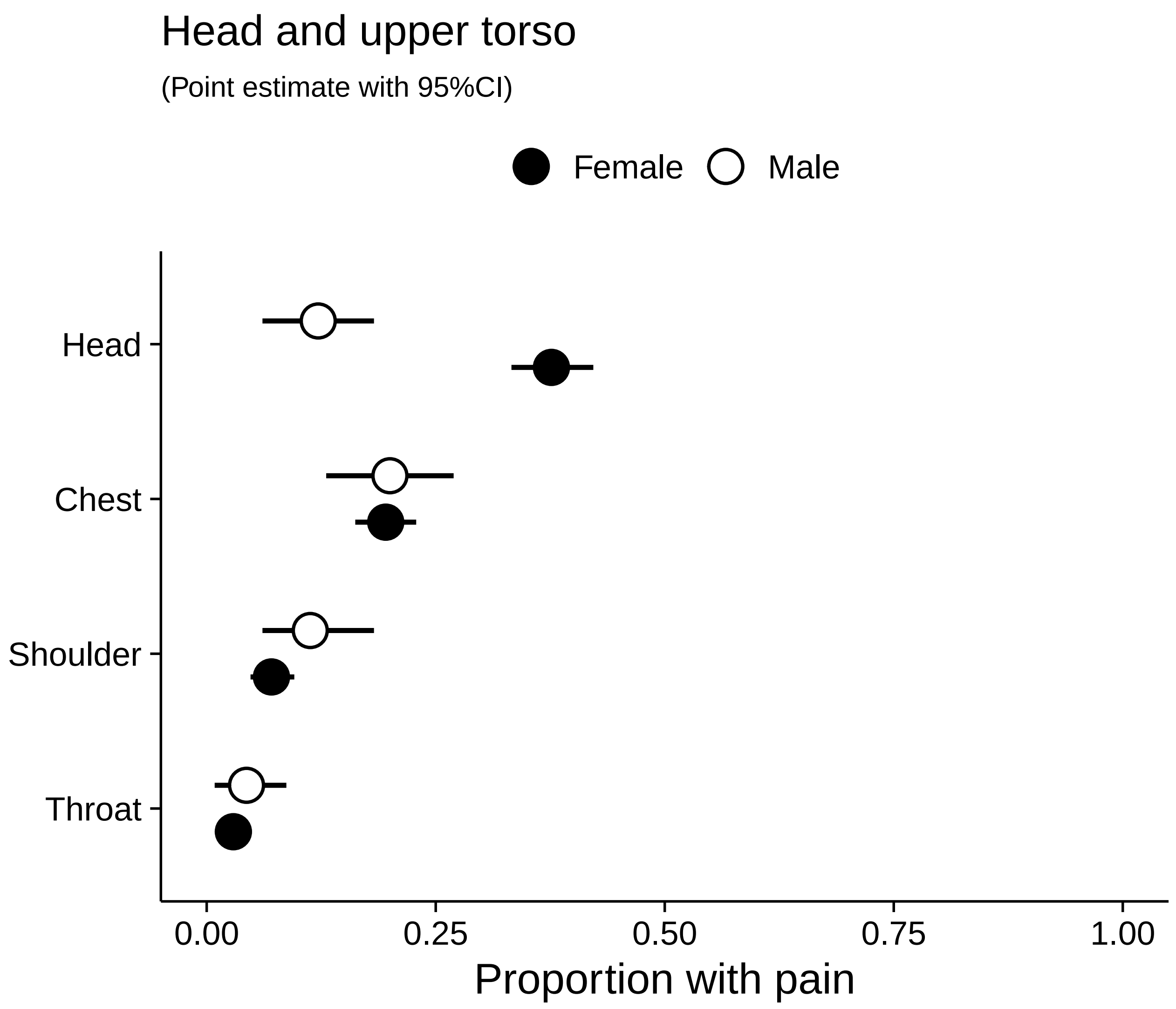
Sex	body_site	point_est	lower_ci	upper_ci
Female	Cervical spine	0.06	0.04	0.08
Female	Thoracic spine	0.14	0.11	0.17
Female	Lumbosacral spine	0.16	0.12	0.19
Male	Cervical spine	0.03	0.00	0.06
Male	Thoracic spine	0.07	0.03	0.12
Male	Lumbosacral spine	0.14	0.08	0.20

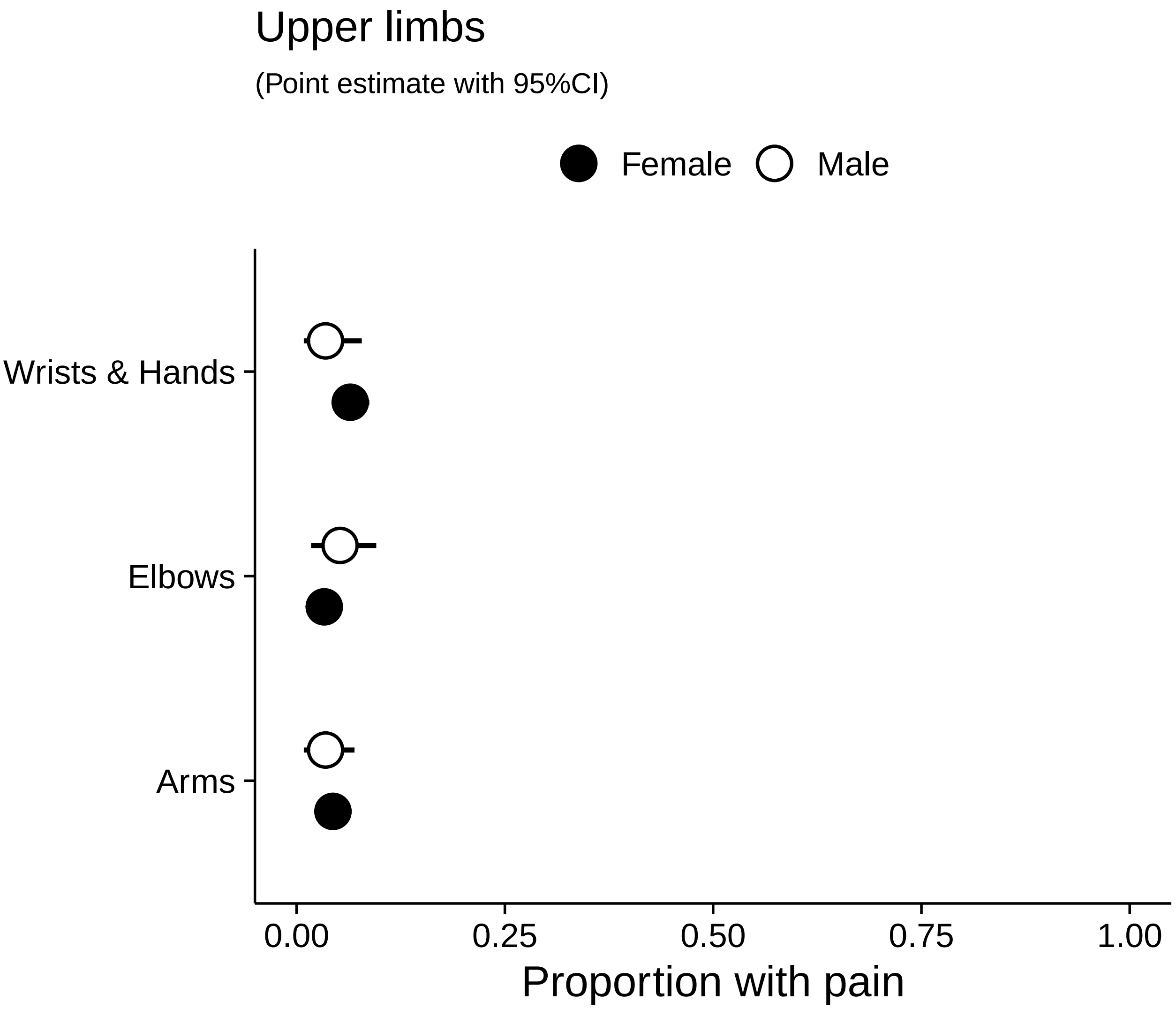
Table 16: Lower limbs

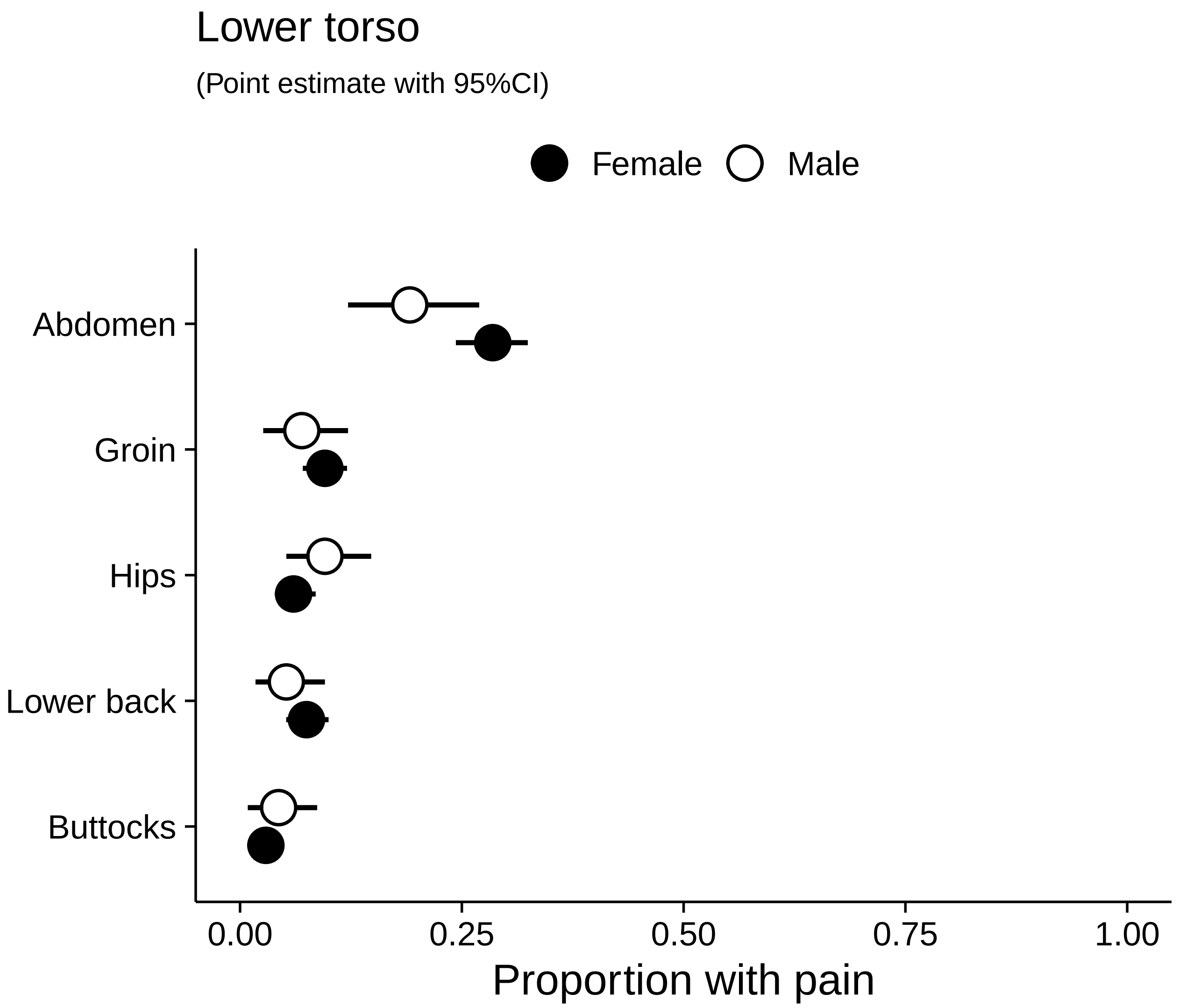
Sex	body_site	point_est	lower_ci	upper_ci
Female	Legs	0.15	0.12	0.18
Female	Knees	0.13	0.10	0.16
Female	Ankles & Feet	0.28	0.25	0.32
Male	Legs	0.23	0.16	0.31
Male	Knees	0.19	0.12	0.26
Male	Ankles & Feet	0.42	0.33	0.50

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

```r
walk(sex_boot2$plots, ~ print(.x))
```

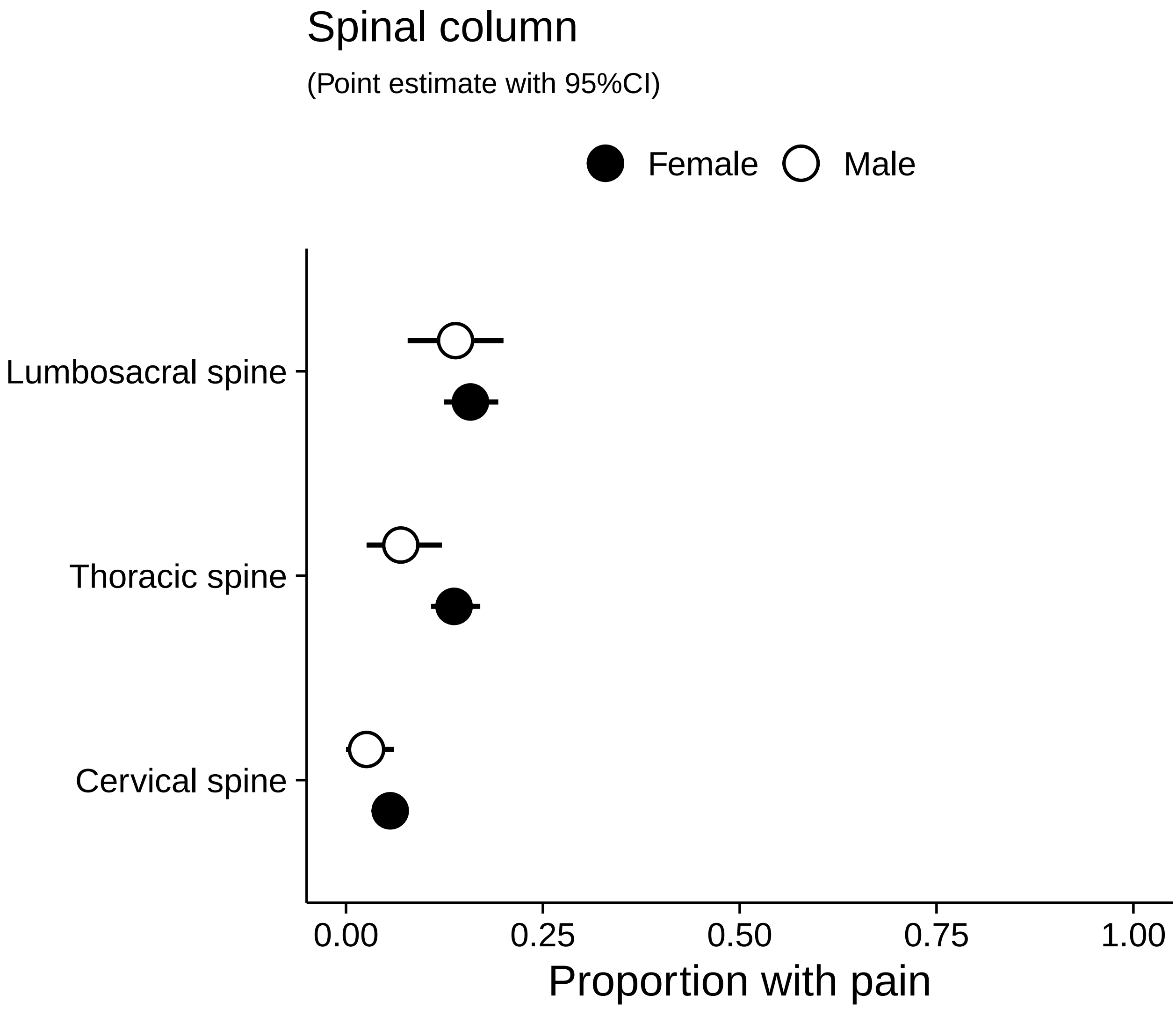

Head and upper torso
(Point estimate with 95%CI)
Female
Male
Head
Chest
Shoulder
Throat
0.00
0.25
0.50
0.75
1.00
Proportion with pain

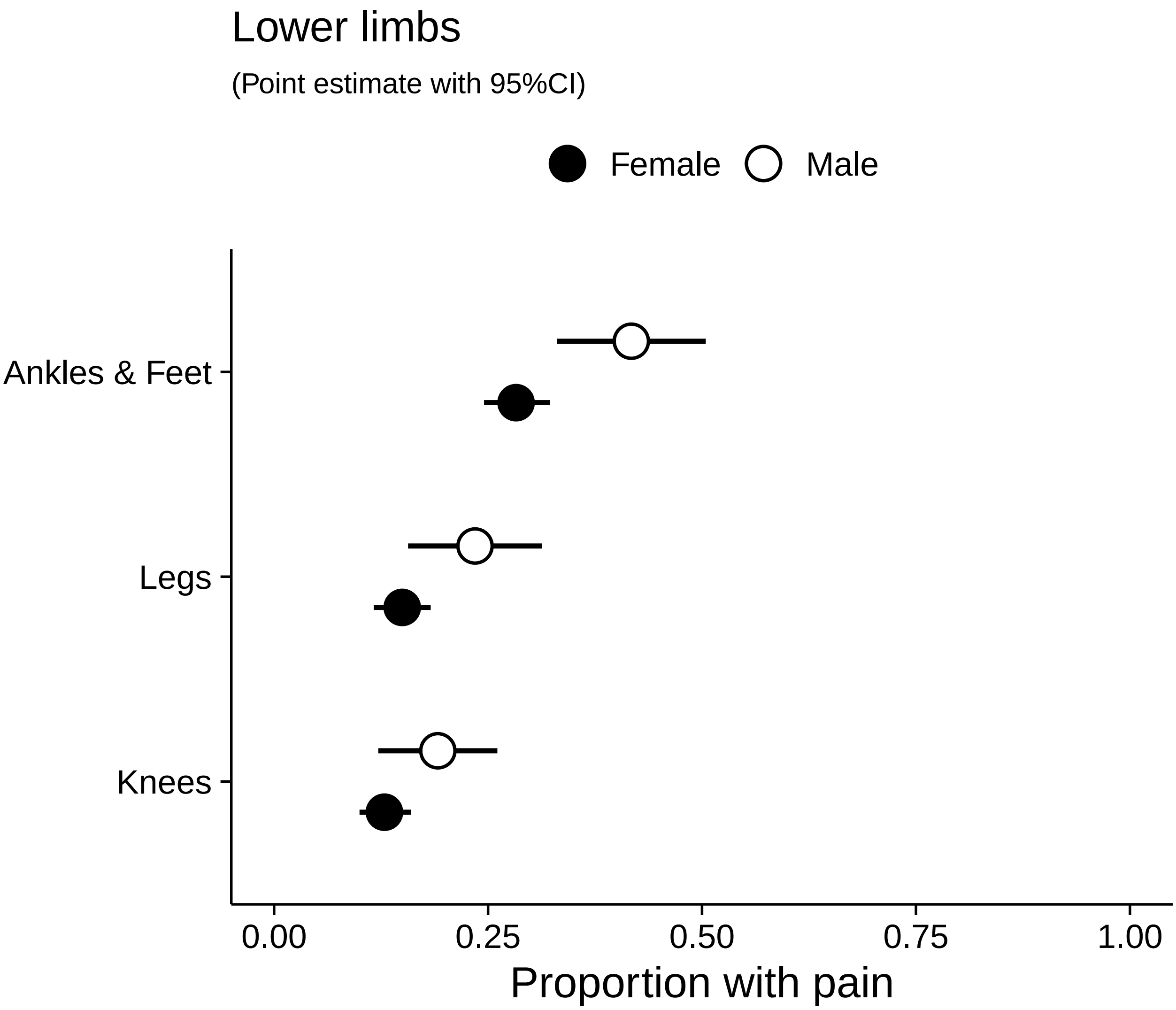

Upper limbs
(Point estimate with 95%CI)
● Female
○ Male
Wrists & Hands
Elbows
Arms
0.00
0.25
0.50
0.75
1.00
Proportion with pain



Lower torso

(Point estimate with 95%CI)


Spinal column
(Point estimate with 95%CI)
● Female
○ Male
Lumbosacral spine
Thoracic spine
Cervical spine
0.00
0.25
0.50
0.75
1.00
Proportion with pain



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 have analysed age as a continuous variable.

5.1 Process data

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

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

```r
# 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)

# 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 17: 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 = 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)) +
                           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 18: 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 19: 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 20: 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

age_group	point_est	lower_ci	upper_ci
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 21: 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 22: 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 23: 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 24: 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 25: 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 26: 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 27: 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 28: 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 29: 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

age_group	point_est	lower_ci	upper_ci
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 30: 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
58-67	0.05	0.00	0.15

Table 31: 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 32: 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 33: 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 34: Ankles & Feet (Age group 68-77 years removed because n = 2)

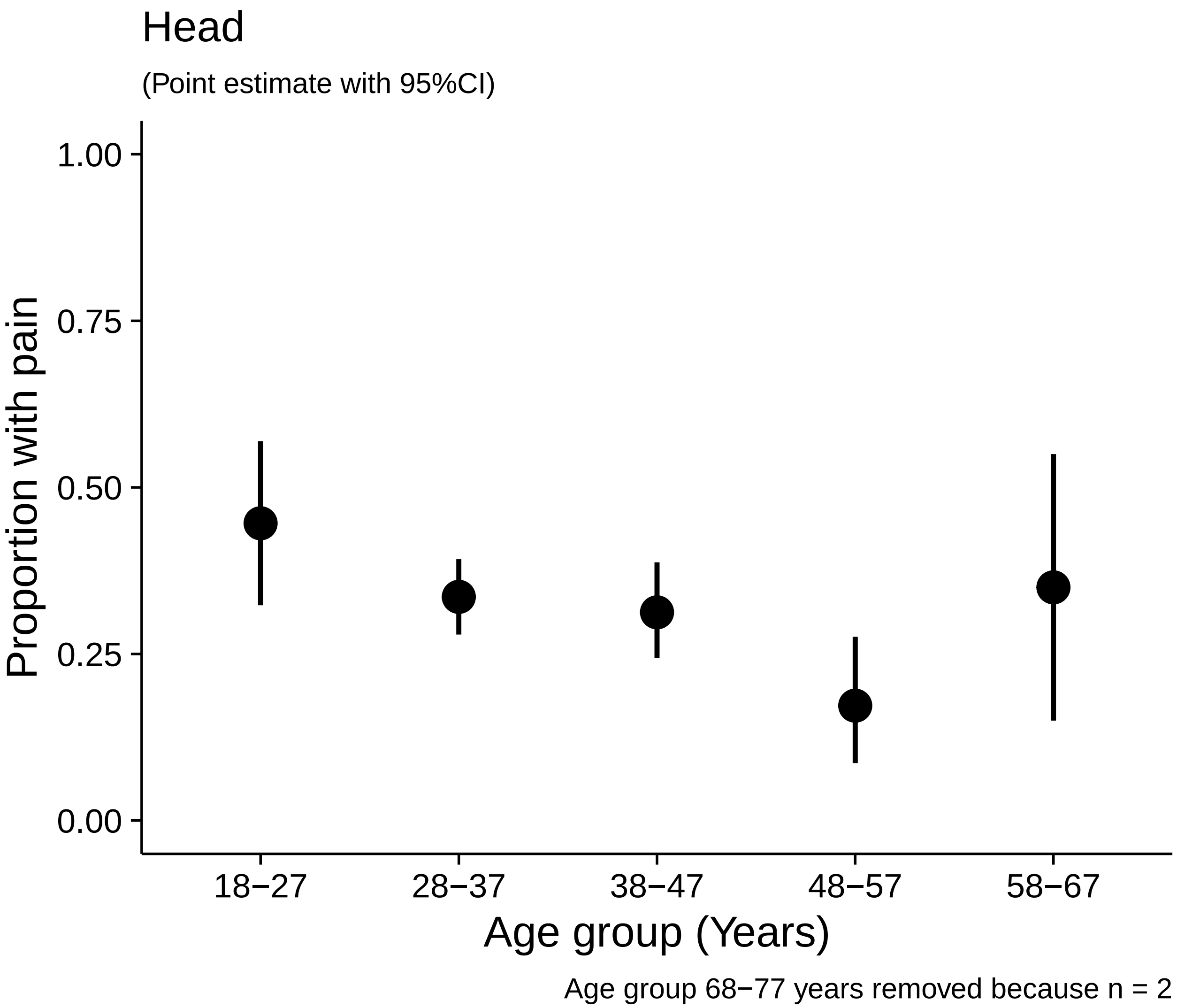
age_group	point_est	lower_ci	upper_ci
18-27	0.22	0.12	0.32
28-37	0.24	0.19	0.29
38-47	0.37	0.29	0.44
48-57	0.52	0.40	0.64
58-67	0.55	0.35	0.75

Table 35: 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
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

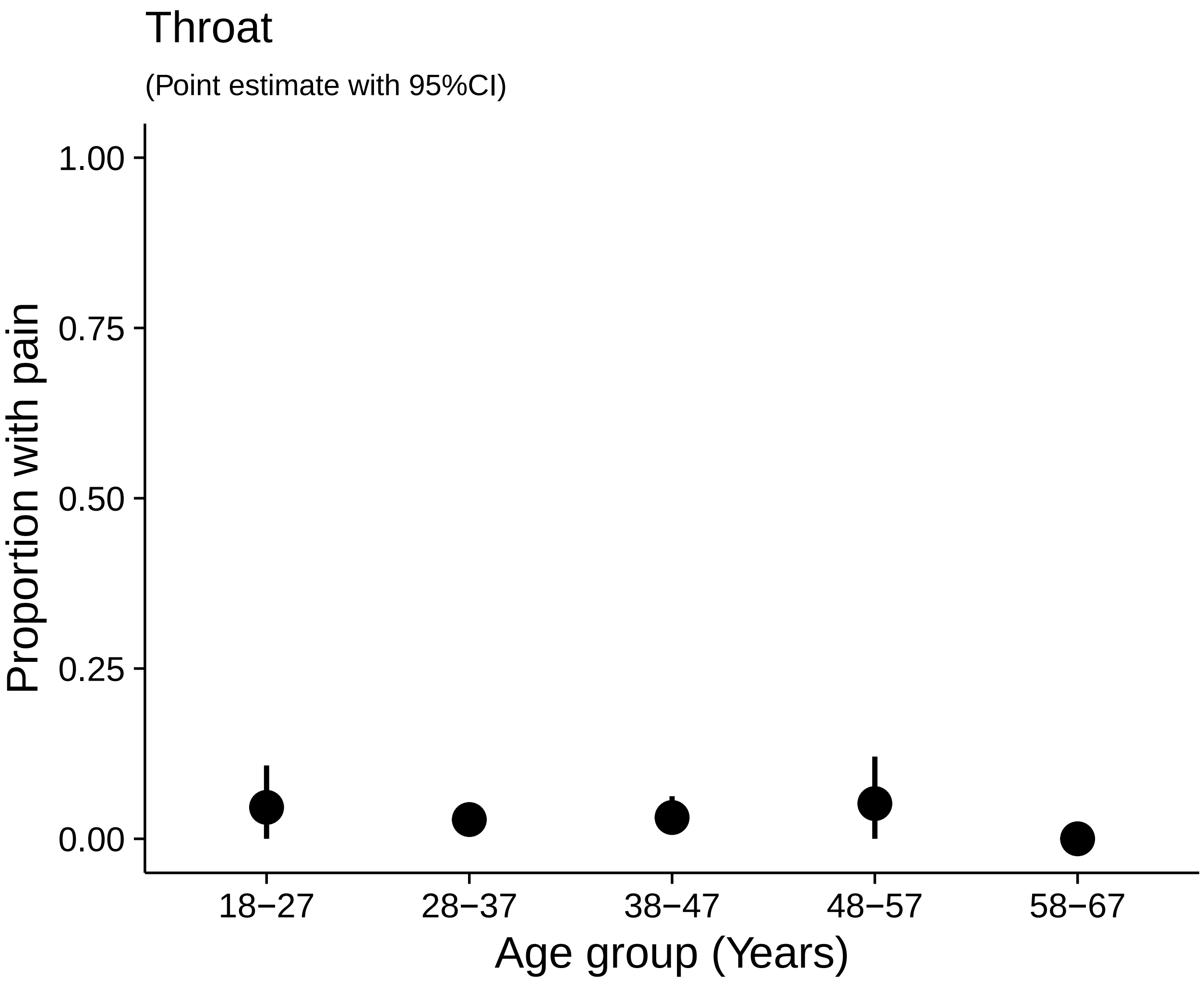
```
walk(age_boot2$plots, ~ print(.x))
```



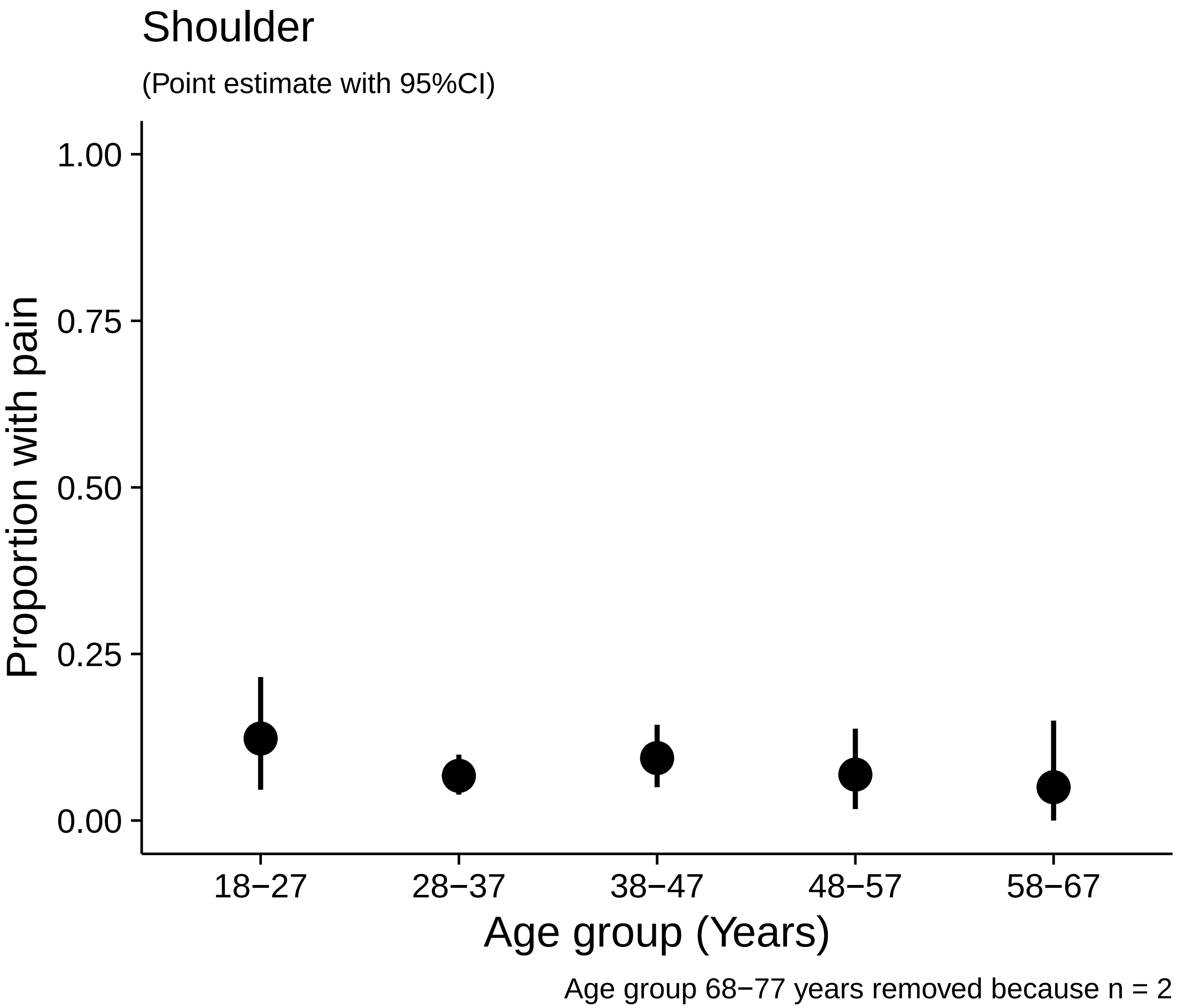
Head

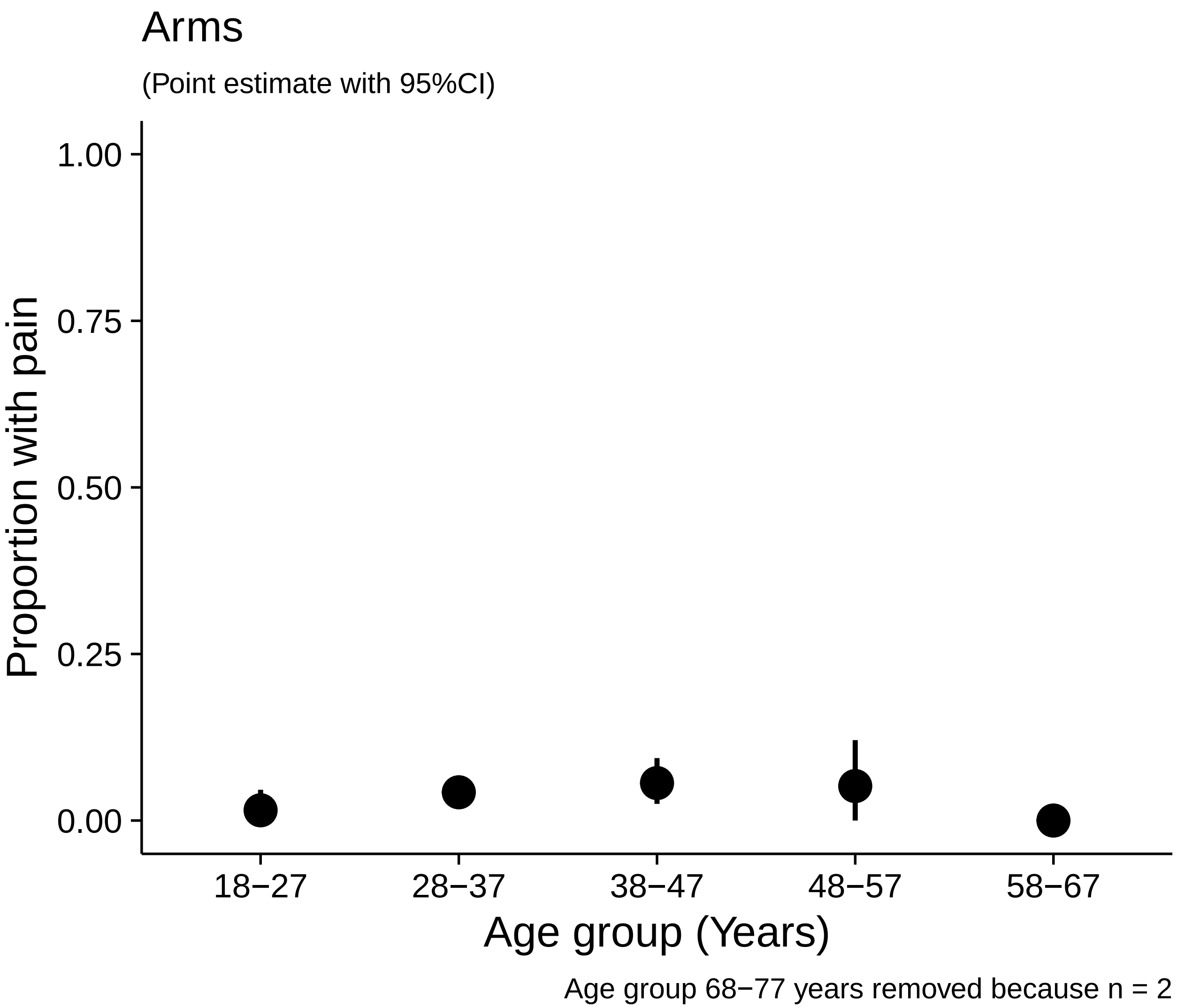
(Point estimate with 95%CI)

Age group 68−77 years removed because n = 2



Age group 68−77 years removed because n = 2

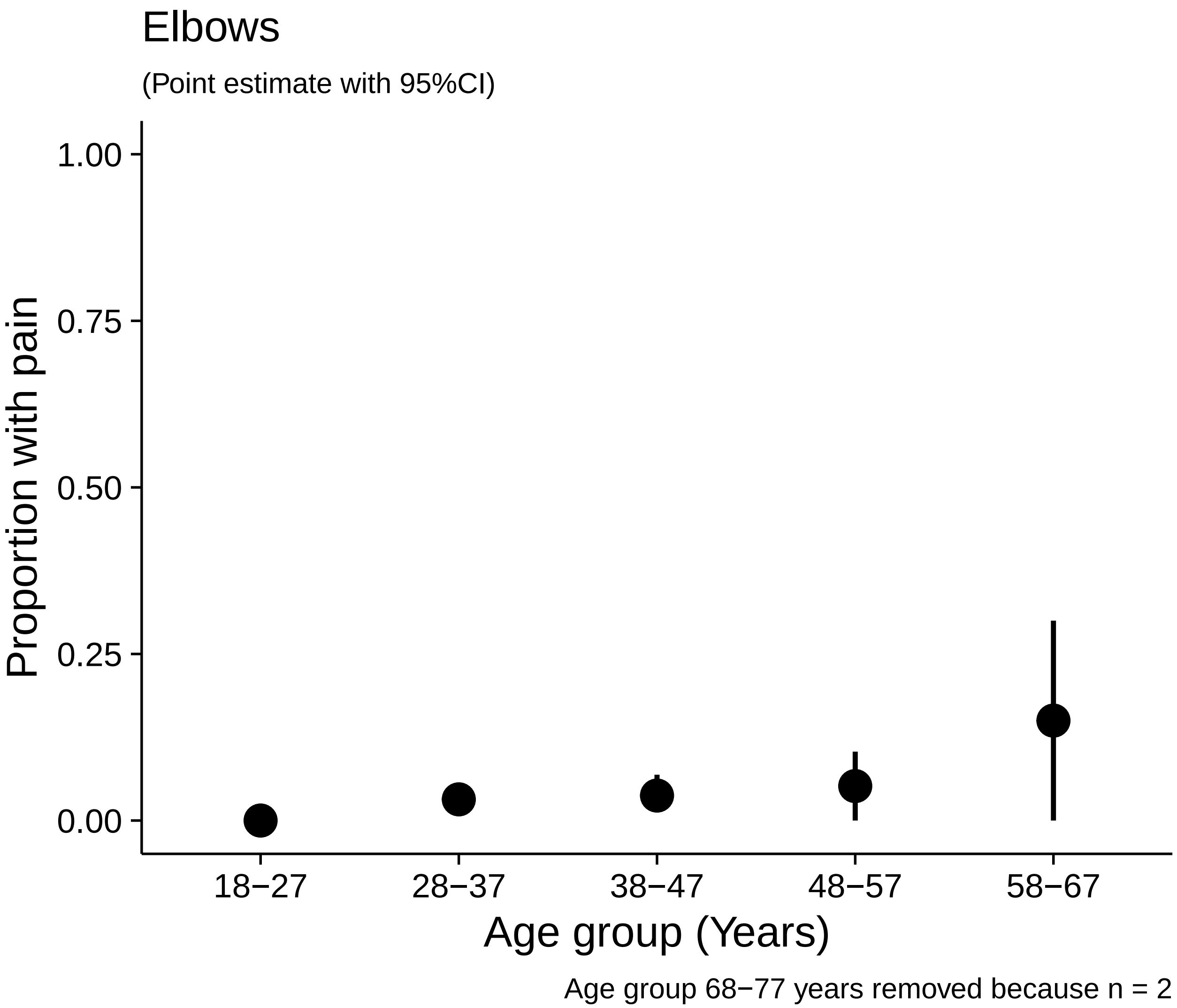

Shoulder
(Point estimate with 95%CI)
1.00
0.75
0.50
0.25
0.00
Proportion with pain
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2



Arms

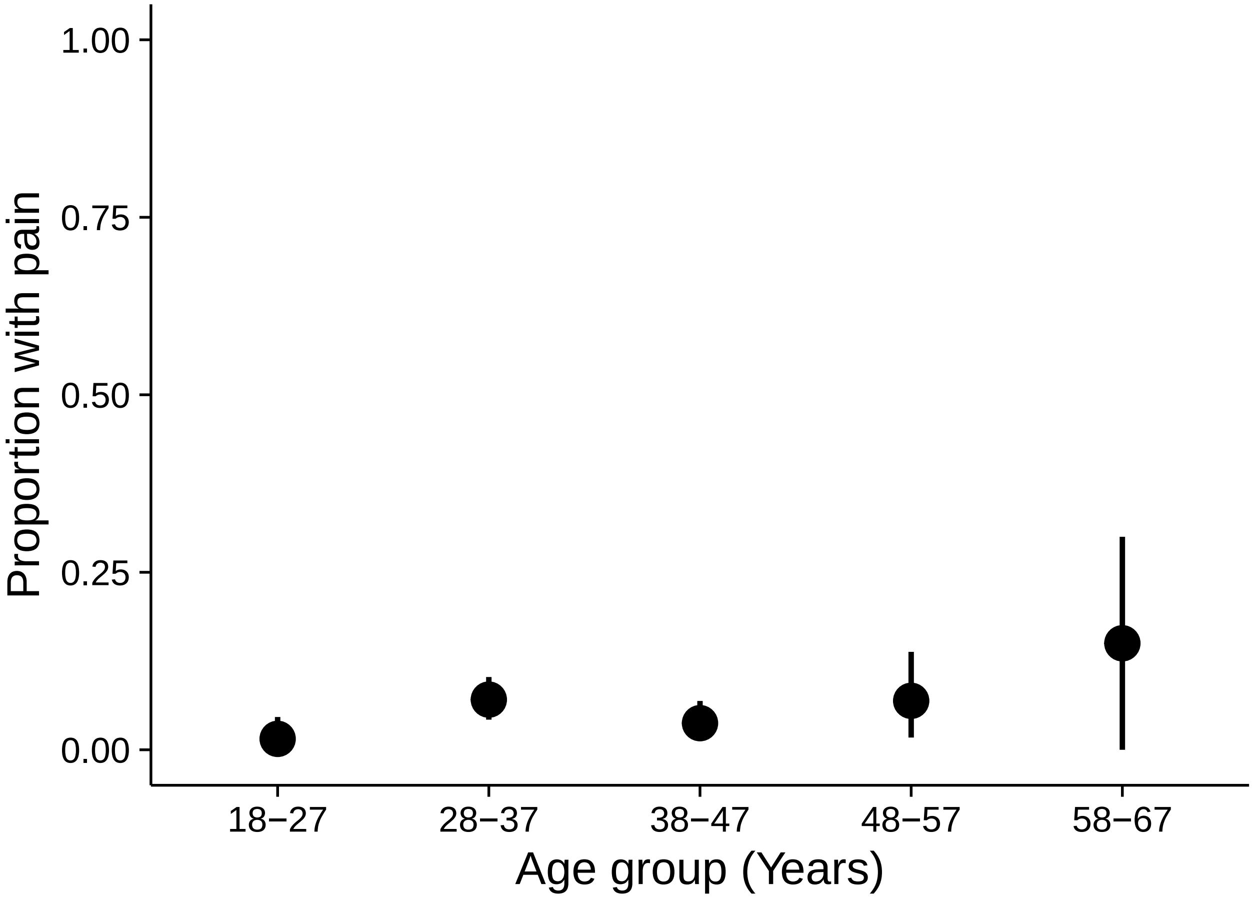
(Point estimate with 95%CI)

Age group 68−77 years removed because n = 2

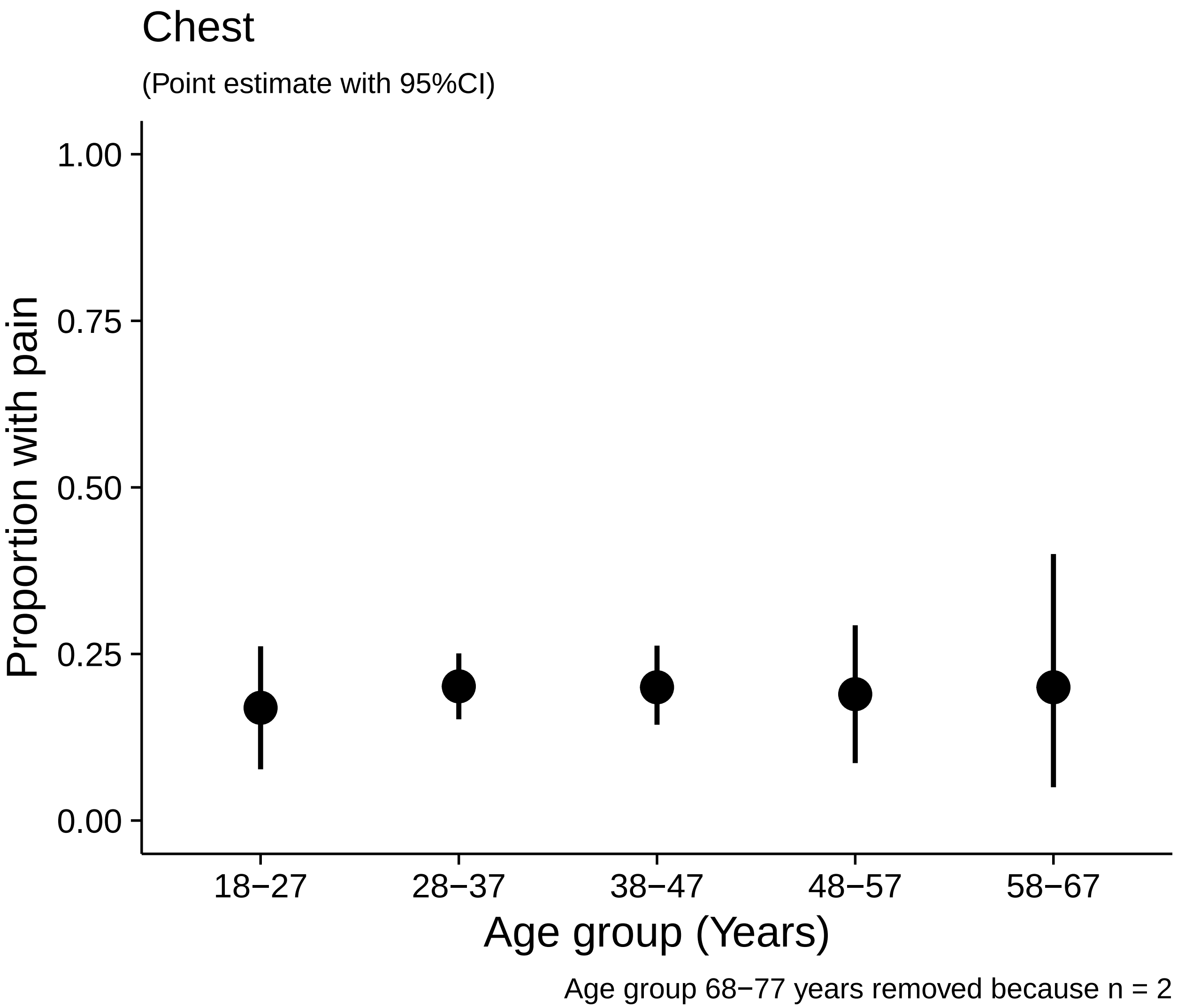

Elbows
(Point estimate with 95%CI)
1.00
0.75
0.50
0.25
0.00
Proportion with pain
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2

Wrists & Hands

(Point estimate with 95%CI)

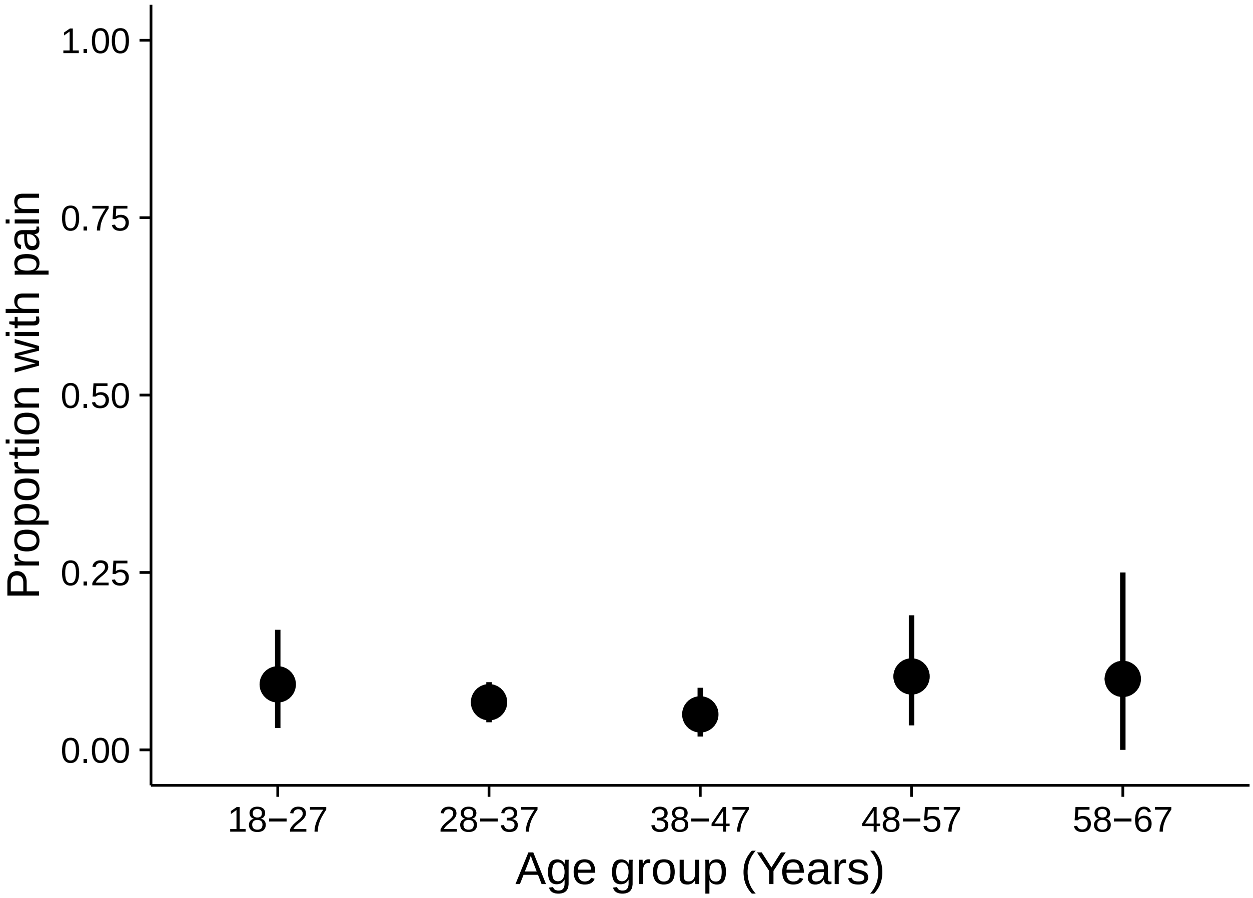


Age group 68−77 years removed because n = 2


Chest
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2

Lower back

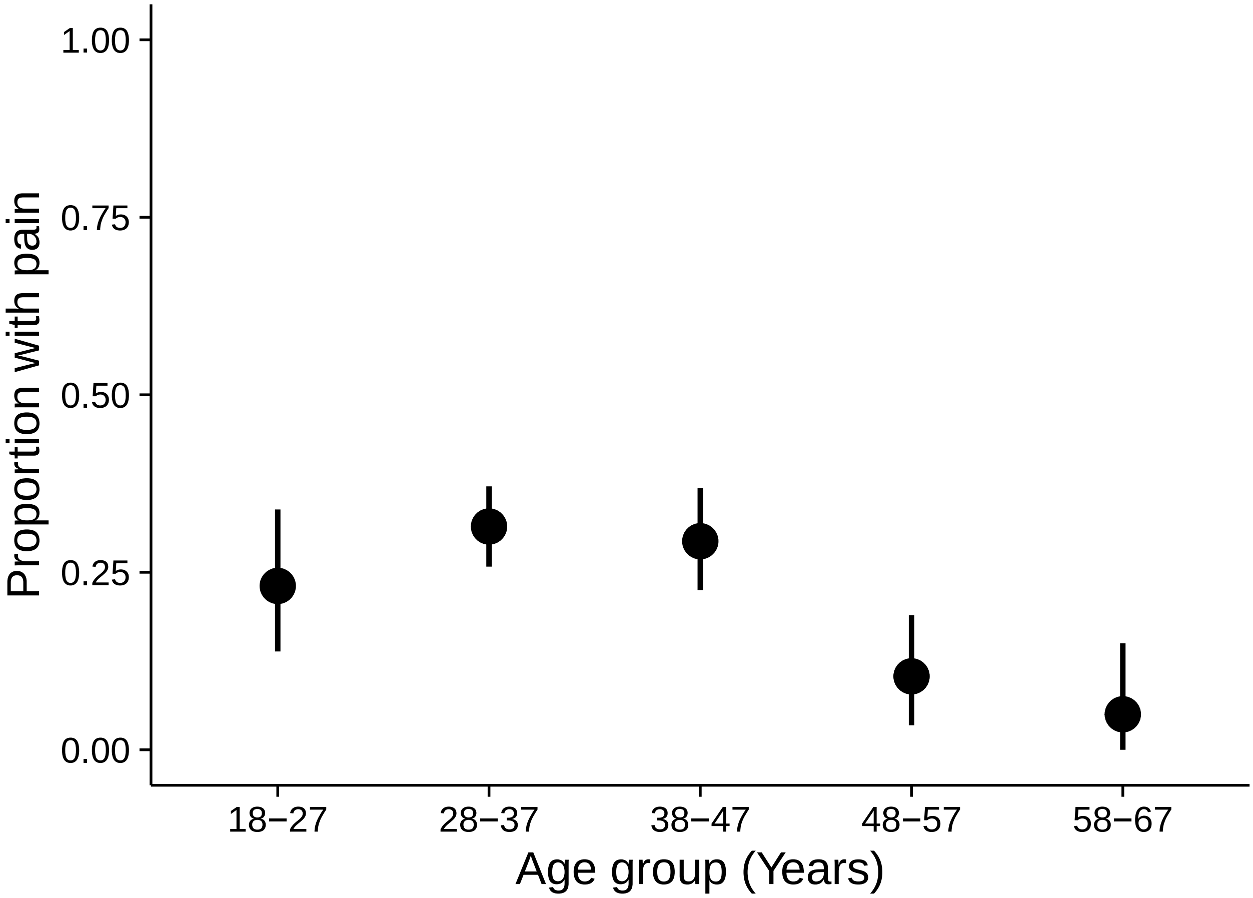
(Point estimate with 95%CI)



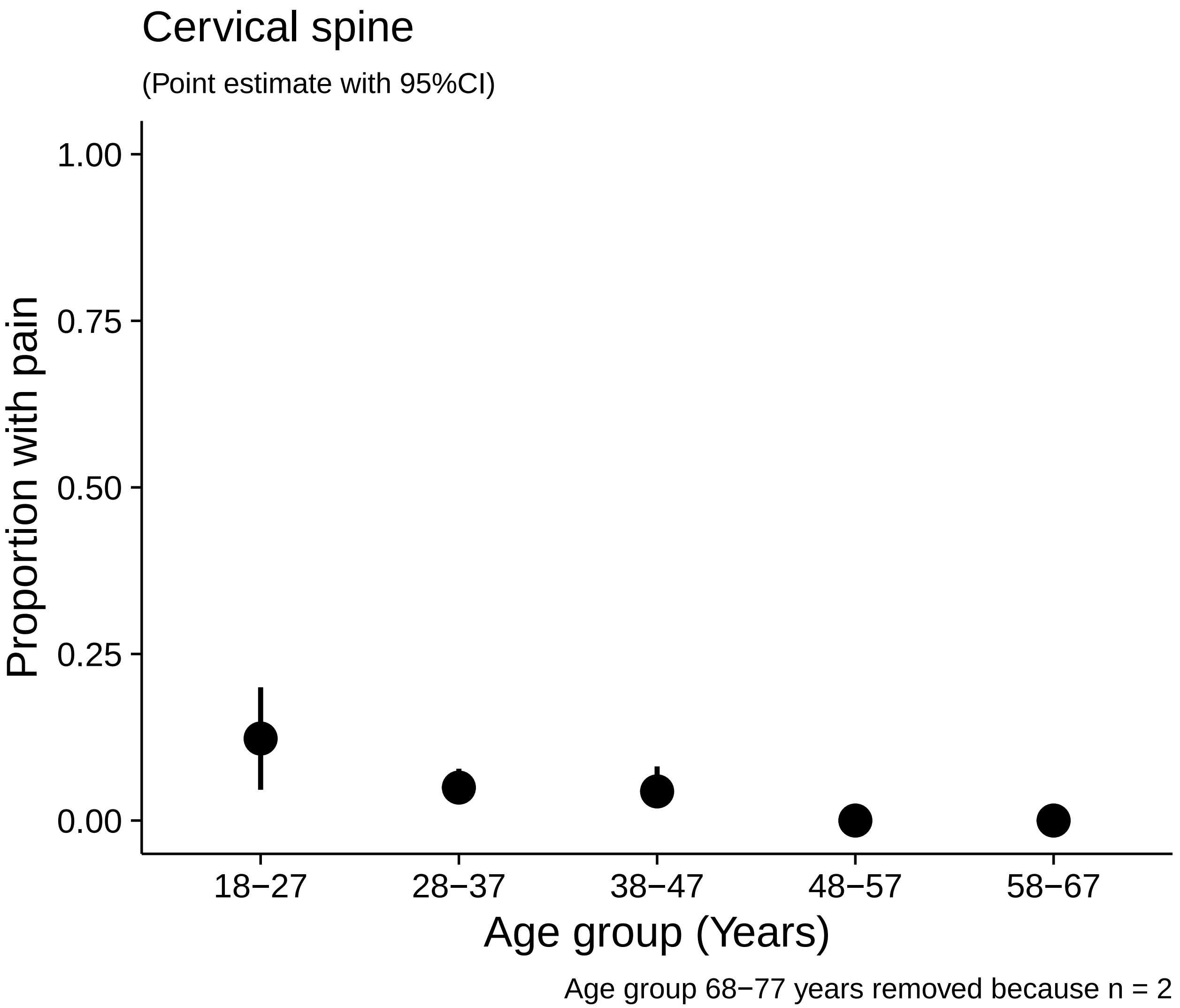
Age group 68−77 years removed because n = 2

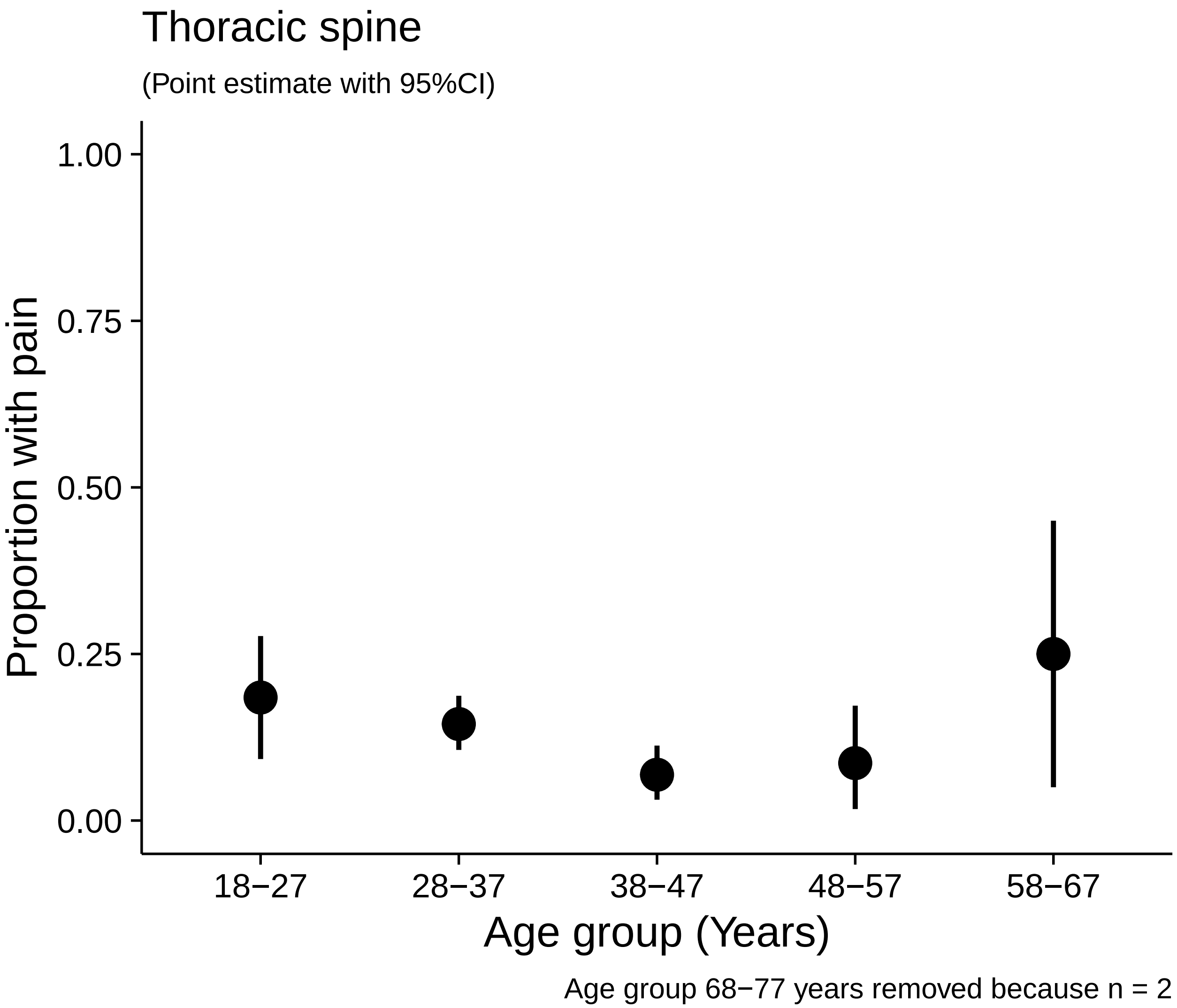
Abdomen

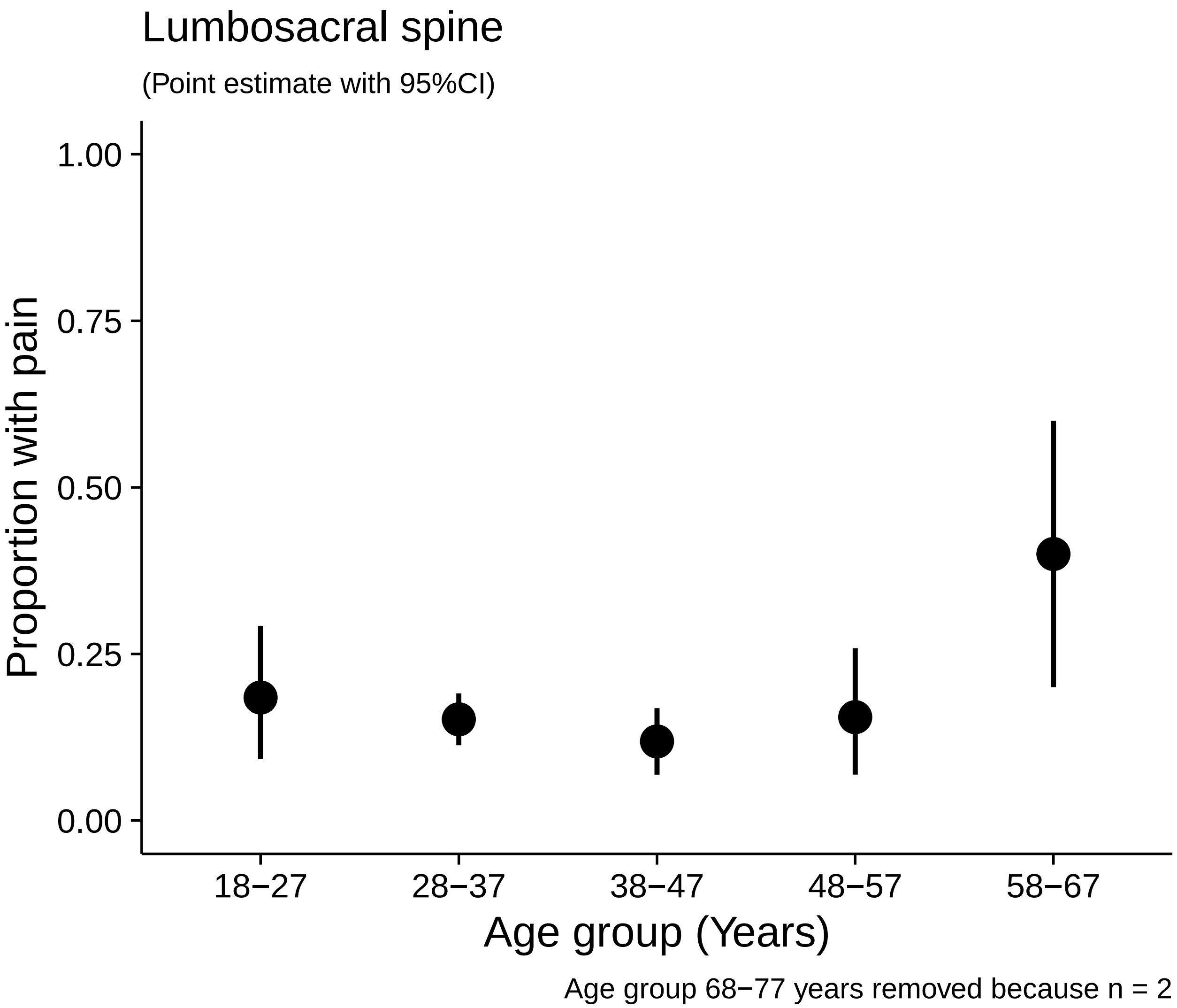
(Point estimate with 95%CI)

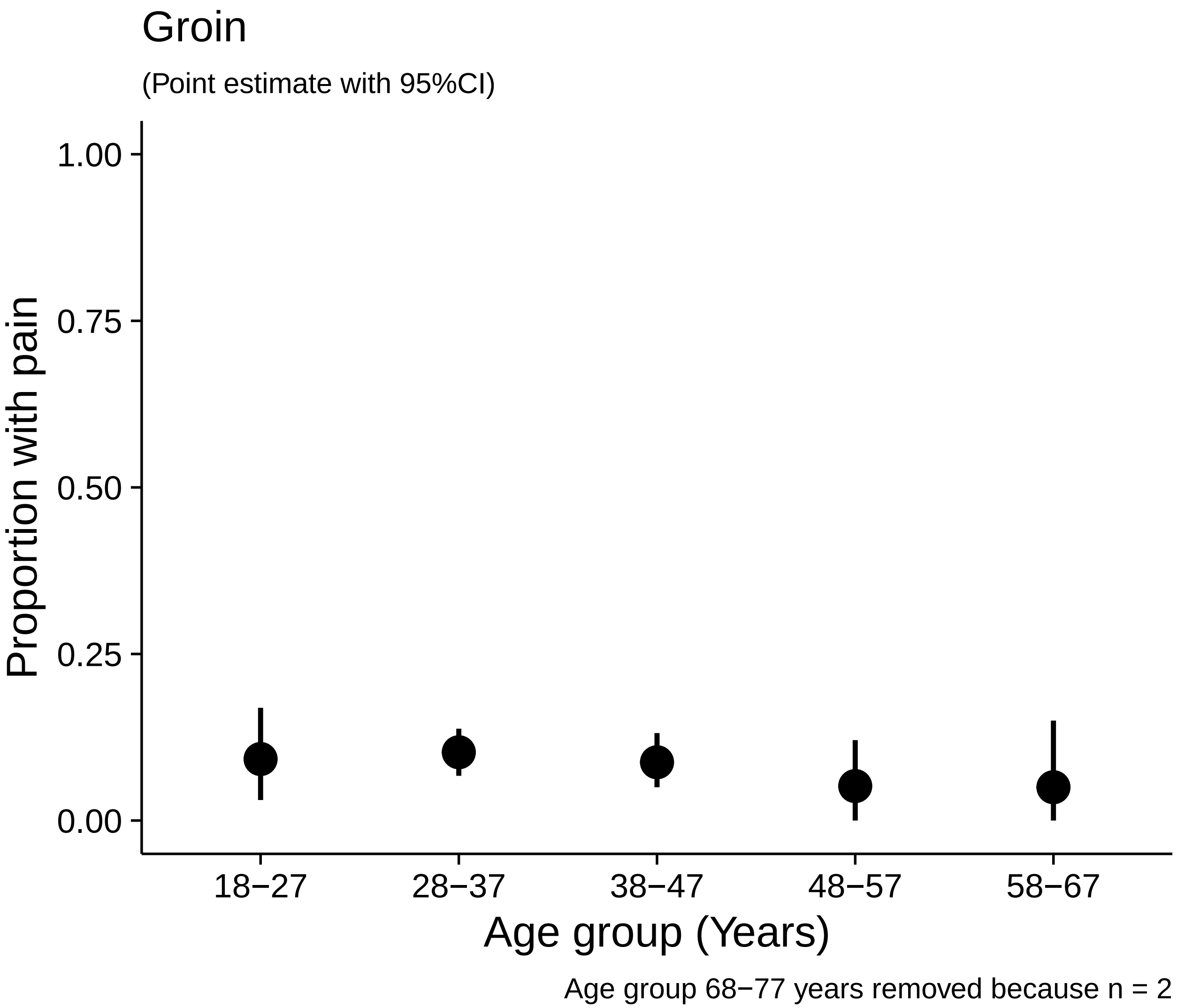


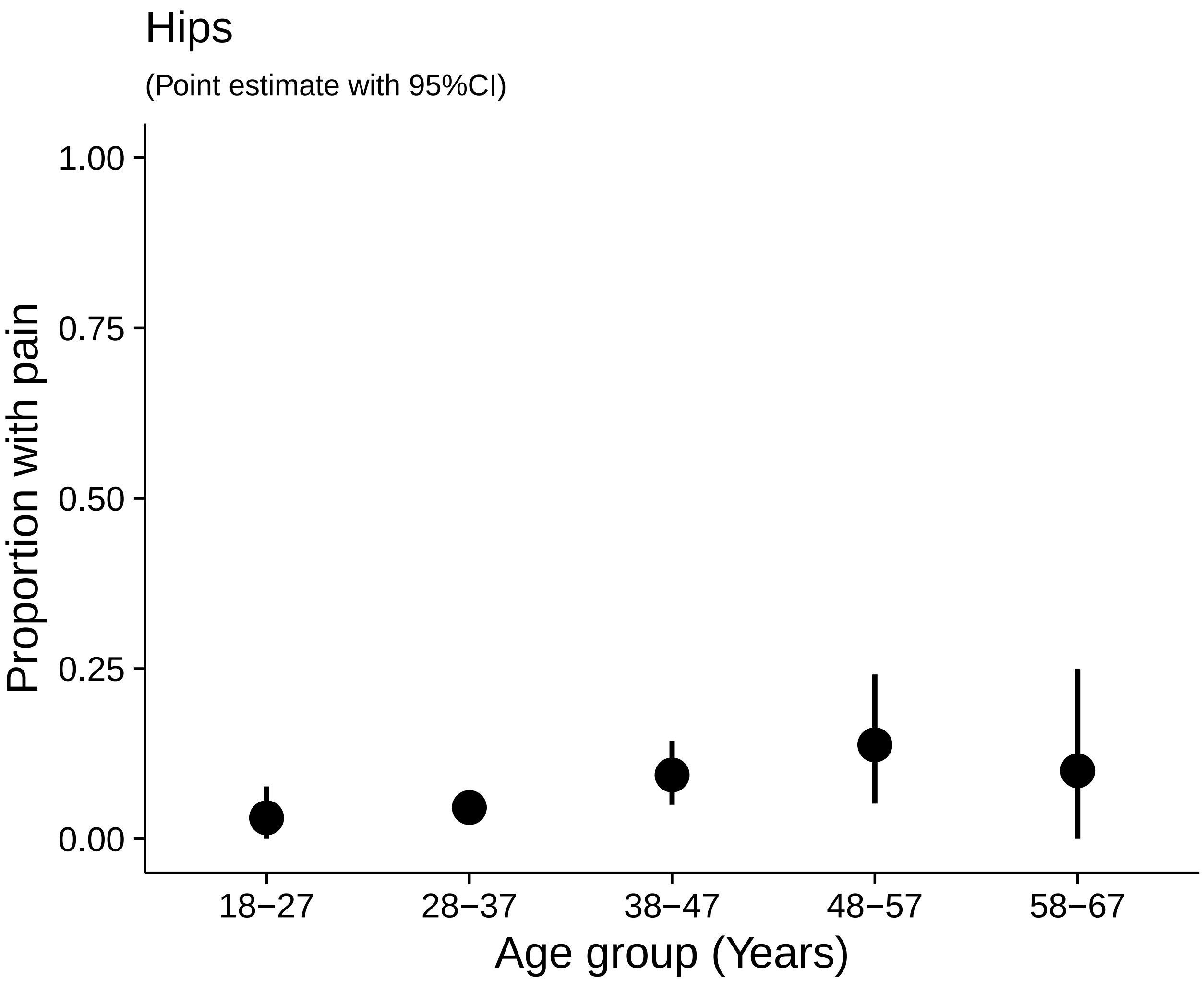
Age group 68−77 years removed because n = 2


Cervical spine
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2


Thoracic spine
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2


Lumbosacral spine
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2

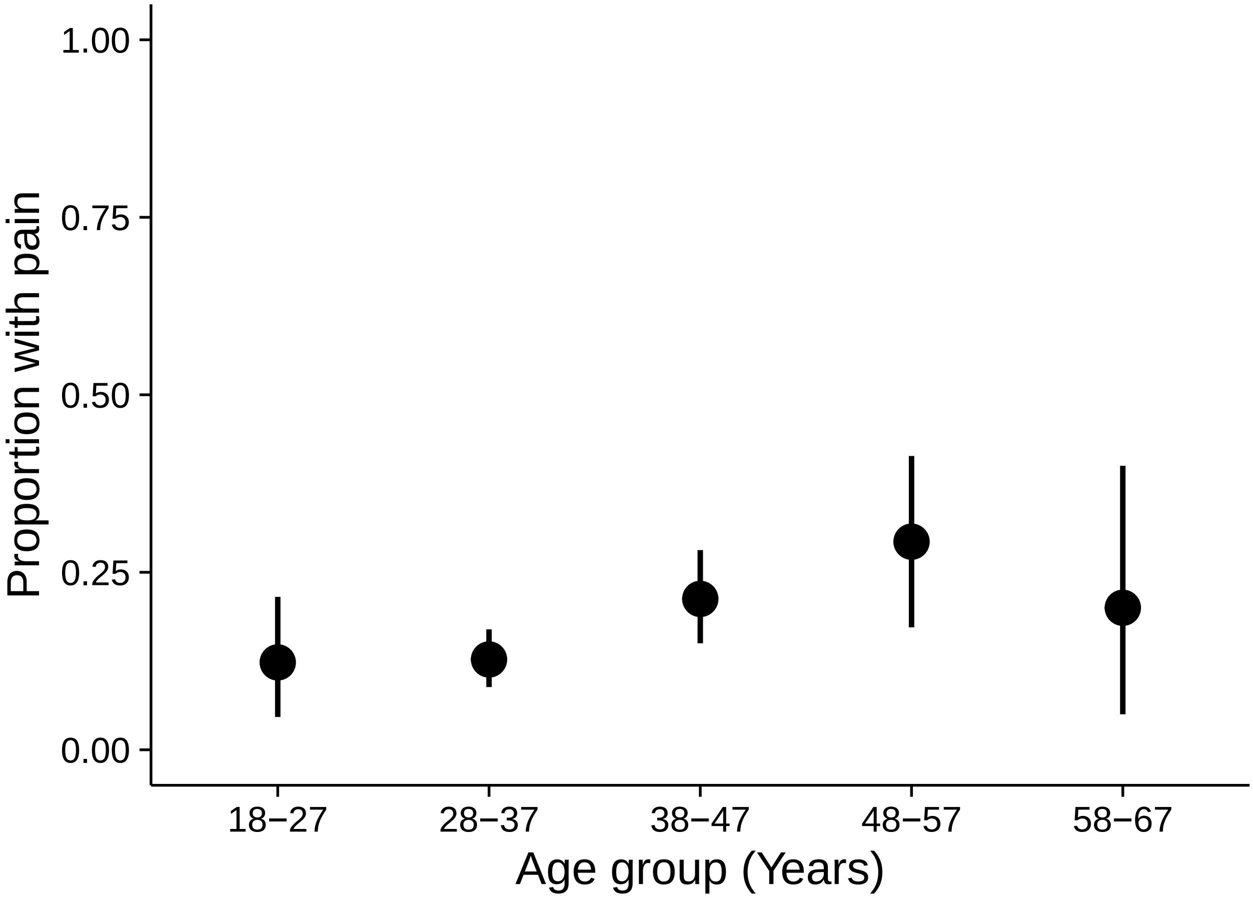

Groin
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
Age group 68−77 years removed because n = 2



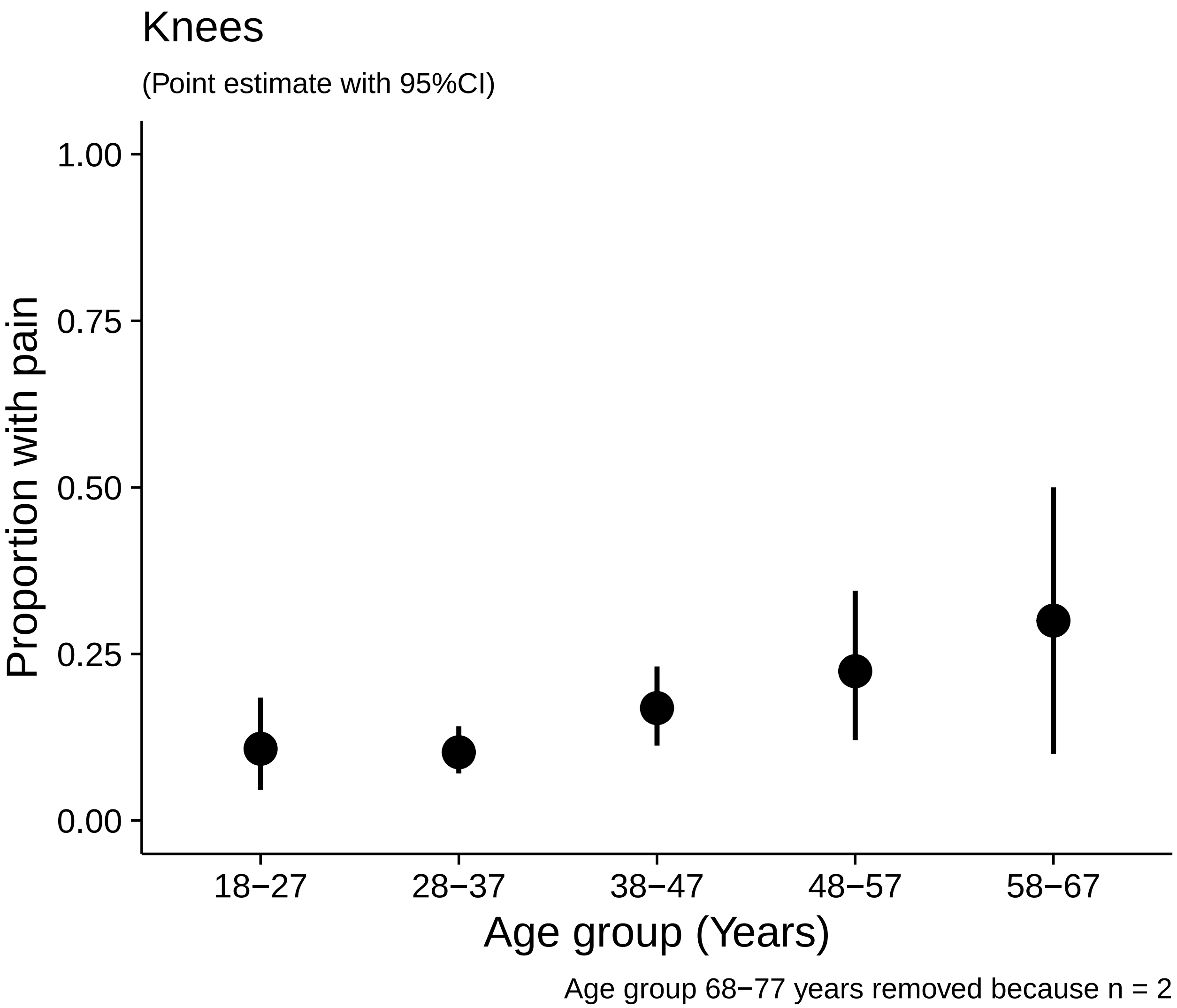
Age group 68−77 years removed because n = 2

Legs

(Point estimate with 95%CI)



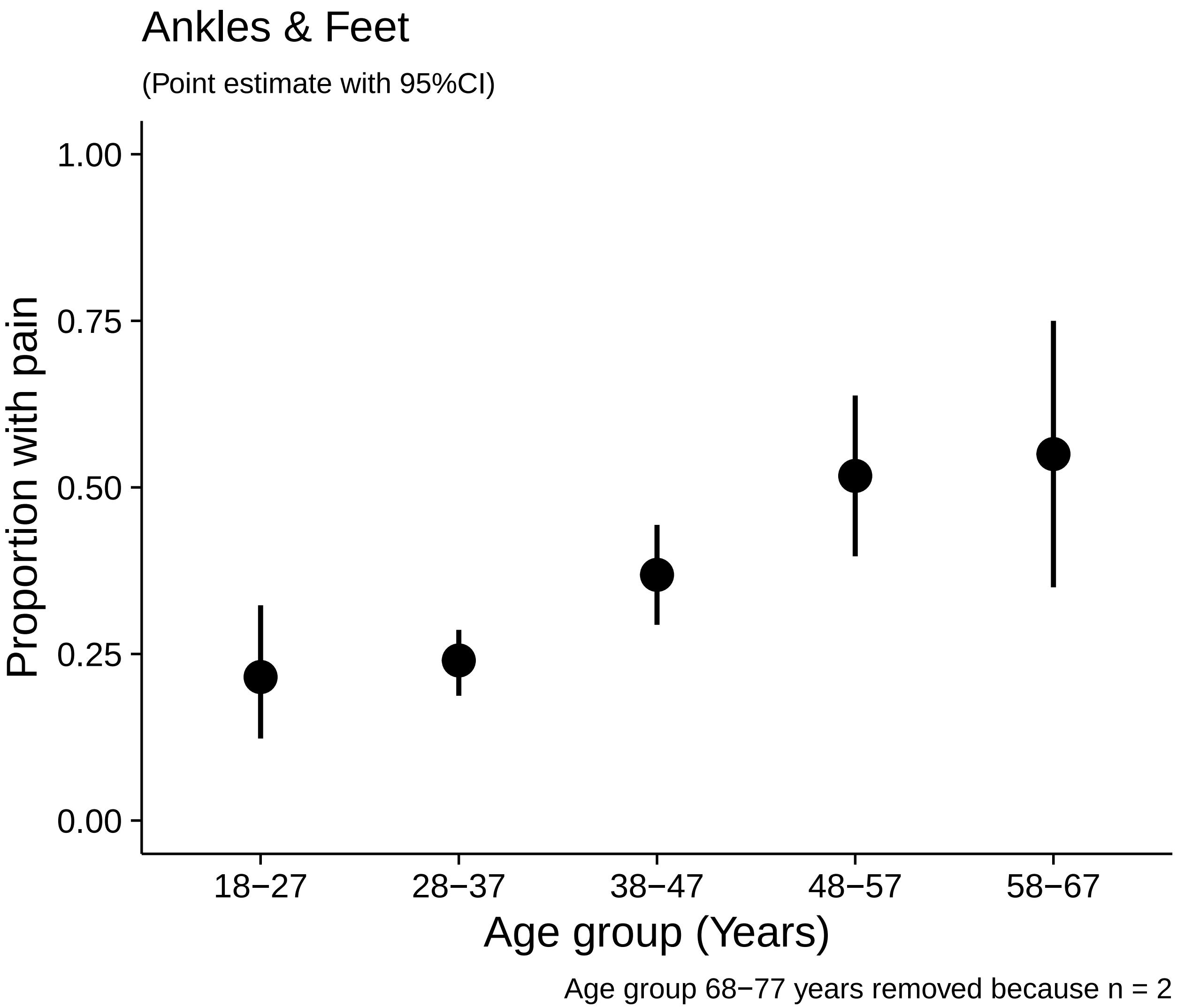
Age group 68−77 years removed because n = 2

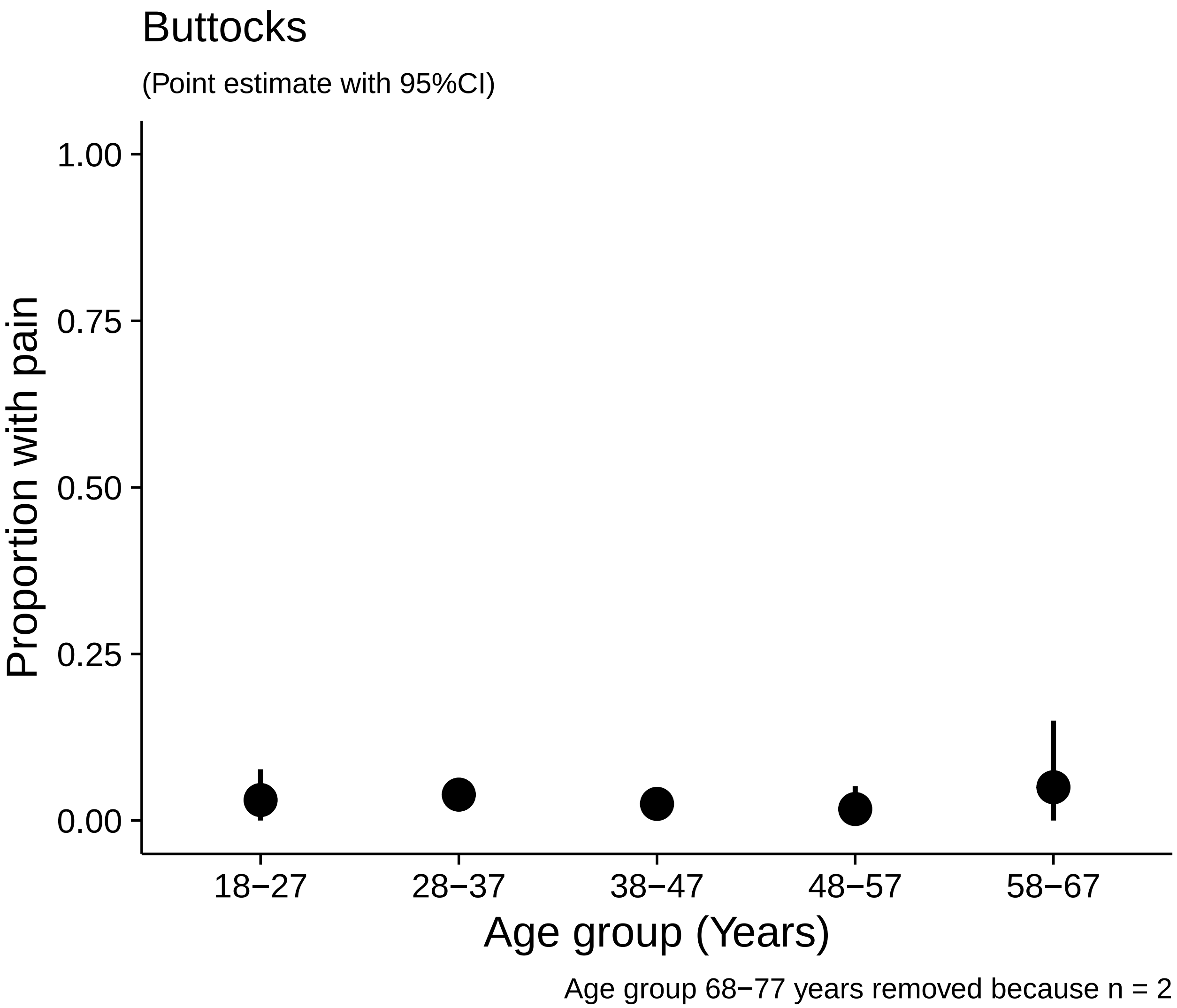


Knees

(Point estimate with 95%CI)

Age group 68−77 years removed because n = 2


Ankles & Feet
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
18−27
28−37
38−47
48−57
58−67
Age group (Years)
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 have 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')]
```

```r
# Join to boot_data & remove ID, study site, and upper_back
# (only one outcome -- no pain)
cd4 <- left_join(data, cd4) %>%
    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 36: 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,
                             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)) +
                           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 37: 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 38: 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 39: 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

cd4_group	point_est	lower_ci	upper_ci
500+	0.08	0.02	0.13

Table 40: 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 41: 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 42: 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 43: 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 44: 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
500+	0.10	0.04	0.16

Table 45: 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 46: 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 47: 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 48: 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

cd4_group	point_est	lower_ci	upper_ci
300-399	0.18	0.10	0.29
400-499	0.30	0.17	0.43
500+	0.21	0.13	0.28

Table 49: 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 50: 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 51: 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 52: 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 53: Ankles & Feet

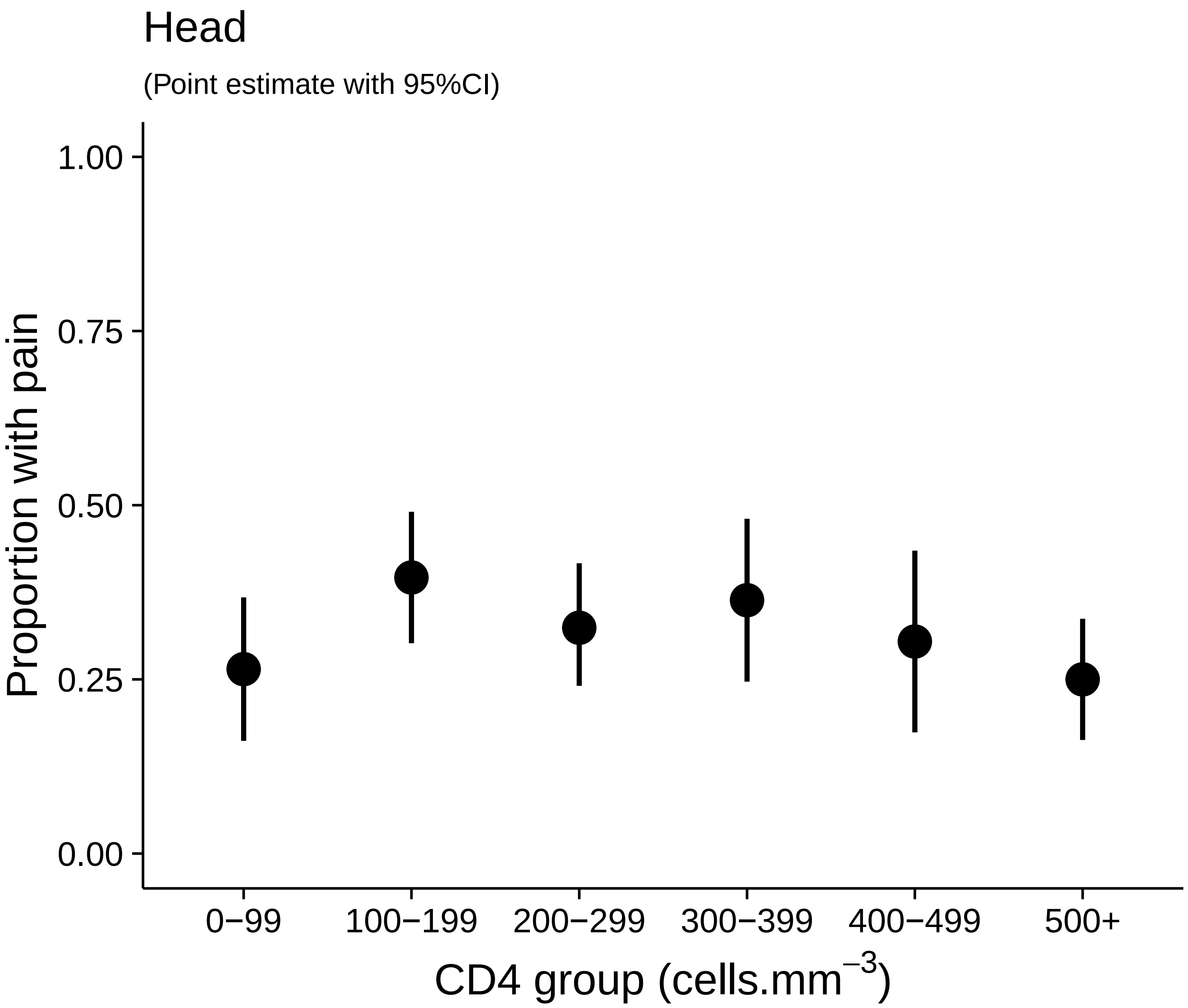
cd4_group	point_est	lower_ci	upper_ci
0-99	0.26	0.16	0.37
100-199	0.30	0.22	0.40
200-299	0.31	0.23	0.41
300-399	0.25	0.16	0.34
400-499	0.26	0.15	0.39
500+	0.35	0.25	0.45

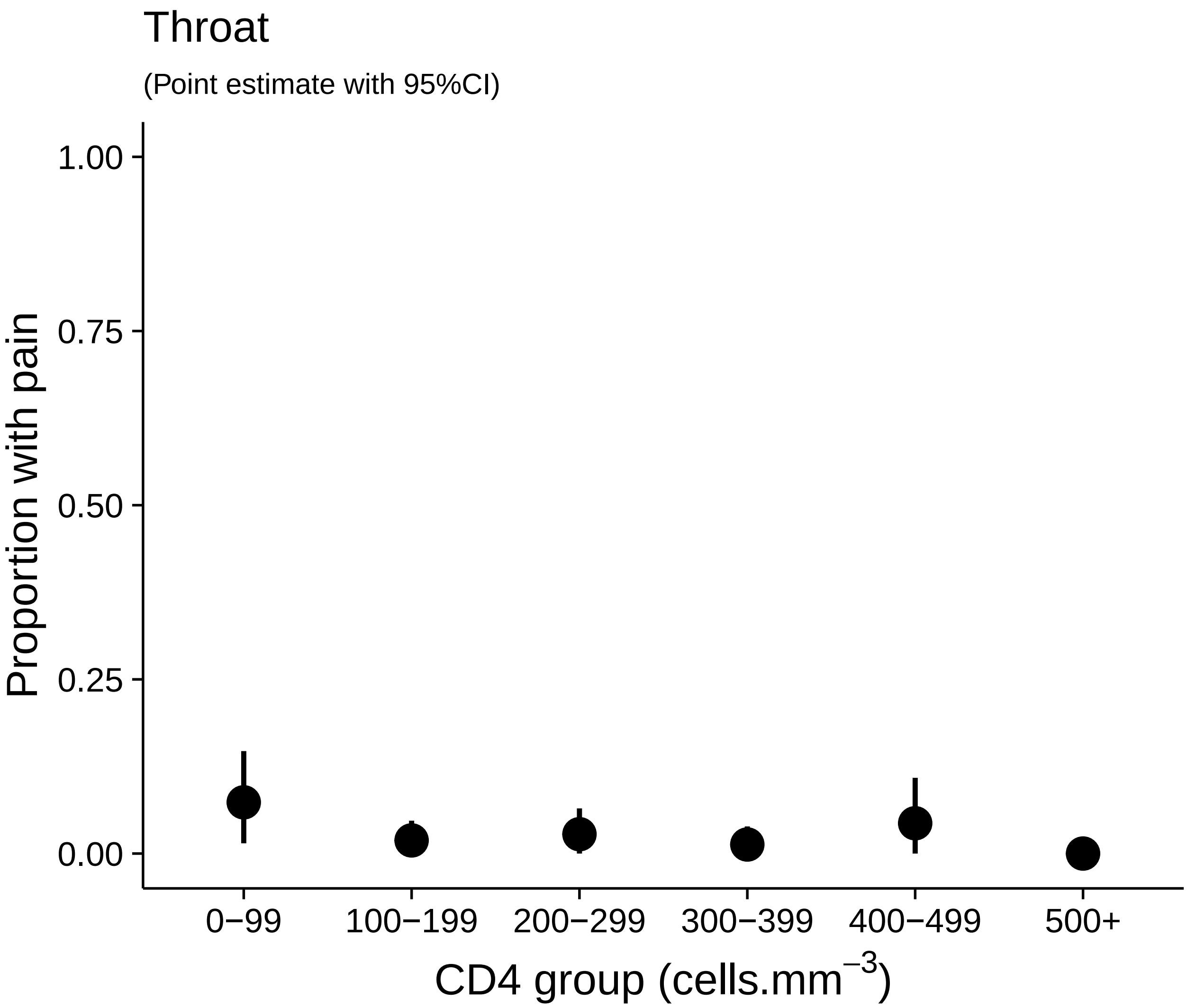
Table 54: Buttocks

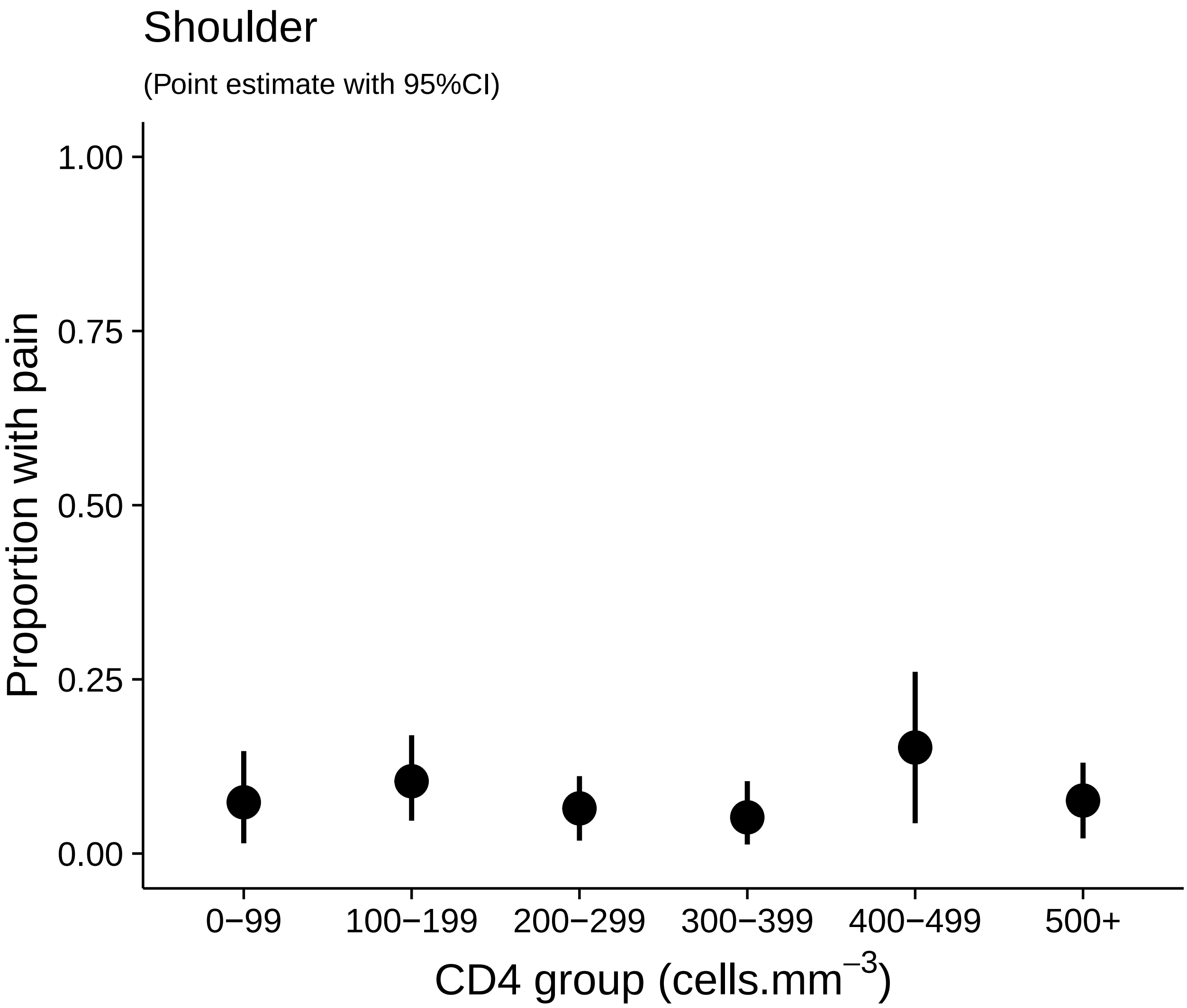
cd4_group	point_est	lower_ci	upper_ci
0-99	0.04	0.00	0.10
100-199	0.01	0.00	0.03
200-299	0.03	0.00	0.06
300-399	0.05	0.01	0.10
400-499	0.04	0.00	0.11
500+	0.07	0.02	0.12

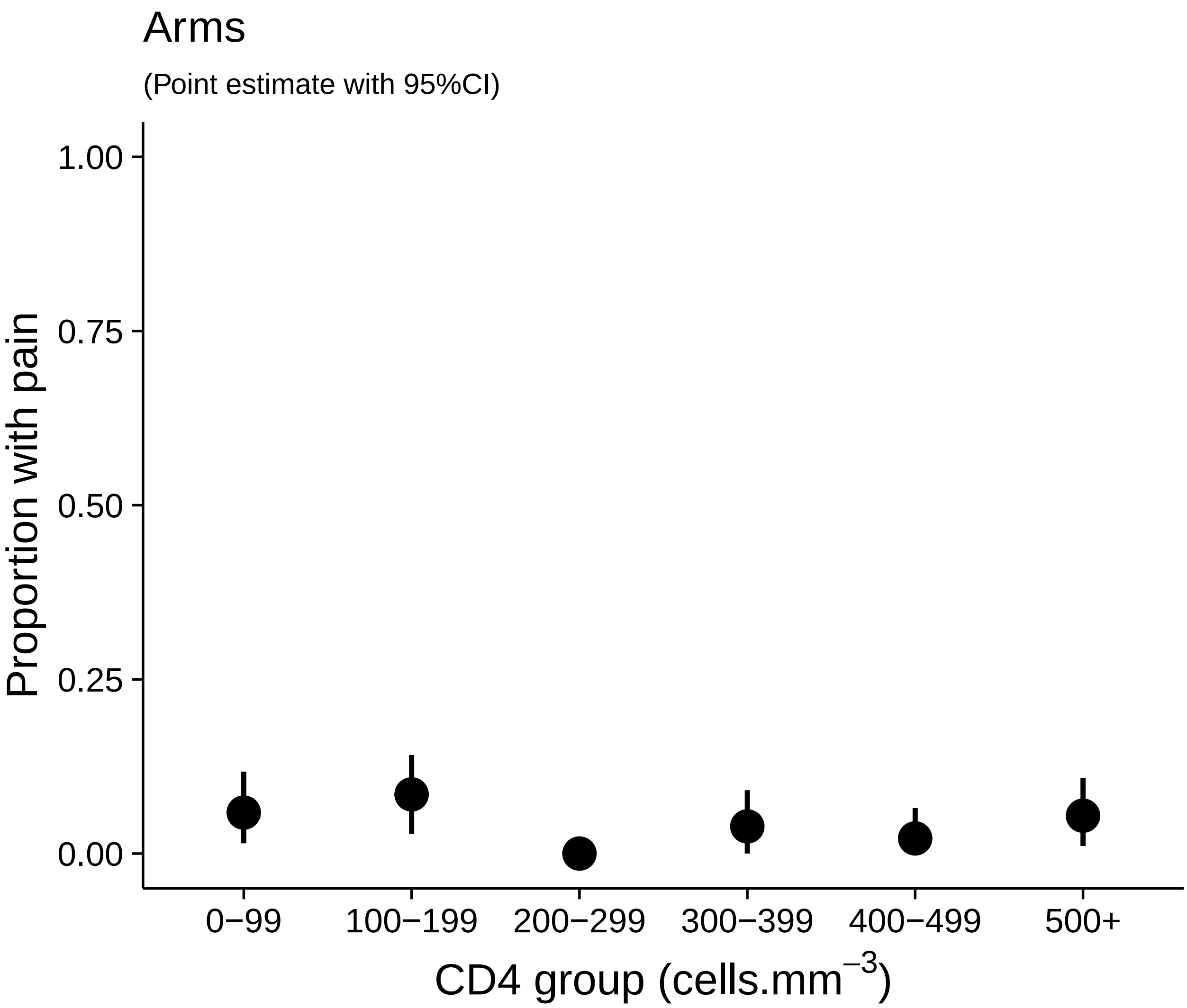
6.3 Plotted proportions (with 95% CIs), by CD4 (recent) group and body site

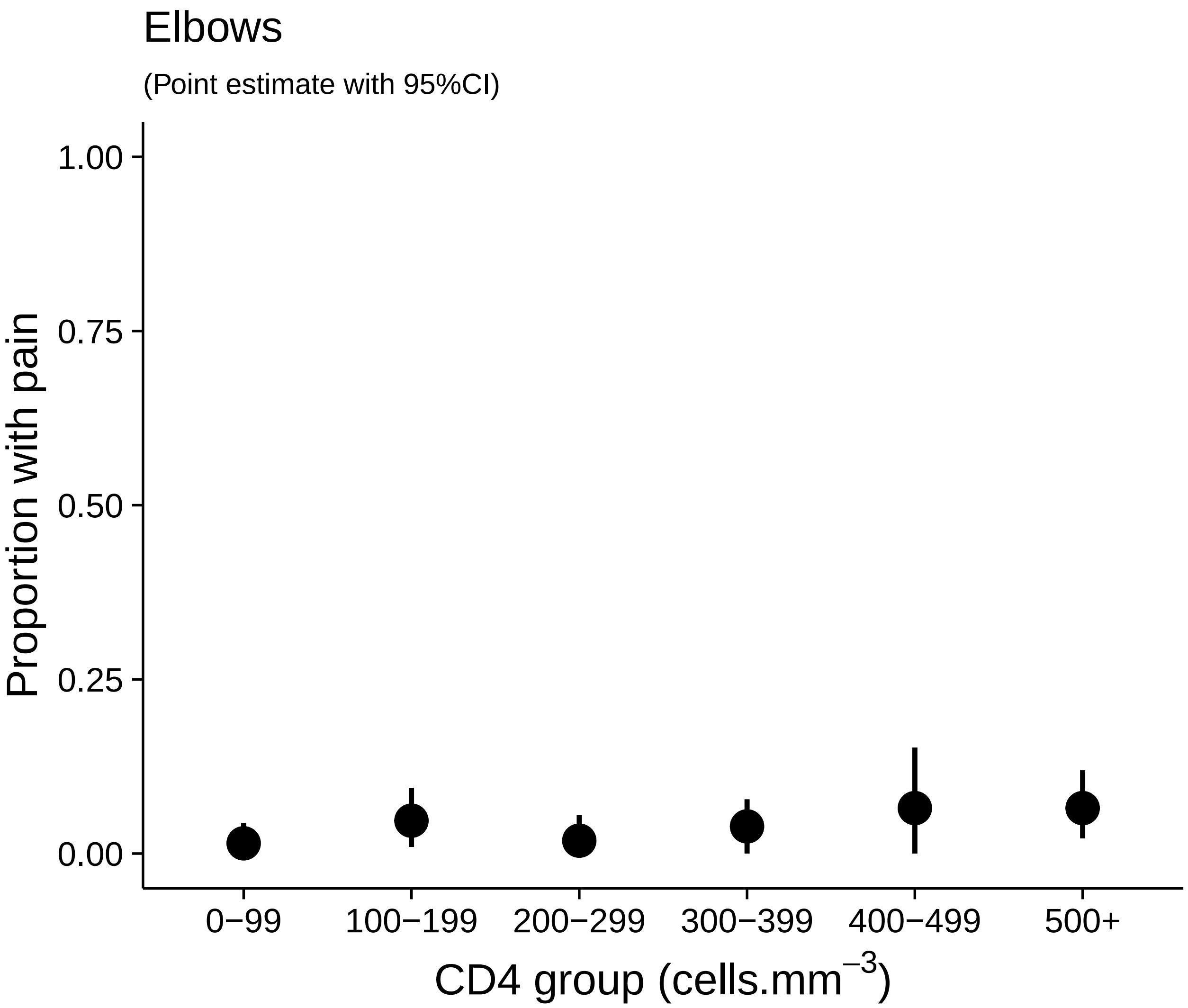
```
walk(cd4_boot2$plots, ~ print(.x))
```

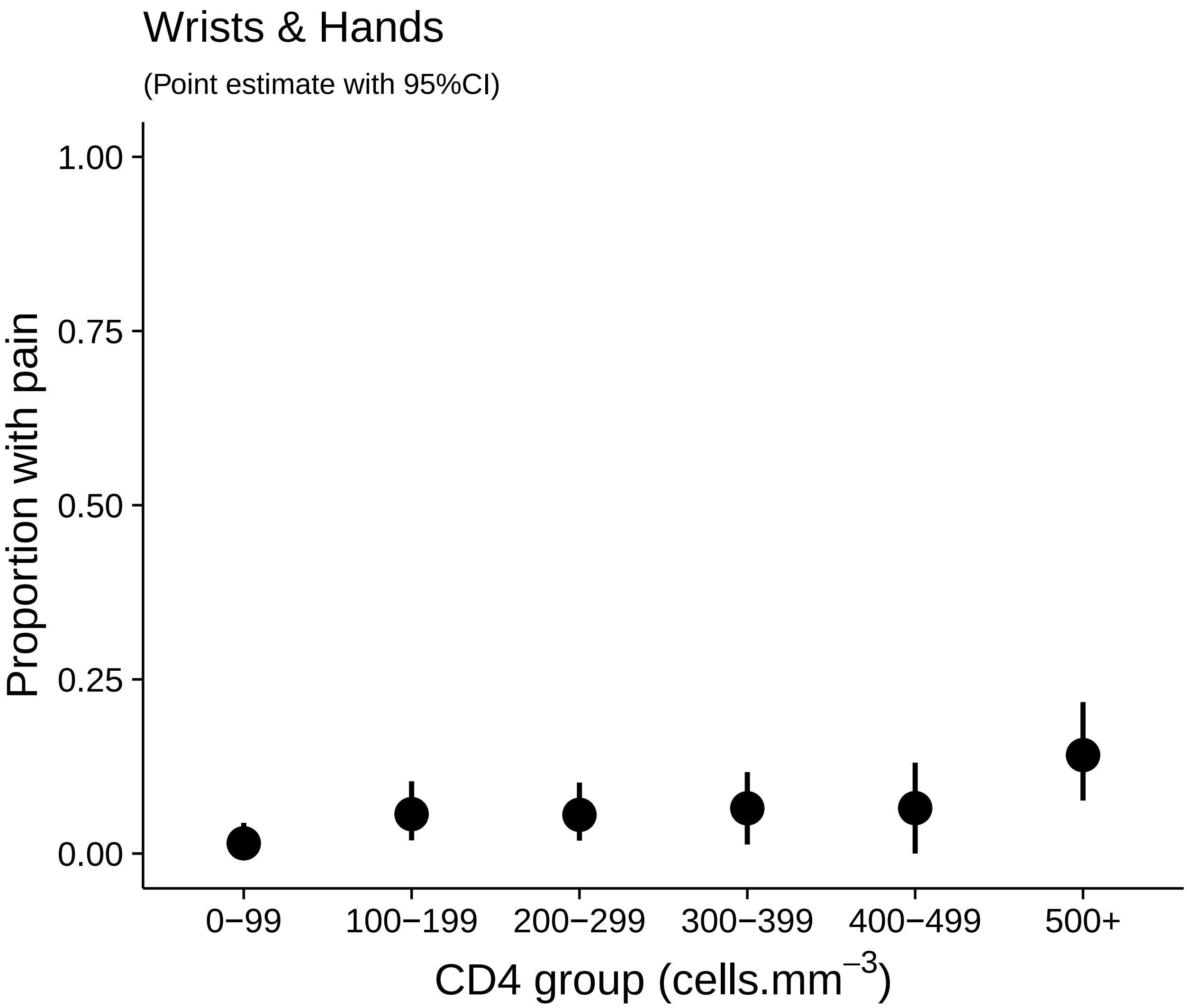

Head
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0–99
100–199
200–299
300–399
400–499
500+
CD4 group (cells.mm^{-3})

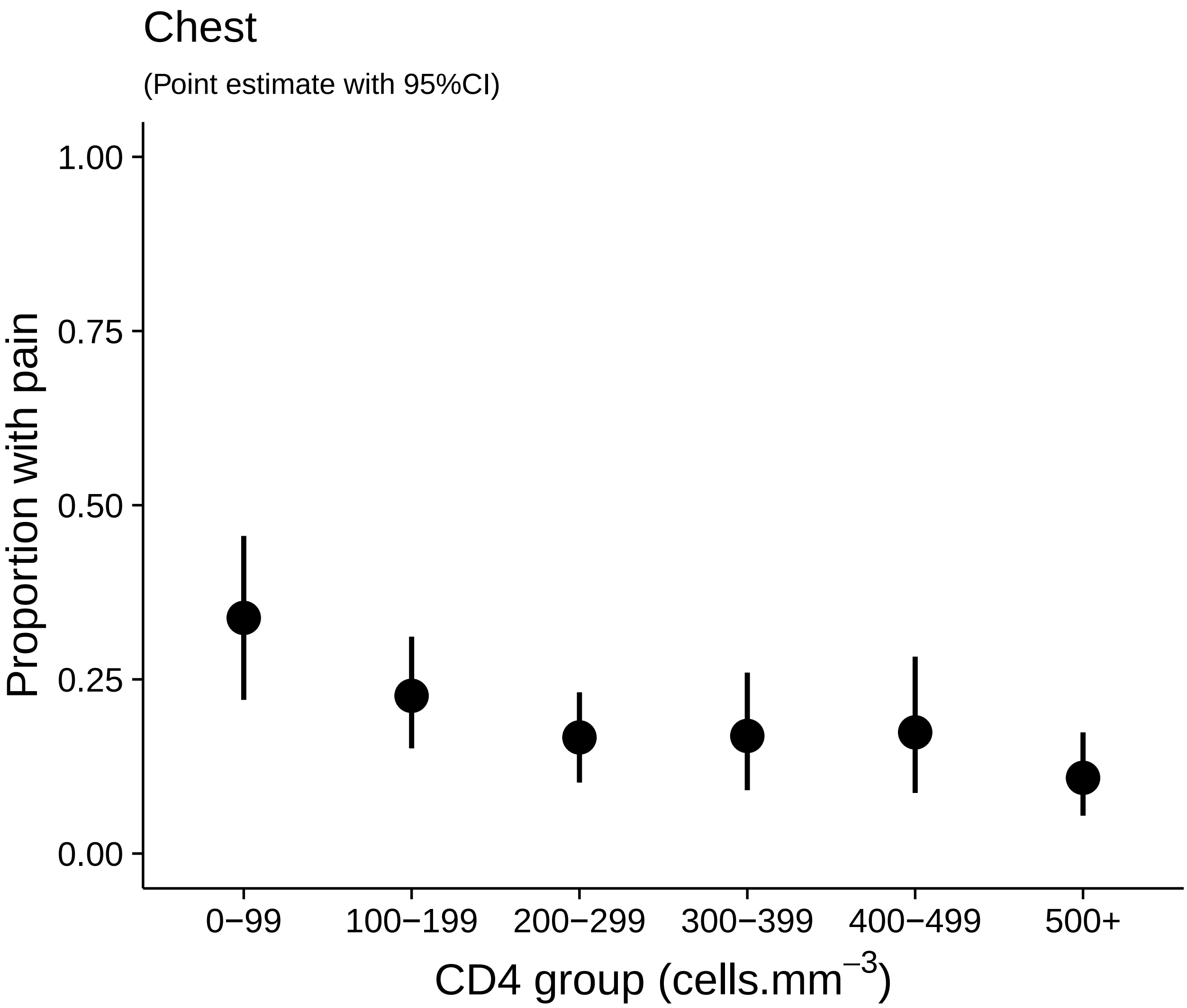

Throat
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

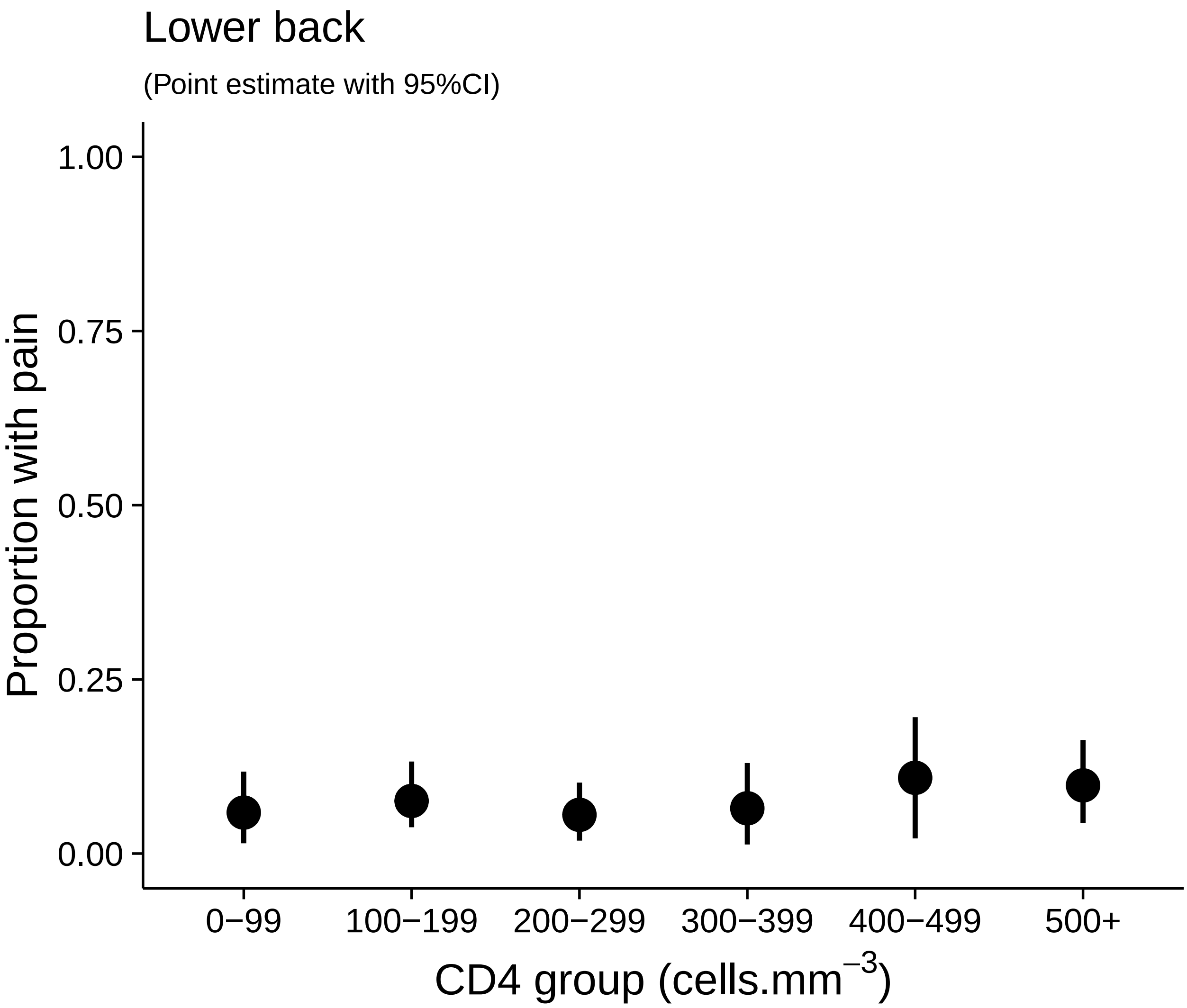

Shoulder
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

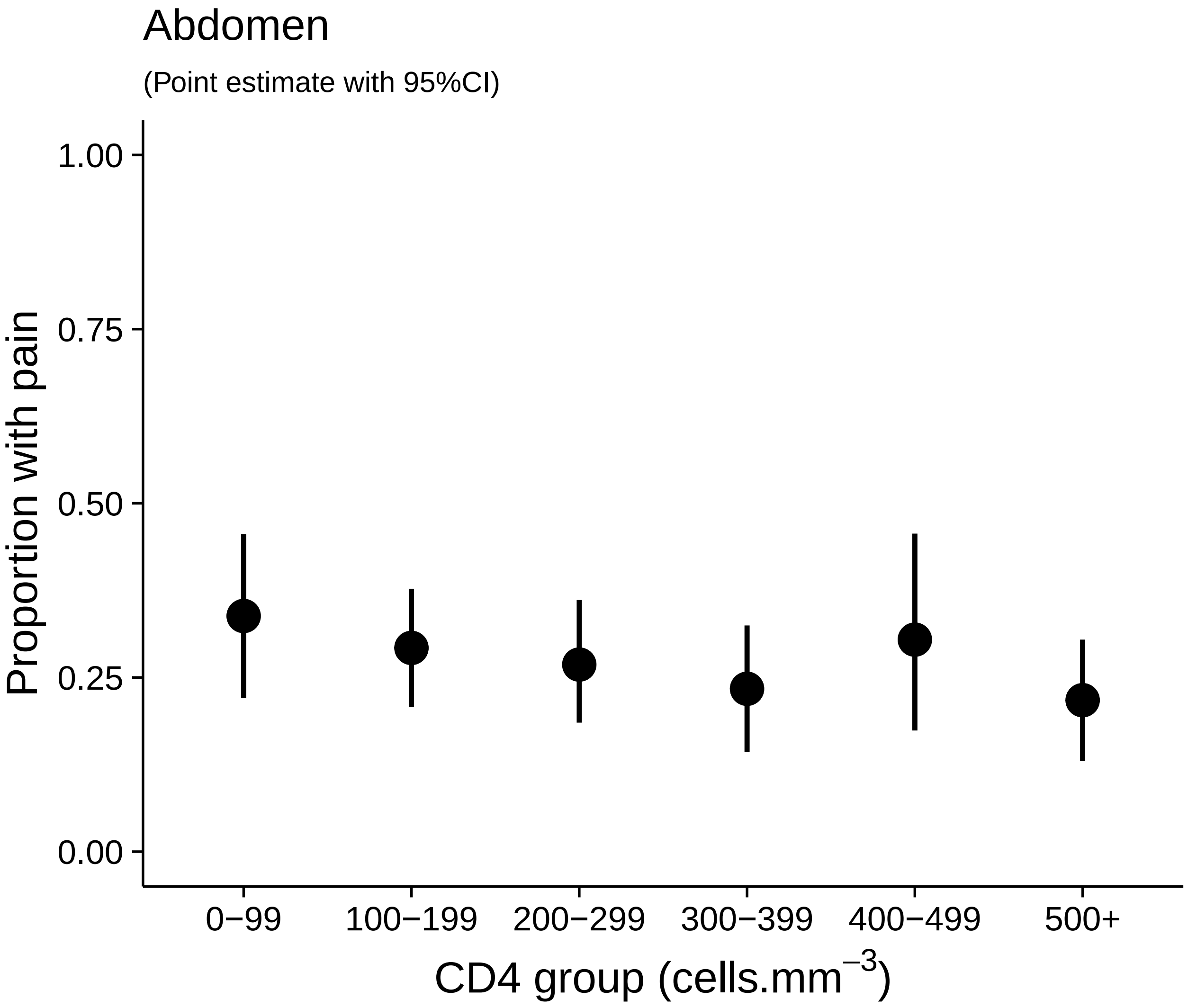

Arms
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})


Elbows
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})


Wrists & Hands
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm⁻³)

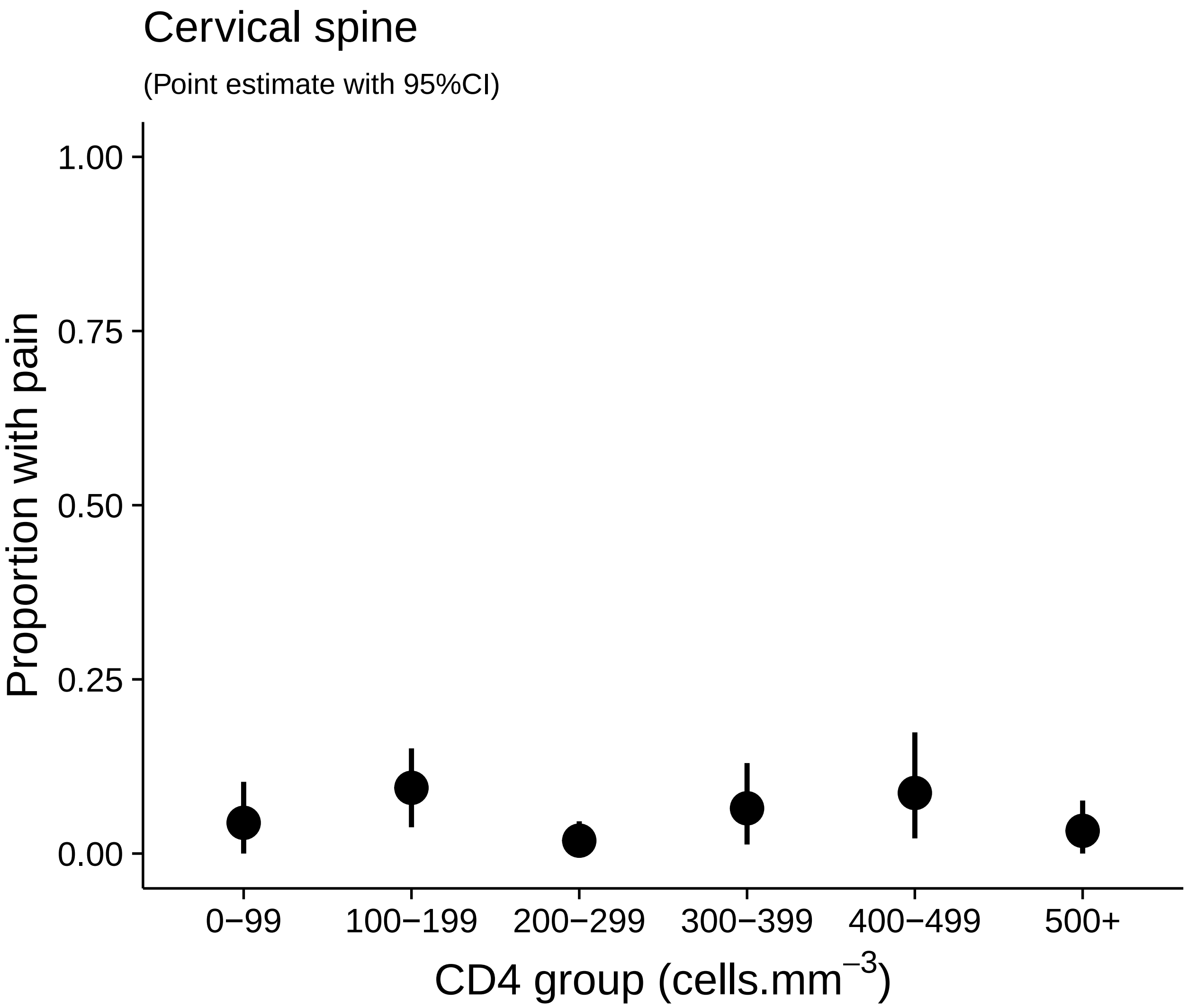

Chest
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm–3)

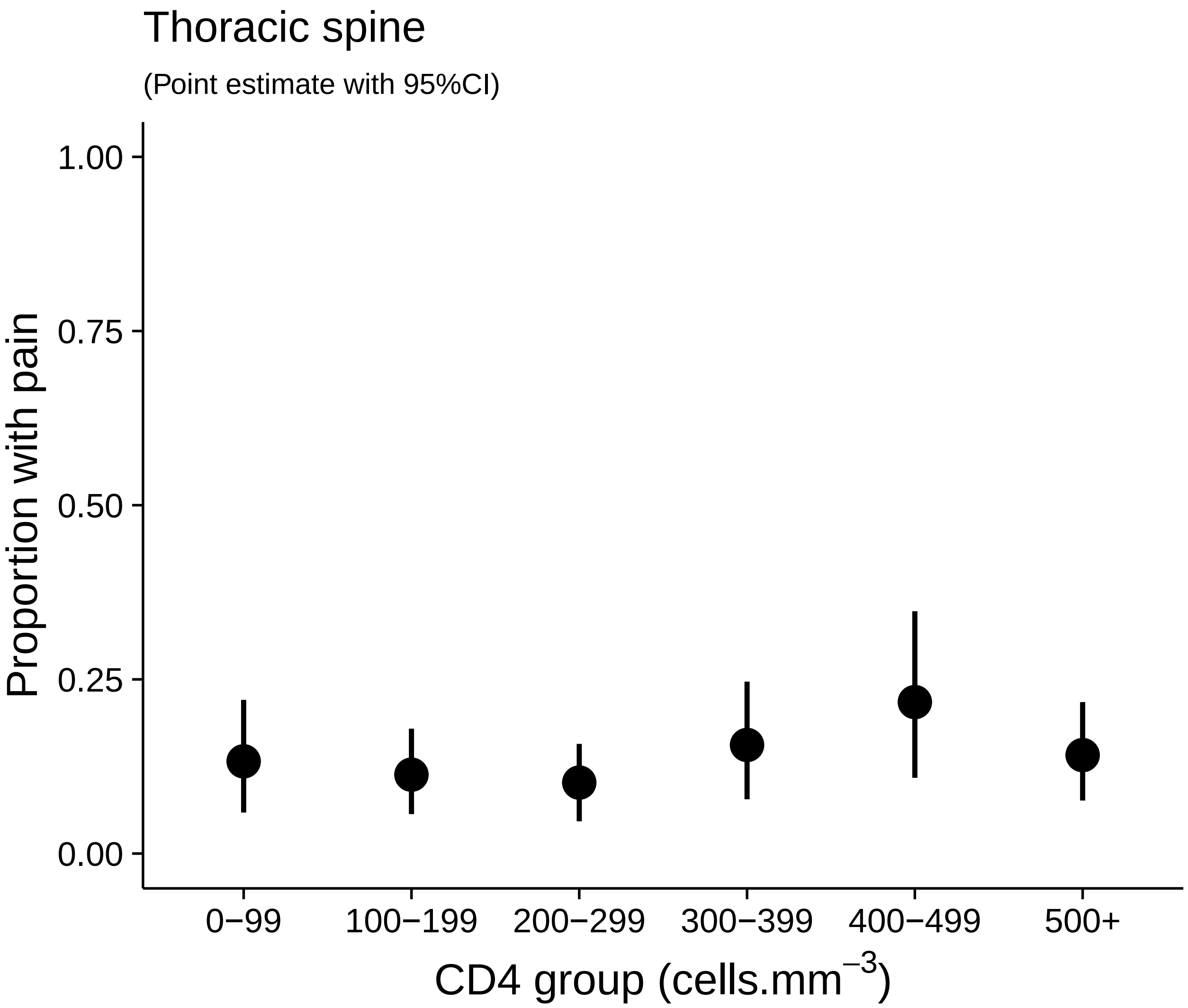

Lower back
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

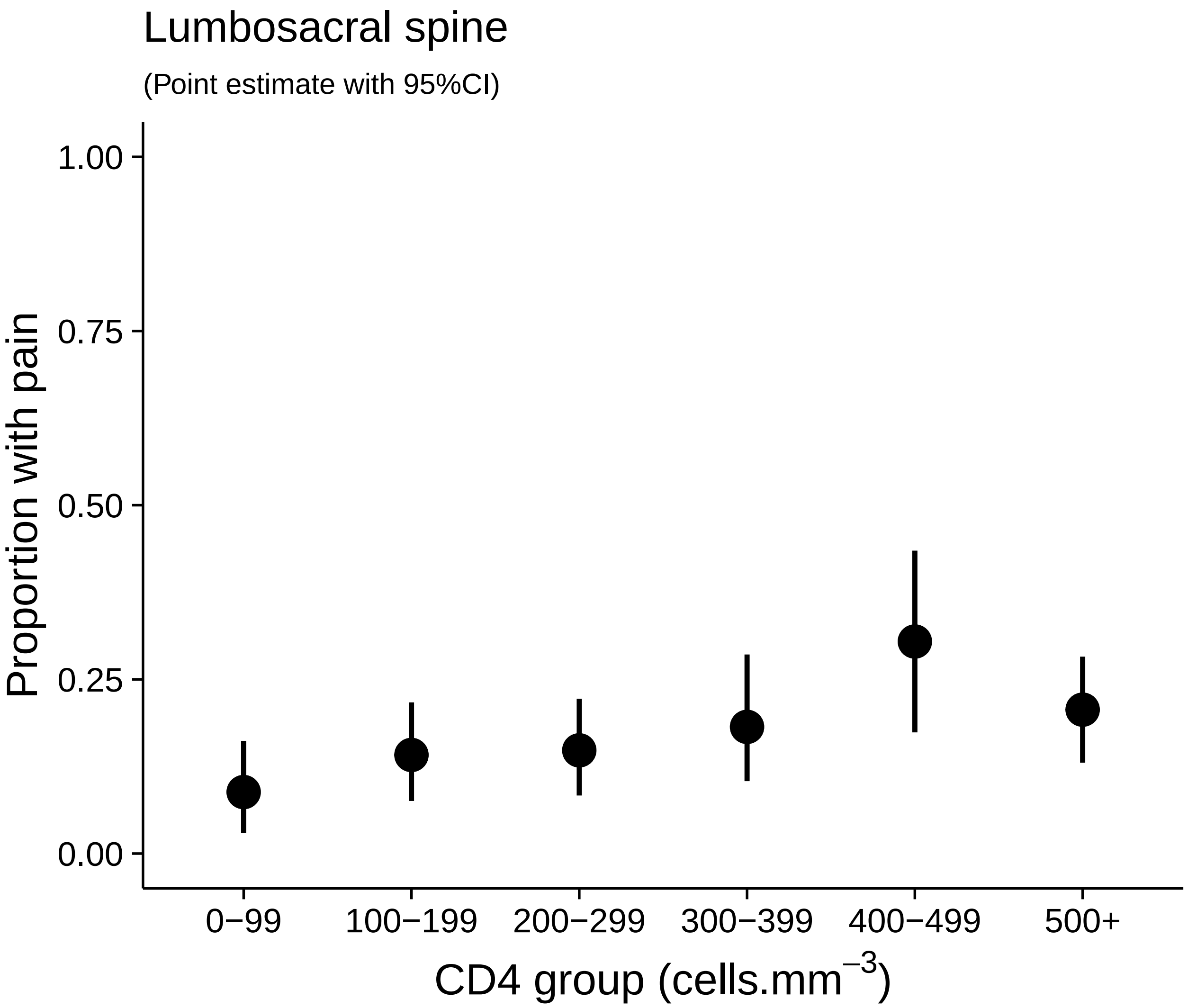


Abdomen

(Point estimate with 95%CI)

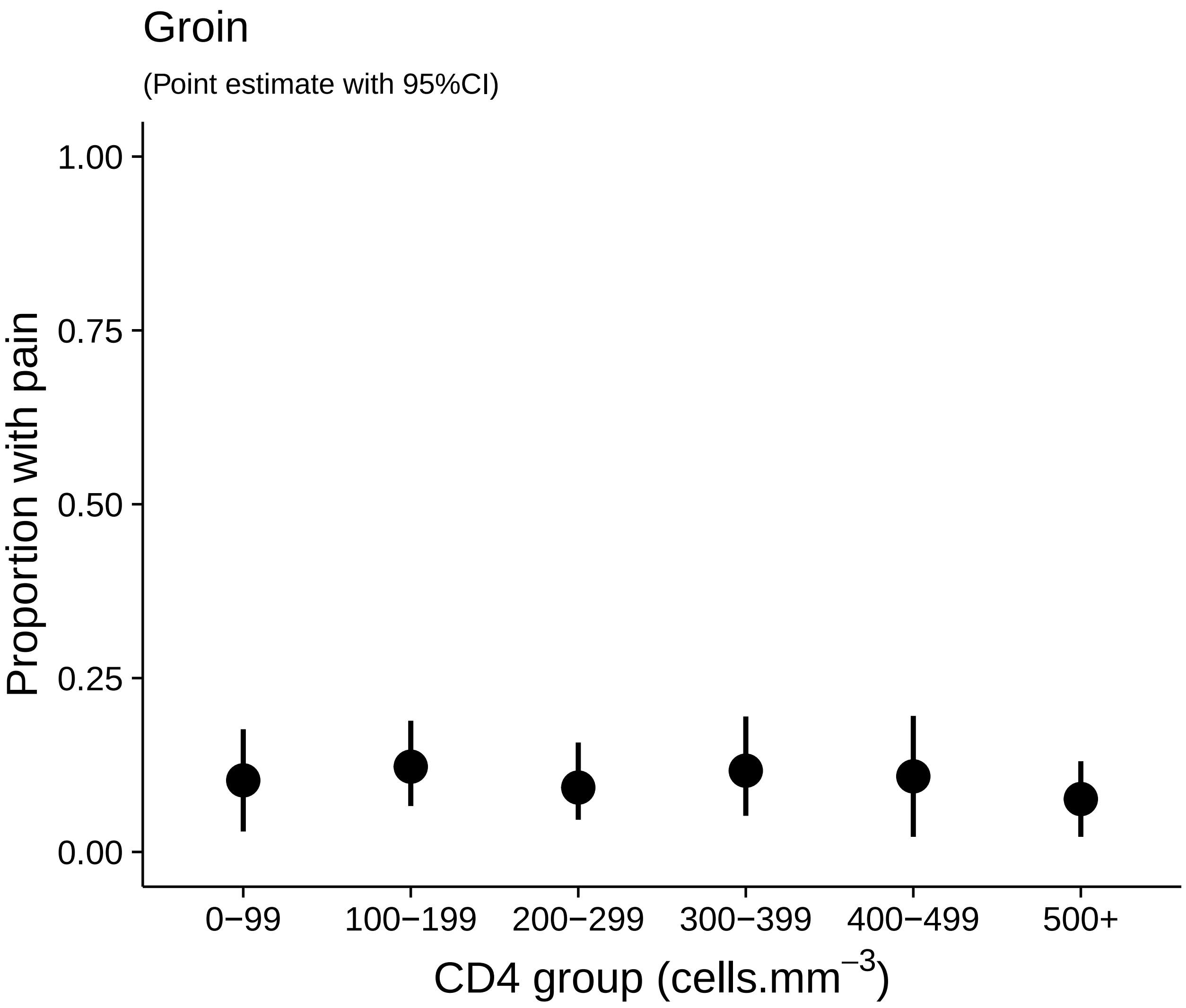

Cervical spine
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

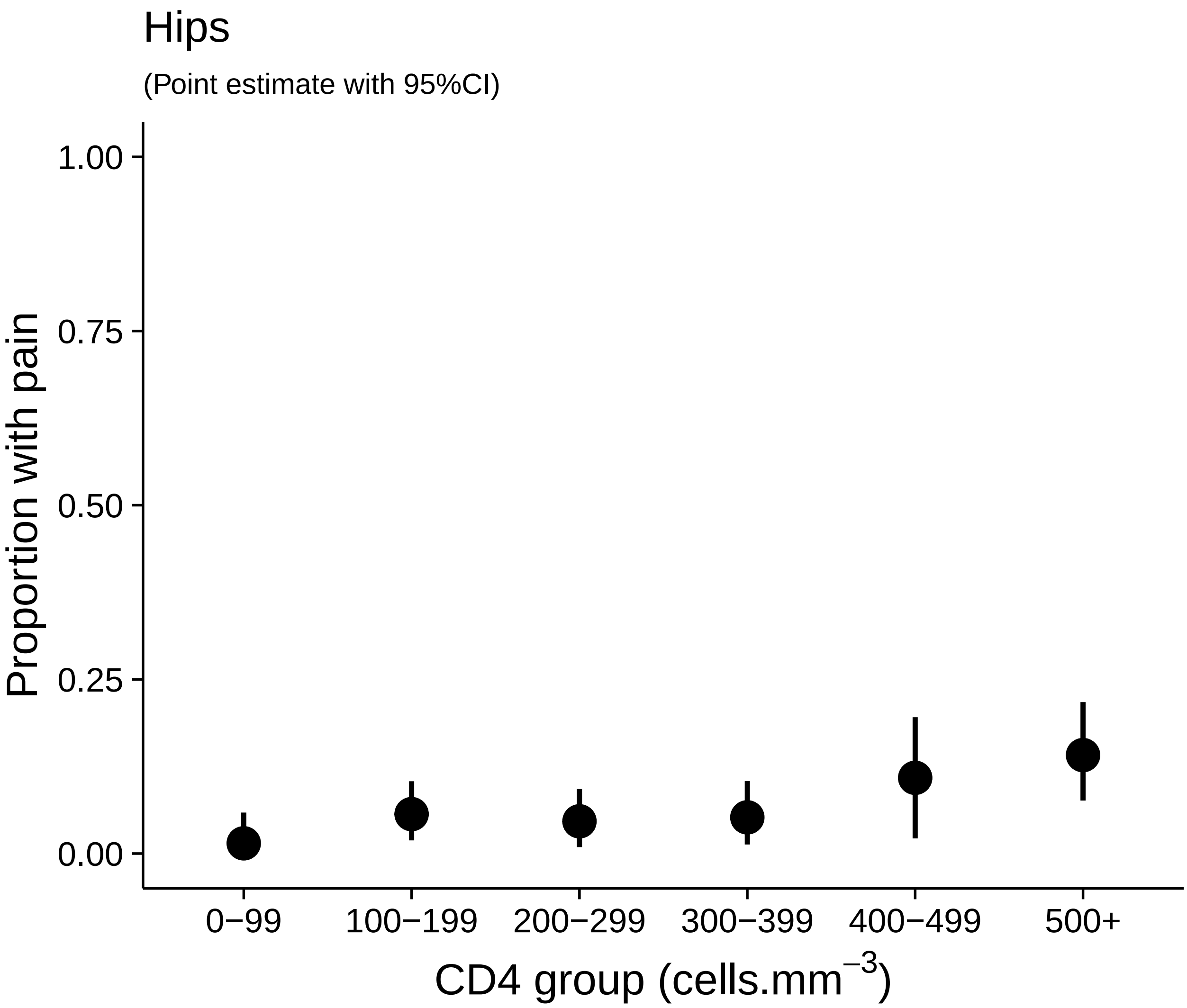

Thoracic spine
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

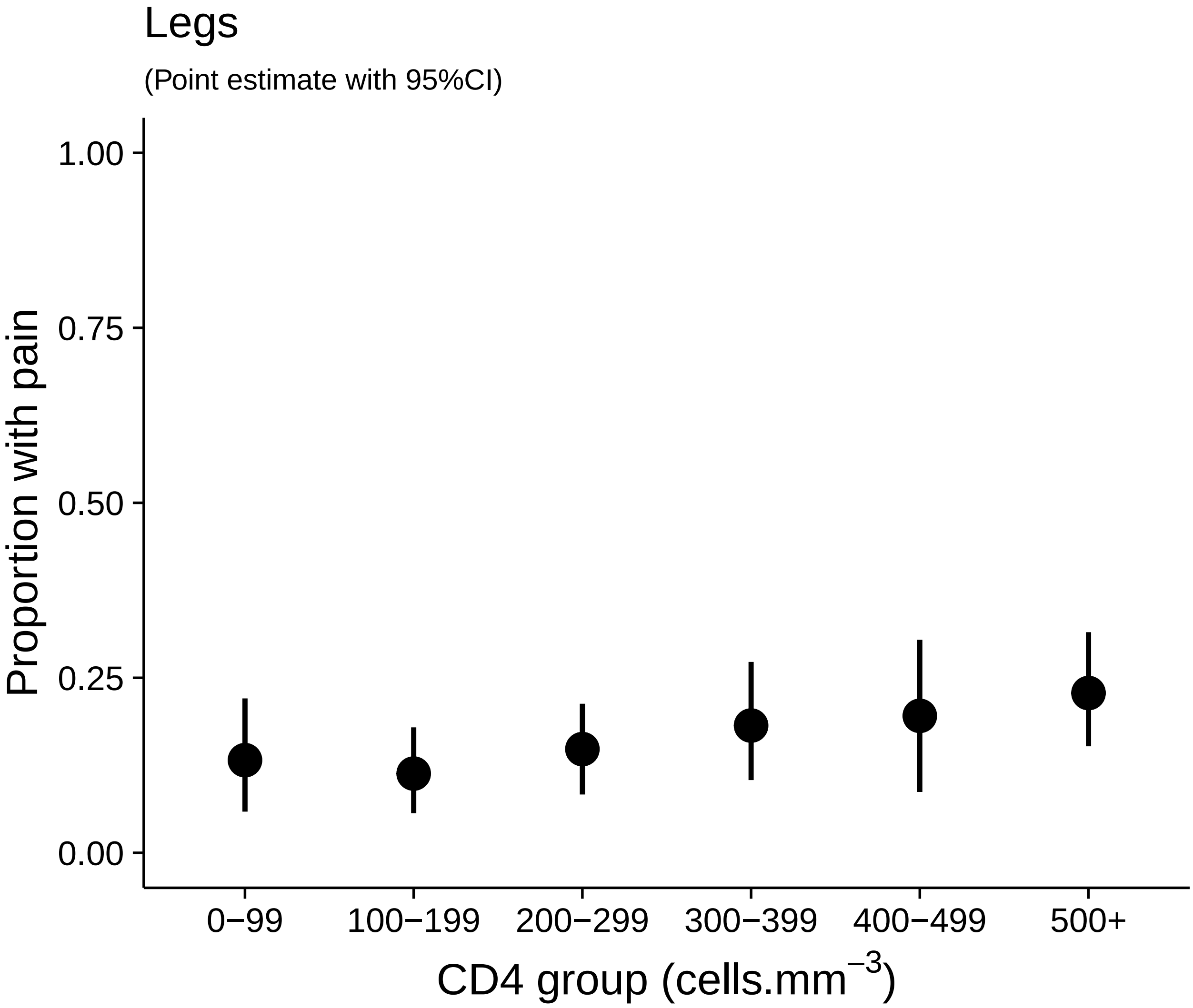


Lumbosacral spine

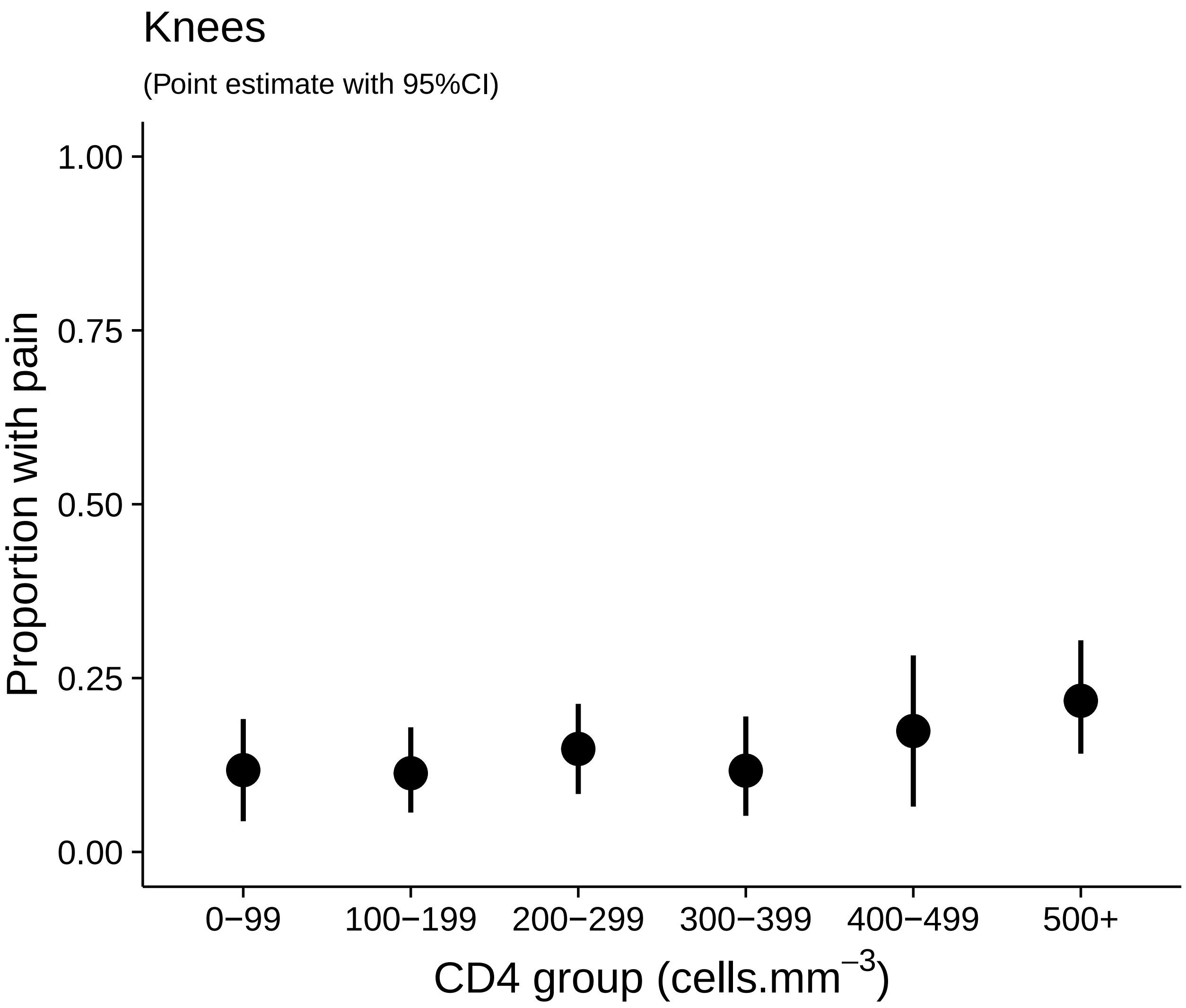
(Point estimate with 95%CI)


Groin
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})


Hips
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})

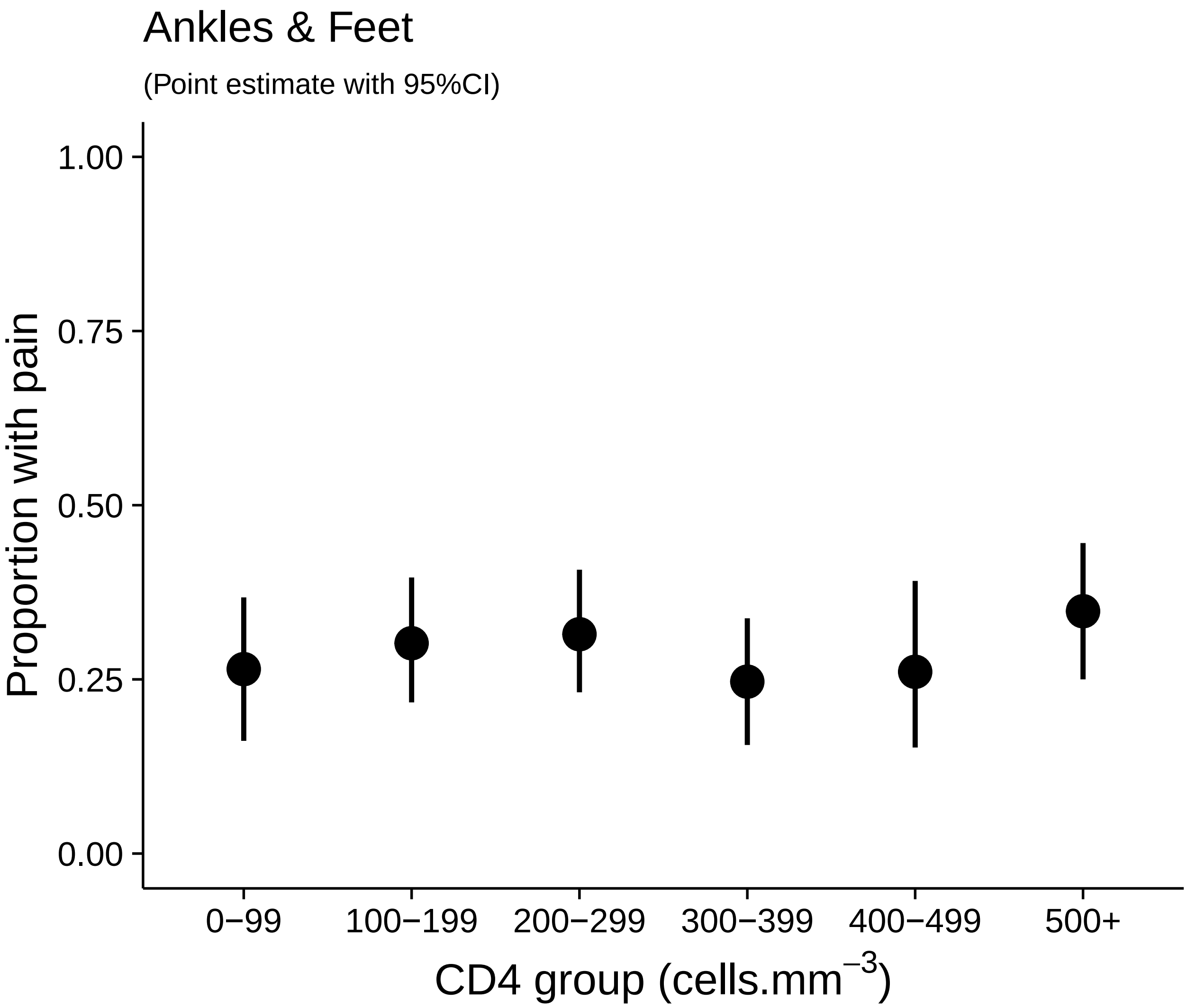

Legs

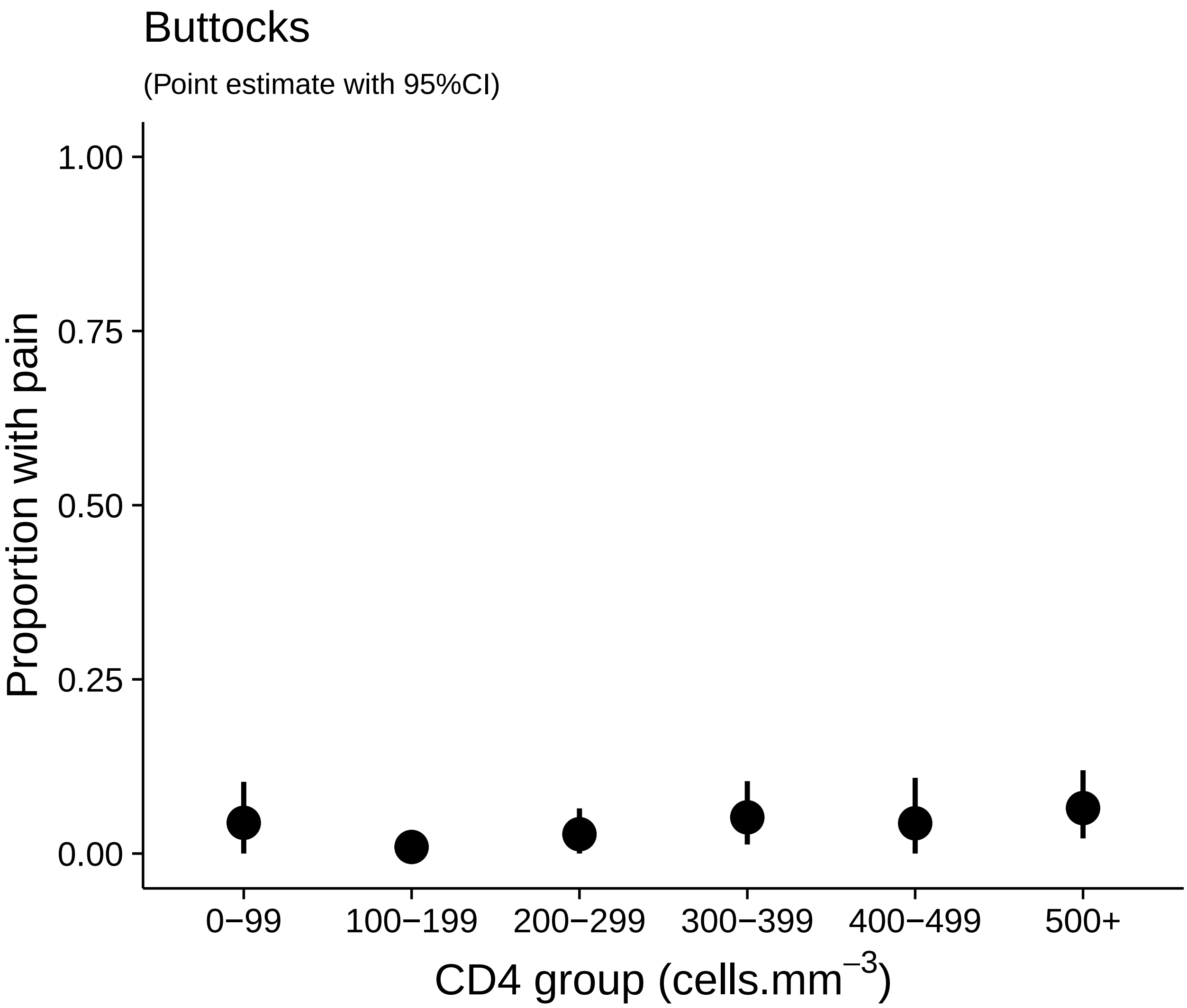
(Point estimate with 95%CI)



Knees

(Point estimate with 95%CI)


Ankles & Feet
(Point estimate with 95%CI)
Proportion with pain
1.00
0.75
0.50
0.25
0.00
0−99
100−199
200−299
300−399
400−499
500+
CD4 group (cells.mm^{-3})



7 Logistic regression

Note: Age and CD4 T-cell count treated as a continuous variable, not like the categories used in the figures shown above.

7.1 Process data

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

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

# Process data
data_log <- data %>%
    # Join with extracted demographics data (demo_reduced)
    left_join(demo_log) %>%
    # Remove upper back (only one outcome -- no pain) and ID
    select(-ID, -Upper_back)

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

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

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

7.2 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 Bonferroni correction.

7.3 Run `glmer` model for every body site to check for issues

P-values reported for the likelihood ratio tests below are uncorrected for multiple comparisons.

```
#-- Head --#
head <- data_log.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 +
                  (1|Site),
              data = head,
              family = binomial())
```

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

```
## Data: head
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## head_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 599.13 607.53 -297.56   595.13
## head_mod  5 589.23 610.23 -289.61   579.23 15.904      3   0.001186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Significant, therefore, print summary
## (family-wise error correction, alpha = 0.0024)

summary(head_mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: head
##
##      AIC      BIC   logLik deviance df.resid
##    589.2    610.2   -289.6    579.2      488
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0009 -0.7400 -0.4814  1.0622  2.7302
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Site   (Intercept) 0.2072   0.4552
## Number of obs: 493, groups:  Site, 5
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.65724    0.24466  -2.686  0.00722 **
## Age          0.01316    0.12246   0.107  0.91441
## SexMale     -1.34715    0.38454  -3.503  0.00046 ***
## CD4_recent  -0.13676    0.11343  -1.206  0.22796
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Age    SexMal
## Age        -0.017
## SexMale    -0.166 -0.097
## CD4_recent -0.097 -0.085  0.096
```

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

```r
    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 +
                  (1|Site),
              data = throat,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: throat
##      AIC      BIC   logLik deviance df.resid
## 123.0598 144.0624 -56.5299 113.0598      488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0.00000008333
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##    -4.06946      0.09379      0.95595     -0.75322
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```r
#-- Shoulder --#
shoulder <- data_log.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 +
                  (1|Site),
              data = shoulder,
              family = binomial())
```

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

```
## Data: shoulder
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## shdr_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 286.42 294.82 -141.21   282.42
## shdr_mod  5 290.30 311.30 -140.15   280.30 2.1205      3     0.5478
```

```
## No significant difference between null and full model

#-- Arm --#
arms <- data_log.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 +
                  (1|Site),
              data = arms,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: arms
##       AIC      BIC   logLik deviance df.resid
## 188.3179 209.3205 -89.1590 178.3179      488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0.00000001979
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -3.0519       0.2332      -0.1355      -0.1925
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
#-- Elbow --#
elbow <- data_log.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 +
                  (1|Site),
              data = elbow,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: elbow
##       AIC      BIC   logLik deviance df.resid
## 170.8357 191.8382 -80.4178 160.8357      488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -3.2355       0.4798      -0.1173       0.1120
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
#-- Wrist & hand --#
hand <- data_log.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 +
                  (1|Site),
              data = hand,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: hand
##       AIC       BIC    logLik  deviance  df.resid
##  242.3033  263.3059 -116.1517  232.3033       488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -2.6311       0.2989      -0.6029       0.3412
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
#-- Chest --#
chest <- data_log.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 +
                  (1|Site),
              data = chest,
              family = binomial())

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

```
## Data: chest
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## chst_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 479.88 488.28 -237.94   475.88
## chst_mod  5 481.62 502.62 -235.81   471.62 4.2655      3     0.2342
```

```
## No significant difference in null and full models

#-- Abdomen --#
```

```r
abdomen <- data_log.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 +
                  (1|Site),
              data = abdomen,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: abdomen
##       AIC       BIC    logLik  deviance  df.resid
##  580.0448  601.0474 -285.0224  570.0448       488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -0.9525      -0.2005      -0.3496      -0.1503
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```r
#-- Lower back (flank) --#
low_back <- data_log.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 +
                  (1|Site),
              data = low_back,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: low_back
##       AIC       BIC    logLik  deviance  df.resid
##  270.8031  291.8056 -130.4015  260.8031       488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -2.4898       0.1578      -0.1832       0.1366
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
#-- Groin --#
groin <- data_log.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 +
                   (1|Site),
               data = groin,
               family = binomial())

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

```
## Data: groin
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## groin_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##           Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod   2 304.54 312.94 -150.27   300.54
## groin_mod  5 306.38 327.38 -148.19   296.38 4.1552      3     0.2452
## No significant difference between null and full model

#-- Buttocks --#
buttocks <- data_log.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 +
                  (1|Site),
              data = buttocks,
              family = binomial())

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##    Data: buttocks
##       AIC       BIC    logLik deviance df.resid
## 168.7539 189.7564 -79.3769 158.7539      488
## Random effects:
##  Groups Name        Std.Dev.
##  Site   (Intercept) 0
## Number of obs: 493, groups:  Site, 5
## Fixed Effects:
## (Intercept)          Age      SexMale   CD4_recent
##     -3.3984      -0.1151       0.6705       0.2500
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
#-- Hips --#
hips <- data_log.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 +
                  (1|Site),
              data = hips,
              family = binomial())

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

```
## Data: hips
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## hips_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 224.25 232.65 -110.12   220.25
## hips_mod  5 228.57 249.57 -109.29   218.57 1.6732      3     0.6429
```

```
## No significant difference between null and full model

#-- Legs --#
legs <- data_log.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 +
                  (1|Site),
              data = legs,
              family = binomial())

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

```
## Data: legs
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## legs_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 435.34 443.74 -215.67   431.34
## legs_mod  5 432.74 453.74 -211.37   422.74 8.5991      3    0.03512 *
## ---
## 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)
```

```
#-- Knees --#
knees <- data_log.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 +
                  (1|Site),
              data = knees,
              family = binomial())

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

```
## Data: knees
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## knee_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)  
## null_mod  2 406.12 414.53 -201.06   402.12                           
## knee_mod  5 403.61 424.61 -196.81   393.61 8.5143      3     0.0365 *
## ---
## 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)

#-- Ankles.Feet --#
ankles <- data_log.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 +
                  (1|Site),
              data = ankles,
              family = binomial())

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

```
## Data: ankles
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## ankl_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 580.66 589.06 -288.33   576.66
## ankl_mod  5 577.83 598.83 -283.92   567.83 8.8317      3    0.03161 *
## ---
## 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)

#-- Cervical spine --#
cervical <- data_log.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 +
                  (1|Site),
              data = cervical,
              family = binomial())

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

```
## Data: cervical
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## neck_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 205.43 213.83 -100.72   201.43
## neck_mod  5 210.87 231.87 -100.43   200.87 0.5666      3      0.904
```

```
## No significant difference between null and full models

#-- Thoracic spine --#
thoracic <- data_log.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 +
                  (1|Site),
              data = thoracic,
              family = binomial())

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

```
## Data: thoracic
## Models:
## null_mod: Pain_present ~ 1 + (1 | Site)
## thrx_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 392.73 401.13 -194.36   388.73
## thrx_mod  5 395.40 416.40 -192.70   385.40 3.3271      3     0.3439
```

```
## No significant difference between null and full models

# Lumbosacral spine
lumbar <- data_log.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 +
                  (1|Site),
              data = lumbar,
              family = binomial())

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

```
## Data: lumbar
## Models:
```

```
## null_mod: Pain_present ~ 1 + (1 | Site)
## lmbr_mod: Pain_present ~ Age + Sex + CD4_recent + (1 | Site)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_mod  2 451.68 460.08 -223.84   447.68
## lmbr_mod  5 454.14 475.14 -222.07   444.14 3.5429      3     0.3152
```

```
## No significant difference between null and full models
```

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

- Throat
- Arms
- Elbows
- Wrists & hands
- Abdomen
- Lower back (flanks)
- Buttocks

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.

7.4 Run `glm` for selected body sites to check for issues

P-values reported for the likelihood ratio tests below are uncorrected for multiple comparisons.

```
#-- Throat --#
throat <- data_log.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,
              data = throat,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     120.18
## 2       489     113.06  3   7.1189   0.0682 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## No significant difference between the null and full models

#-- Arms --#
arms <- data_log.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,
              data = arms,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     179.82
## 2       489     178.32  3   1.5019   0.6818
```

```
## No significant difference between the null and full models

#-- Elbow --#
elbows <- data_log.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,
              data = elbows,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     167.37
## 2       489     160.84  3   6.5329  0.08838 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## No significant difference between the null and full models

#-- Wrist & hand --#
wrists <- data_log.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,
              data = wrists,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492      242.2
## 2       489      232.3  3   9.9008  0.01943 *
## ---
## 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)
```

```r
#-- Abdomen --#
abdomen <- data_log.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,
              data = abdomen,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     578.82
## 2       489     570.04  3   8.7714  0.03249 *
## ---
## 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)

#-- Lower back (flank) --#
low_back <- data_log.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,
              data = low_back,
              family = binomial())
```

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     262.78
## 2       489     260.80  3   1.9776   0.5771
```

```
## No significant difference between the null and full models

#-- Buttocks --#
buttocks <- data_log.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,
              data = buttocks,
              family = binomial())

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

```
## Analysis of Deviance Table
##
## Model 1: Pain_present ~ 1
## Model 2: Pain_present ~ Age + Sex + CD4_recent
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       492     160.99
## 2       489     158.75  3   2.2349   0.5251
```

```
## No significant difference between the null and full models
```

7.5 GLMM data for tabulation and plotting

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

```
# Perform GLMM on each selected body sites
glmm_mods <- data_log.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")) %>%
    # 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 +
                                      (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 = 'LR'))) %>%
    # 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,
                          ~ exp(confint.merMod(.x,
                                               method = 'Wald')[2:5, ]))) %>%
    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) %>%
                         .[2:4, ] %>%
                         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'
                                      )) %>%
                                  # 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 = 3.3,
                                           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 = 12),
                                        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))))
```

7.6 GLM data for tabulation and plotting

Perform GLM on selected body sites (throat, arms, elbows, wrists and hands, abdomen, and buttocks) 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_log.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")) %>%
    # 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,
                                data = .x,
                                family = binomial())),
           null = map(.x = data,
                      ~glm(Pain_present ~ 1,
                             data = .x,
                             family = binomial()))) %>%
    # Perform likelihood ratio test
    mutate(LRT = map2(.x = logistic,
                      .y = null,
                      ~ anova(.y, .x, test = 'LR'))) %>%
    # 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) %>%
                         .[2:4, ] %>%
                         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'
                                      )) %>%
                                  # 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 = 3.3,
                                           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 = 12),
                                        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))))
```

7.7 Tabulated results (fixed effects)

7.7.1 GLMM tabulations

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

Table 55: Head (GLMM; LRT corrected p-value = 0.025)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.013	0.797	1.288
SexMale	0.260	0.122	0.552
CD4_recent	0.872	0.698	1.089

Table 56: Shoulder (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.875	0.605	1.266
SexMale	1.745	0.773	3.943
CD4_recent	0.948	0.659	1.363

Table 57: Chest (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.033	0.795	1.344
SexMale	0.948	0.511	1.757
CD4_recent	0.742	0.552	0.995

Table 58: Cervical spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.152	0.685	1.939
SexMale	0.641	0.132	3.115
CD4_recent	0.951	0.593	1.527

Table 59: Thoracic spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.952	0.702	1.291
SexMale	0.505	0.198	1.284
CD4_recent	1.118	0.861	1.452

Table 60: Lumbosacral spine (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.023	0.787	1.329
SexMale	1.025	0.524	2.008
CD4_recent	1.248	0.987	1.580

Table 61: Groin (GLMM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.939	0.676	1.304
SexMale	0.530	0.224	1.257
CD4_recent	1.219	0.880	1.687

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

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.146	0.783	1.676
SexMale	1.392	0.581	3.339
CD4_recent	1.157	0.818	1.637

Table 63: Legs (GLMM; LRT corrected p-value = 0.738)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.170	0.899	1.522
SexMale	1.981	1.087	3.612
CD4_recent	1.224	0.954	1.571

Table 64: Knees (GLMM; LRT corrected p-value = 0.766)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.345	1.033	1.751
SexMale	1.618	0.862	3.038
CD4_recent	1.151	0.883	1.500

Table 65: Ankles & Feet (GLMM; LRT corrected p-value = 0.664)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.335	1.062	1.678
SexMale	1.429	0.845	2.419
CD4_recent	0.977	0.785	1.217

7.7.2 GLM tabulations

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

Table 66: Throat (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.098	0.587	2.054
SexMale	2.601	0.780	8.678
CD4_recent	0.471	0.201	1.105

Table 67: Arms (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.263	0.834	1.912
SexMale	0.873	0.279	2.734
CD4_recent	0.825	0.510	1.333

Table 68: Elbows (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.616	1.101	2.371
SexMale	0.889	0.279	2.834
CD4_recent	1.118	0.729	1.715

Table 69: Wrists & Hands (GLM; LRT corrected p-value = 0.408)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.348	0.976	1.862
SexMale	0.547	0.180	1.661
CD4_recent	1.407	1.033	1.916

Table 70: Lower back/flanks (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	1.171	0.844	1.624
SexMale	0.833	0.326	2.127
CD4_recent	1.146	0.834	1.576

Table 71: Abdomen (GLM; LRT corrected p-value = 0.682)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.818	0.653	1.026
SexMale	0.705	0.393	1.264
CD4_recent	0.860	0.695	1.066

Table 72: Buttocks (GLM; LRT corrected p-value = 1)

	OR	Wald lower 95%CI	Wald upper 95%CI
Age	0.891	0.541	1.469
SexMale	1.955	0.640	5.971
CD4_recent	1.284	0.844	1.952

7.8 Plotted results (fixed effects)

Includes both GLM and GLMM results

```
# 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())

legs <- glmm_mods$forest_plots[[9]] +
    theme(axis.title.x = element_blank())

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

ankles.feet <- glmm_mods$forest_plots[[11]] +
    theme(axis.text.y = 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]] +
    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())

buttocks <- glm_mods$forest_plots[[7]] +
    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_3.png',
       plot = log_patch,
       width = 16,
       height = 18)
```

8 Summary plots

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

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

# Generate a filter to extract sites with >10% pain prevalence
filter <- data_log %>%
    select(-Site, -CD4_recent, -Age, -Sex) %>%
    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_log %>%
    pivot_longer(cols = -c(Site, CD4_recent, Sex, Age),
                 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',
```

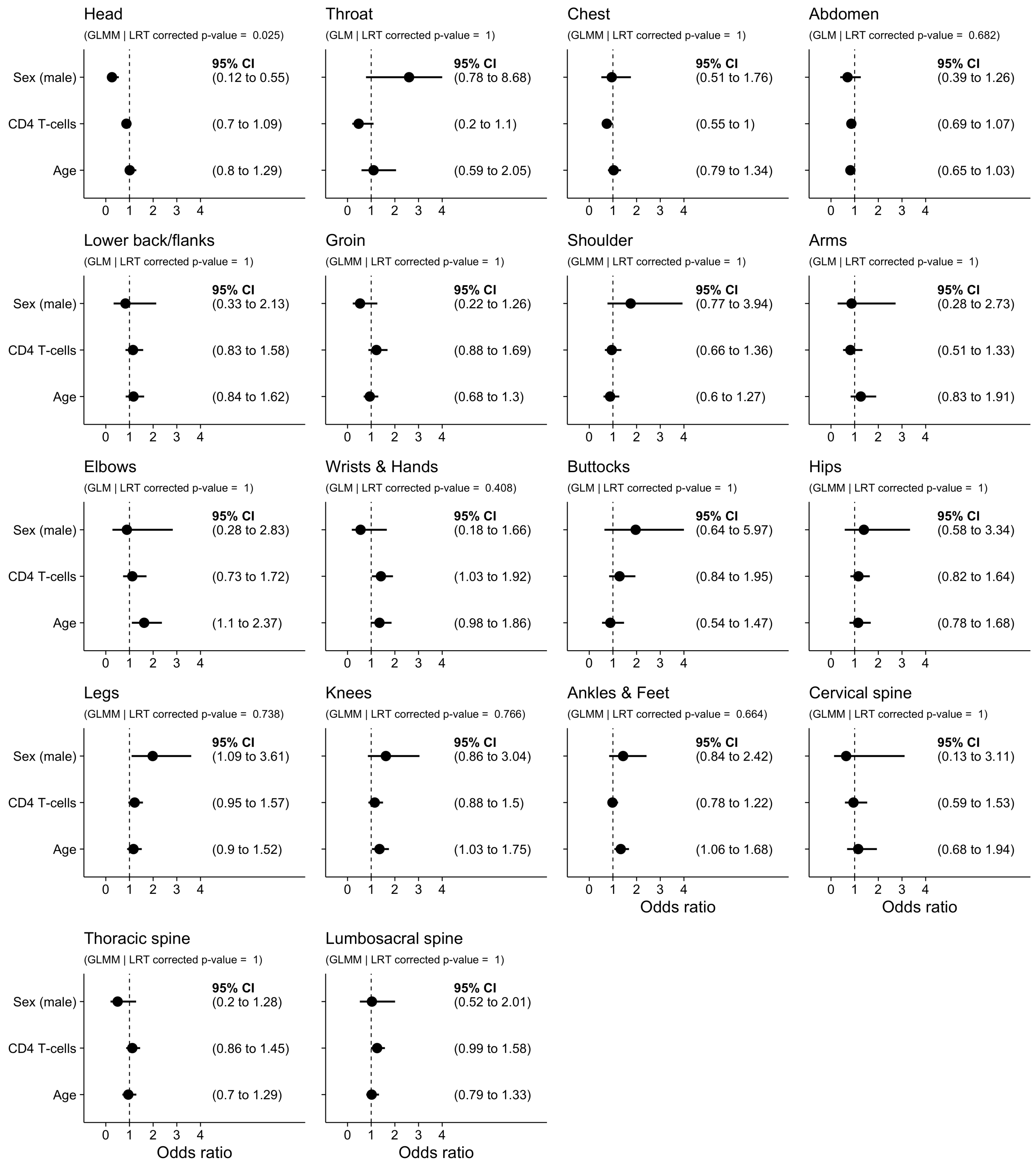


Figure 1: Plotted results (fixed effects)

```
                         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) %>%
    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', 'Groin',
                                         '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) %>%
    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', 'Groin',
                                         '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))

#-- Body sites by age --#
pubs_nofacet.age <- data_geq10 %>%
    select(-CD4_recent, -Site, -Sex) %>%
    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)) %>%
```

```r
    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(),
          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) %>%
    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(-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),
```

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

# Save
ggsave(filename = 'figures/figure_2.png',
       width = 14,
       height = 12)
```

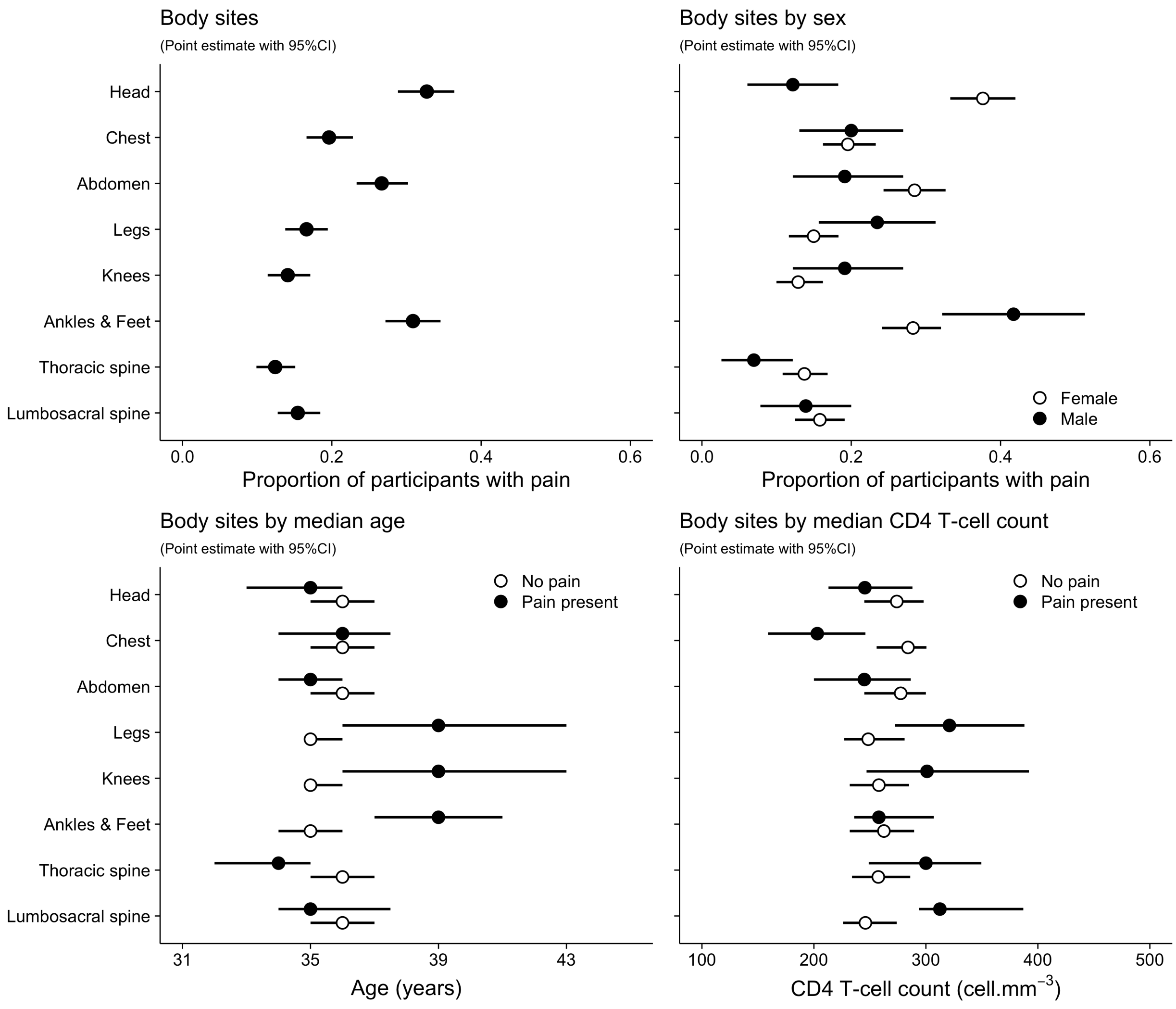


Figure 2: Body sites with a point estimate pain proportion >10%

8.2 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_log %>%
    pivot_longer(cols = -c(Site, CD4_recent, Sex, Age),
                 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'))) %>%
    # 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) %>%
    distinct() %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '_',
                                   replacement = ' ')) %>%
    mutate(Pain_site = str_replace(Pain_site,
                                   pattern = '\\.',
                                   replacement = ' & ')) %>%
```

```r
    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', '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) %>%
    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', '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,
                   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 age --#
pubs_nofacet.age2 <- data_less10 %>%
    select(-CD4_recent, -Site, -Sex) %>%
    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', '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(),
          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) %>%
    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', '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(),
          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))
```

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

# Save
ggsave(filename = 'figures/figure_supp2.png',
       width = 14,
       height = 12)
```

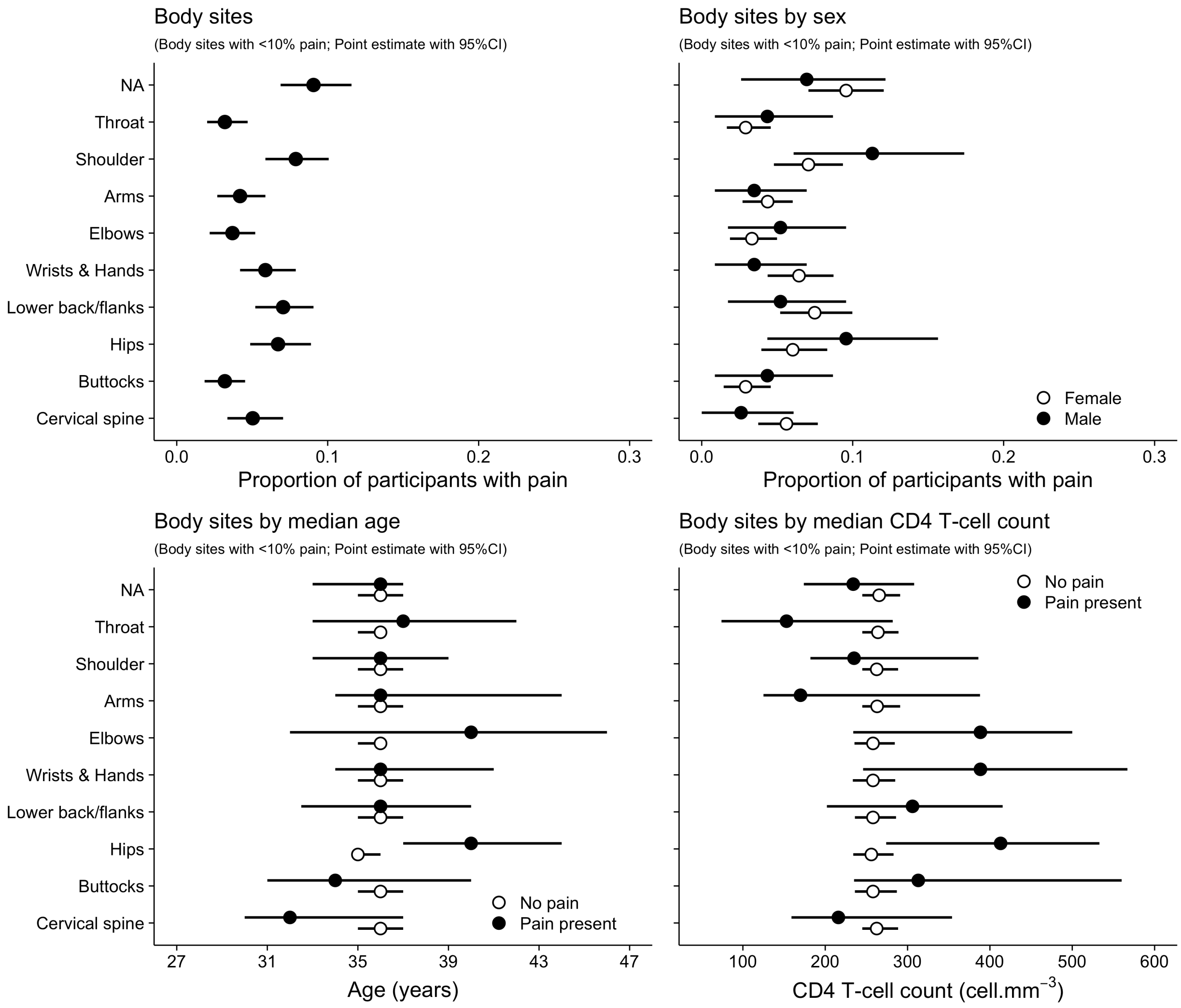


Figure 3: Body sites with a point estimate pain proportion <10%

9 Session information

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] patchwork_1.0.0    lme4_1.1-21        Matrix_1.2-18      boot_1.3-24
##  [5] knitr_1.28         skimr_2.1          forcats_0.5.0      stringr_1.4.0
##  [9] dplyr_0.8.5        purrr_0.3.3        readr_1.3.1        tidyr_1.0.2
## [13] tibble_3.0.0       ggplot2_3.3.0.9000 tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4       lubridate_1.7.4  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_1.1.5  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    munsell_0.5.0
## [25] broom_0.5.5      compiler_3.6.3   modelr_0.1.6     xfun_0.12
## [29] pkgconfig_2.0.3  base64enc_0.1-3  htmltools_0.4.0  tidyselect_1.0.0
## [33] fansi_0.4.1      crayon_1.3.4     dbplyr_1.4.2     withr_2.1.2
## [37] MASS_7.3-51.5    grid_3.6.3       nlme_3.1-145     jsonlite_1.6.1
## [41] gtable_0.3.0     lifecycle_0.2.0  DBI_1.1.0        magrittr_1.5
## [45] scales_1.1.0     cli_2.0.2        stringi_1.4.6    farver_2.0.3
## [49] fs_1.3.1         xml2_1.3.0       ellipsis_0.3.0   generics_0.0.2
## [53] vctrs_0.2.4      tools_3.6.3      glue_1.3.2       hms_0.5.3
## [57] parallel_3.6.3   yaml_2.2.1       colorspace_1.4-1 rvest_0.3.5
## [61] haven_2.2.0
```