

Supplement 2

Sites of pain

Peter Kamerman

Last knitted: 25 May 2021

Contents

1	Import and check data	1
2	Basic descriptive statistics	3
2.1	Pain sites	3
2.2	Demographics	3
3	Bootstrap function	4
4	Proportion with pain, by body site	4
4.1	Process data	4
4.2	Tabulated data	7
4.3	Plotted data	8
5	Proportion with pain, by body site and sex	12
5.1	Process data	12
5.2	Tabulated data	15
5.3	Plotted data	16
5.4	Hypothesis testing	22
6	Session information	23

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", "RPB78"~
## $ Head <chr> "No", "No", "No", "Yes", "Yes", "No", "No", "No", "Y~
## $ Throat <chr> "No", "No", "No", "No", "No", "No", "No", "No", "Yes~
## $ Shoulder <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Arms <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Elbows <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Wrists.Hands <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Chest <chr> "No", "No", "No", "Yes", "No", "No", "No", "No", "Ye~
## $ Upper_back <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Lower_back <chr> "No", "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", "No"~
## $ Thoracic_spine <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Lumbosacral_spine <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Groin <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ Hips <chr> "No", "No", "No", "No", "No", "No", "No", "Yes", "No~
## $ Legs <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes", "N~
## $ Knees <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes", "N~
## $ Ankles.Feet <chr> "No", "No", "No", "No", "No", "Yes", "No", "Yes", "N~
## $ Buttocks <chr> "No", "No", "No", "No", "No", "No", "No", "Yes", "No~
## $ Site <chr> "RP", "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", "RPB78"~
## $ Site <chr> "RP", "RP", "RP", "RP", "RP", "RP", "RP", "RP"~
## $ Sex <chr> "Female", "Female", "Female", "Female", "Female", "F~
## $ Age <dbl> 36, 27, 39, 36, 31, 32, 28, 37, 31, 25, 31, 24, 35, ~
## $ Employment_status <chr> "Other", "Unemployed", "Other", "Unemployed", "Unemp~
## $ CD4_recent <dbl> 391, 571, 591, 207, 126, 225, 543, 410, 74, 212, 579~
## $ ART_currently <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye~
## $ Education <chr> "Tertiary", "Secondary", "Secondary", "Primary", "Se~
```

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

2.2 Demographics

```
demo %>%  
  select(-ID, -Site) %>%  
  select(Sex) %>%
```

```
mutate(Sex = factor(Sex)) %>%
skim()
```

Table 3: Data summary

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

Variable type: factor

skim_variable	missing	complete	n_unique	top_counts
Sex	0	596	2	Fem: 481, Mal: 115

3 Bootstrap function

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

4 Proportion with pain, by body site

Plotted as point estimates with bootstrapped 95% CIs

4.1 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 %>%
  # Remove Site column
  select(-Site) %>%
  # Pivot to long format
  pivot_longer(cols = everything(),
               names_to = 'body_site',
               values_to = 'pain_present') %>%
  # Add body regions
  mutate(region = case_when(
    body_site == 'Chest' |
      body_site == 'Head' |
      body_site == 'Throat' |
      body_site == 'Shoulder' ~ 'Head and upper torso',
    body_site == 'Lower_back' |
      body_site == 'Abdomen' |
      body_site == 'Hips' |
      body_site == 'Buttocks' |
      body_site == 'Groin' ~ 'Lower torso',
    body_site == 'Legs' |
      body_site == 'Knees' |
      body_site == 'Ankles.Feet' ~ 'Lower limbs',
    body_site == 'Arms' |
      body_site == 'Elbows' |
      body_site == 'Wrists.Hands' ~ 'Upper limbs',
    body_site == 'Cervical_spine' |
      body_site == 'Thoracic_spine' |
      body_site == 'Lumbosacral_spine' ~ 'Spinal column',
    TRUE ~ 'other'
  )) %>%
  # Nest by body region and body site
  group_by(region, body_site) %>%
  nest() %>%
  # Bootstrap 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_linerange(size = 1,
                                    colour = '#000000') +
                     geom_point(shape = 21,
                                colour = '#000000',
                                fill = '#000000',
                                size = 3,
                                stroke = 1) +
                     coord_flip() +
                     labs(title = .y,
                          subtitle = '(Point estimate with 95%CI)',
                          y = 'Proportion 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)))) %>%

```

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

4.2 Tabulated data

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

Table 5: 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 6: 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 7: 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 8: Spinal column

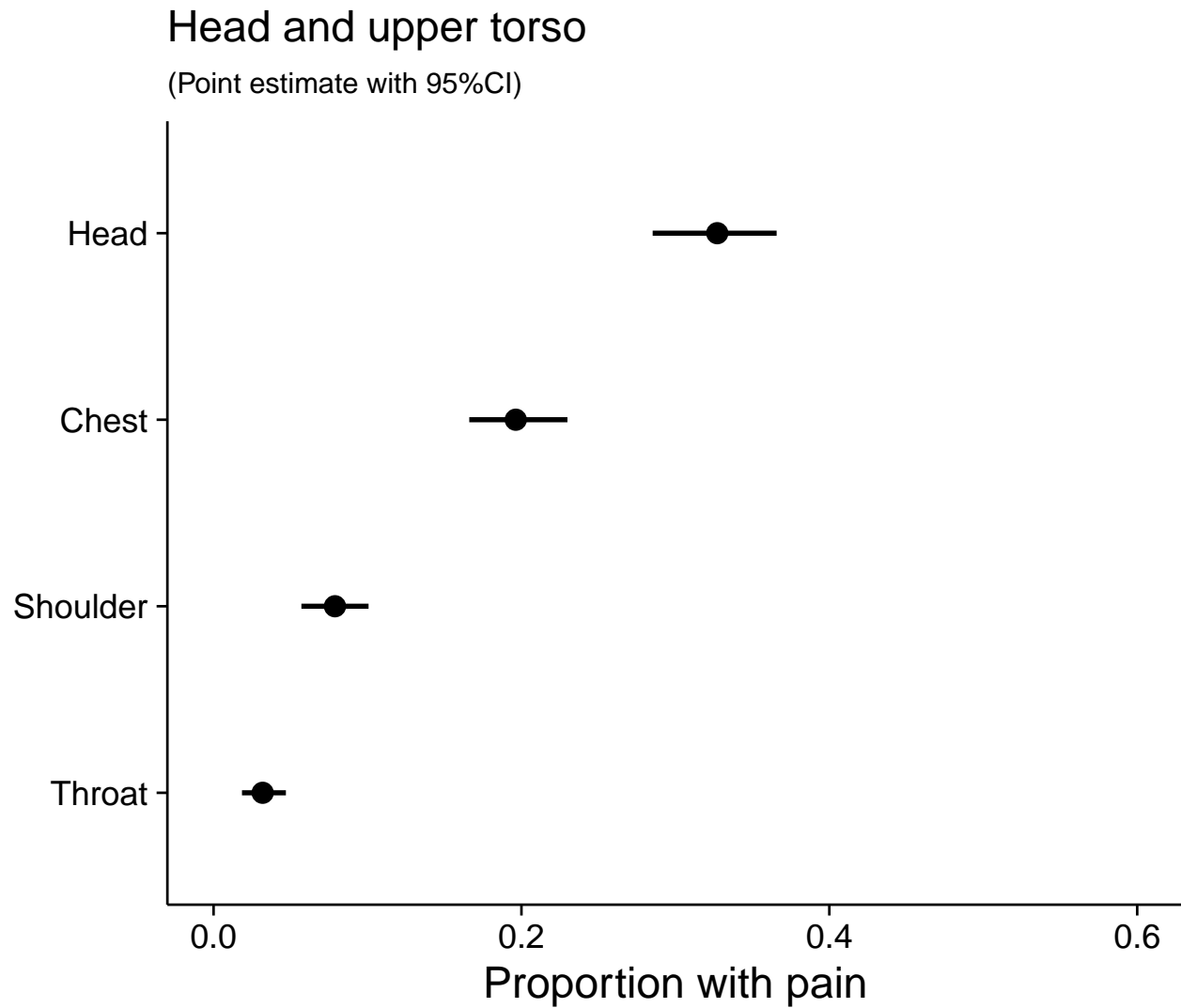
body_site	point_est	lower_ci	upper_ci
Cervical spine	0.05	0.03	0.07
Thoracic spine	0.12	0.10	0.15
Lumbosacral spine	0.15	0.13	0.18

Table 9: 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

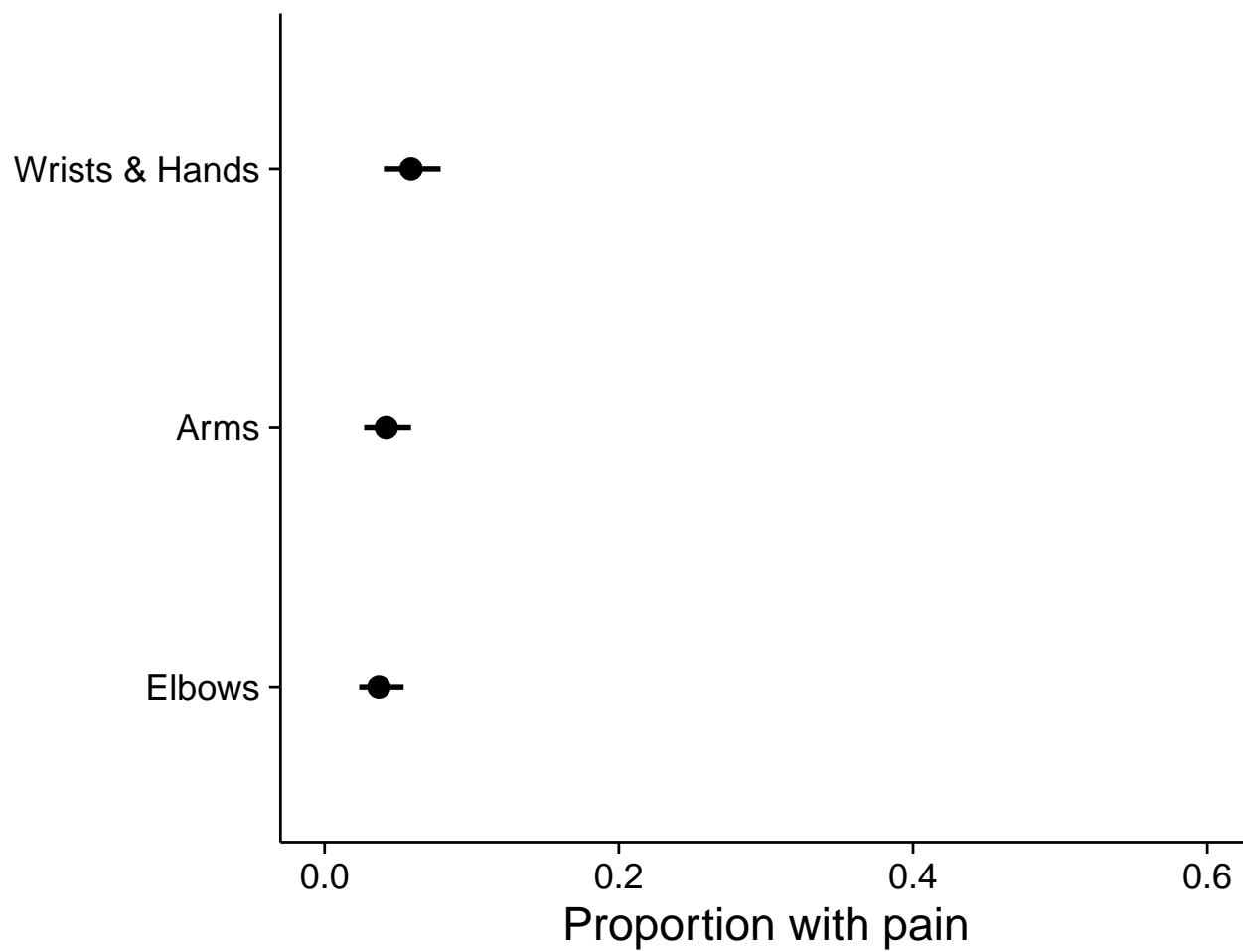
4.3 Plotted data

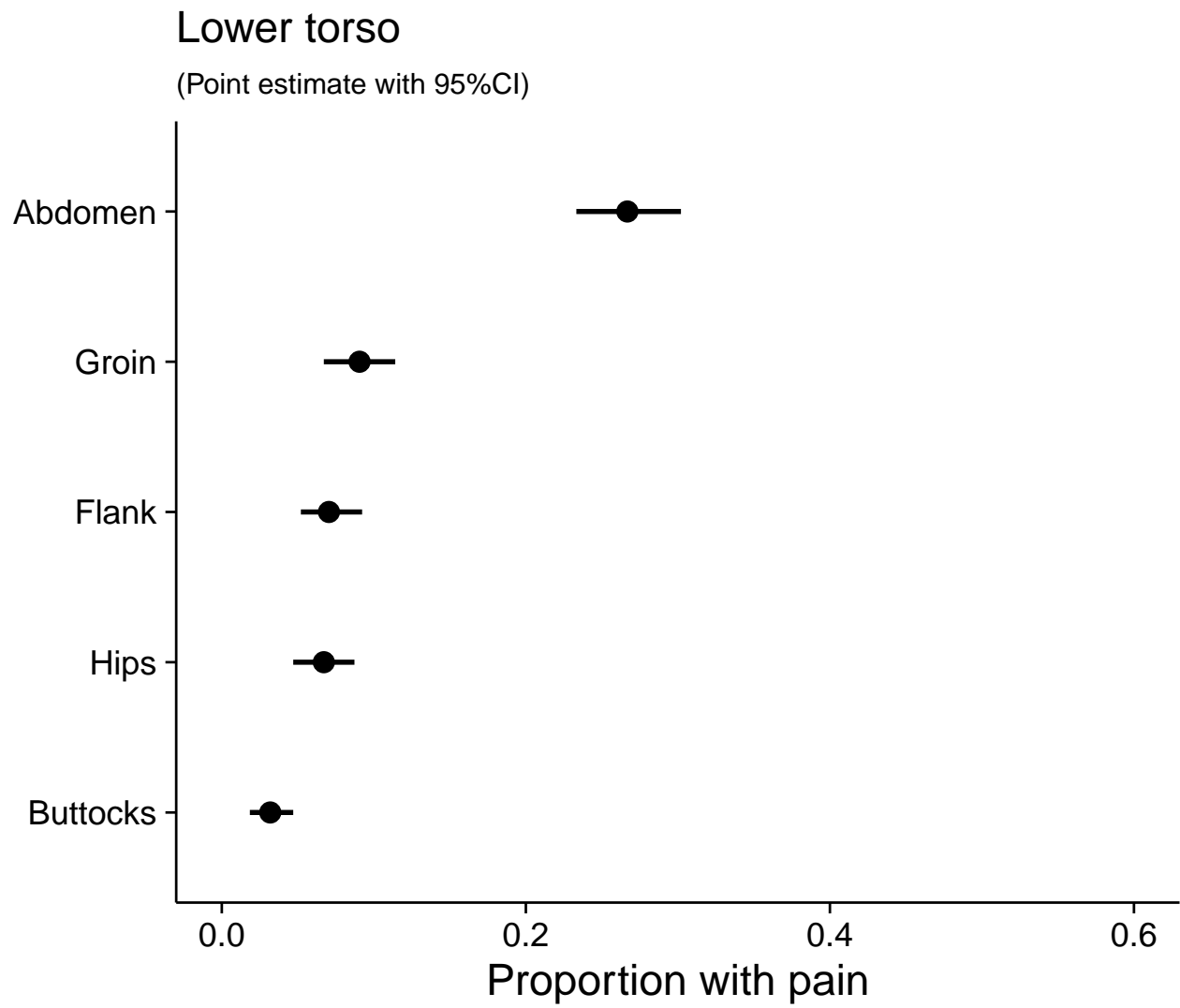
```
walk(prop_boot2$plots, ~ print(.x))
```

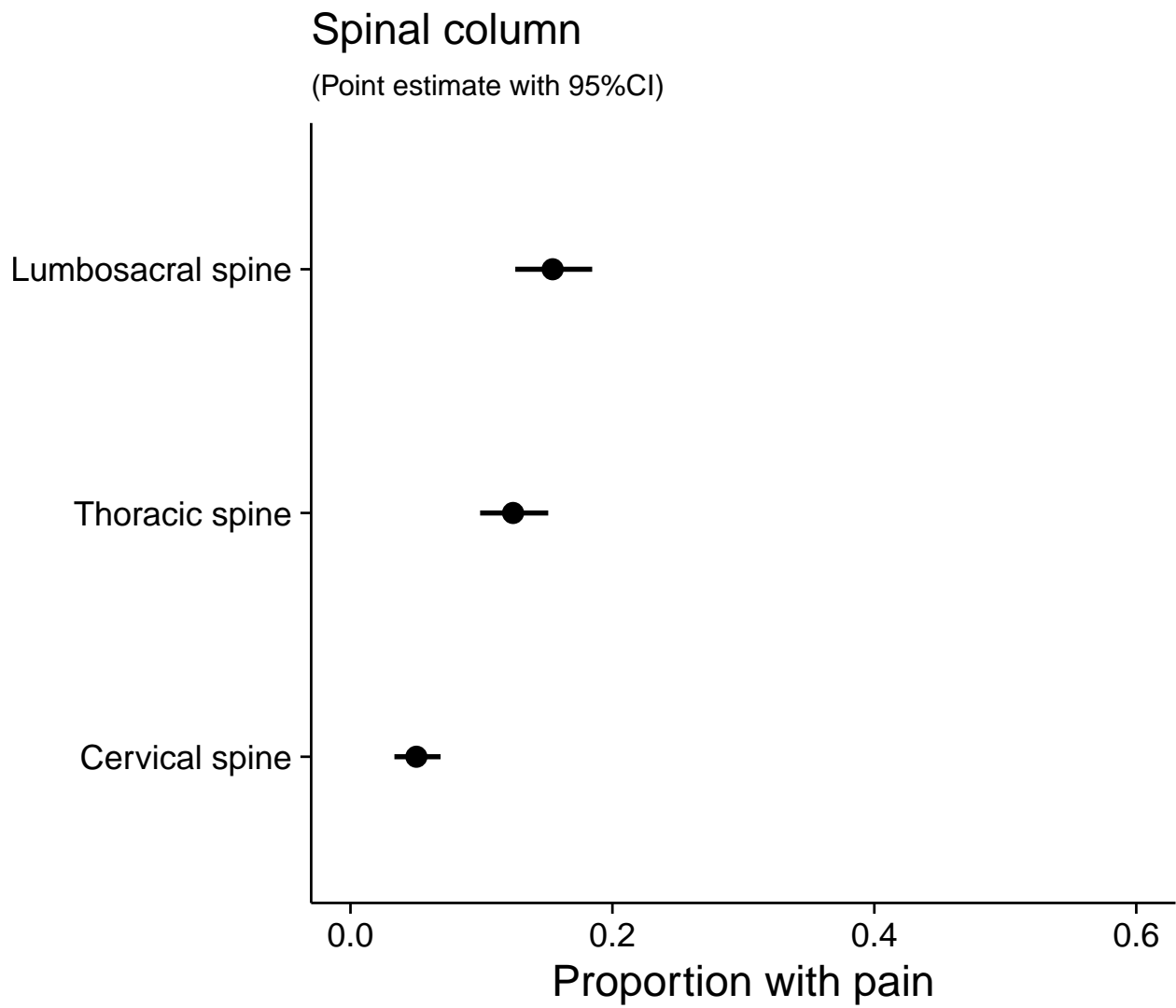


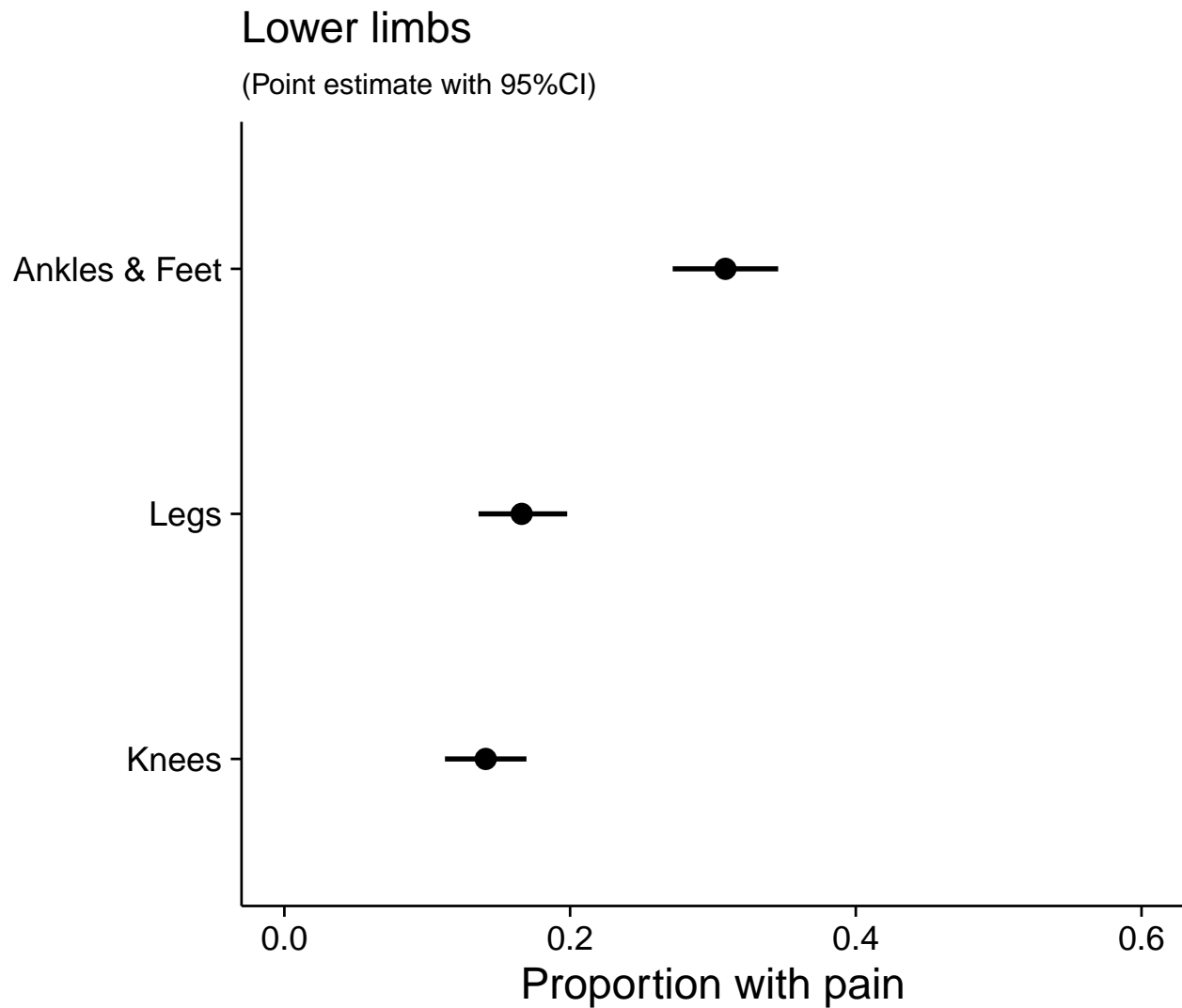
Upper limbs

(Point estimate with 95%CI)









5 Proportion with pain, by body site and sex

Plotted as point estimates with bootstrapped 95% CIs.

5.1 Process data

```
# 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() %>%
  # Bootstrap 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 = 4,
                                   stroke = 1) +
                        coord_flip() +
                        labs(title = .y,
                             subtitle = '(Point estimate with 95%CI)',
                             y = 'Proportion with pain') +
                        scale_y_continuous(limits = c(0, 0.6)) +
                        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 = c(0.8, 0.87),
                              legend.background = element_rect(colour = '#000000',
                                                                    size = 0.3),

                              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 %>%
      arrange(body_site) %>%
      kable(caption = .y,
        digits = 2)))

```

5.2 Tabulated data

```

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

```

Table 10: Head and upper torso

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

Table 11: Upper limbs

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

Table 12: Lower torso

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

Table 13: Spinal column

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

Table 14: Lower limbs

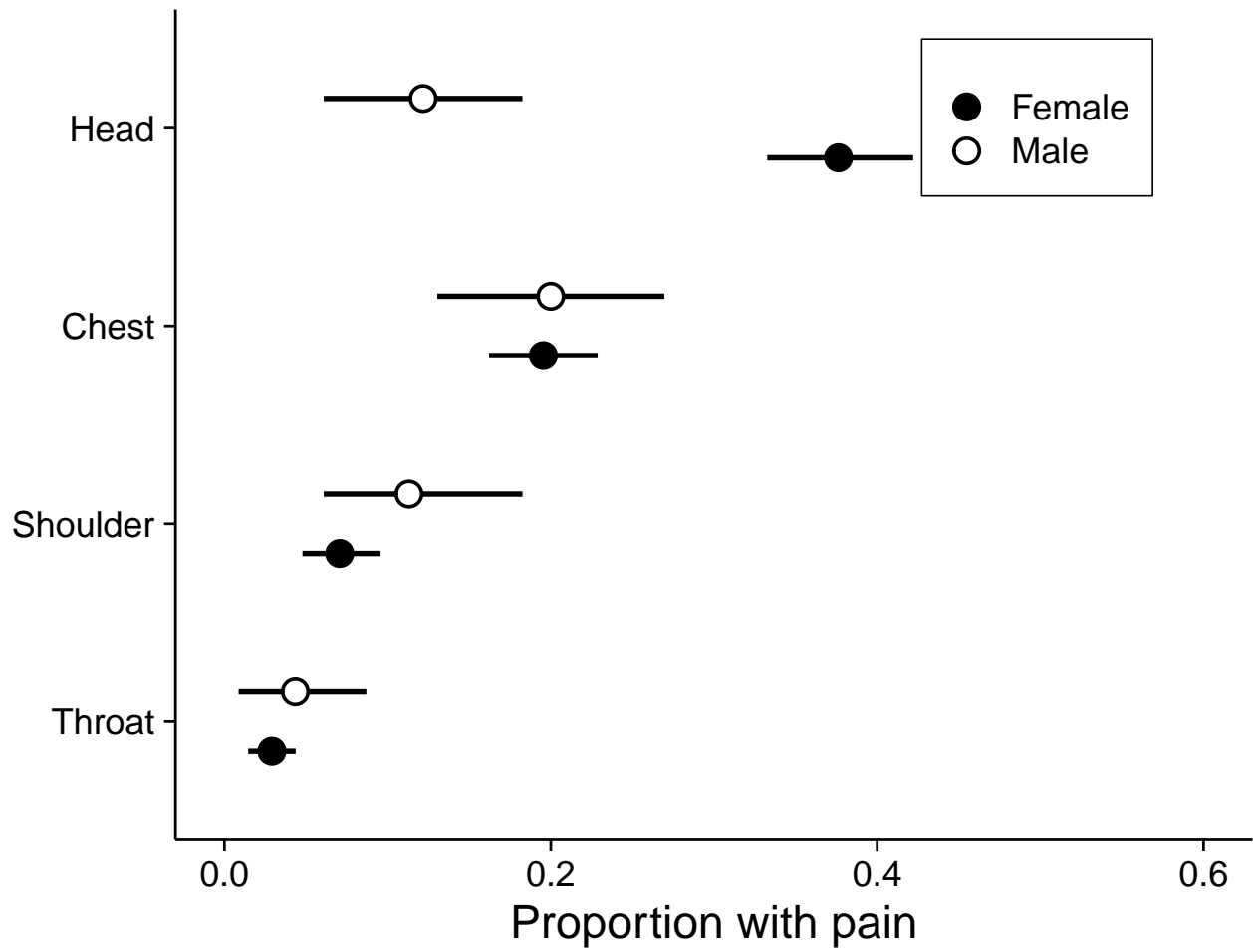
Sex	body_site	point_est	lower_ci	upper_ci
Female	Knees	0.13	0.10	0.16
Male	Knees	0.19	0.12	0.26
Female	Legs	0.15	0.12	0.18
Male	Legs	0.23	0.16	0.31
Female	Ankles & Feet	0.28	0.25	0.32
Male	Ankles & Feet	0.42	0.33	0.50

5.3 Plotted data

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

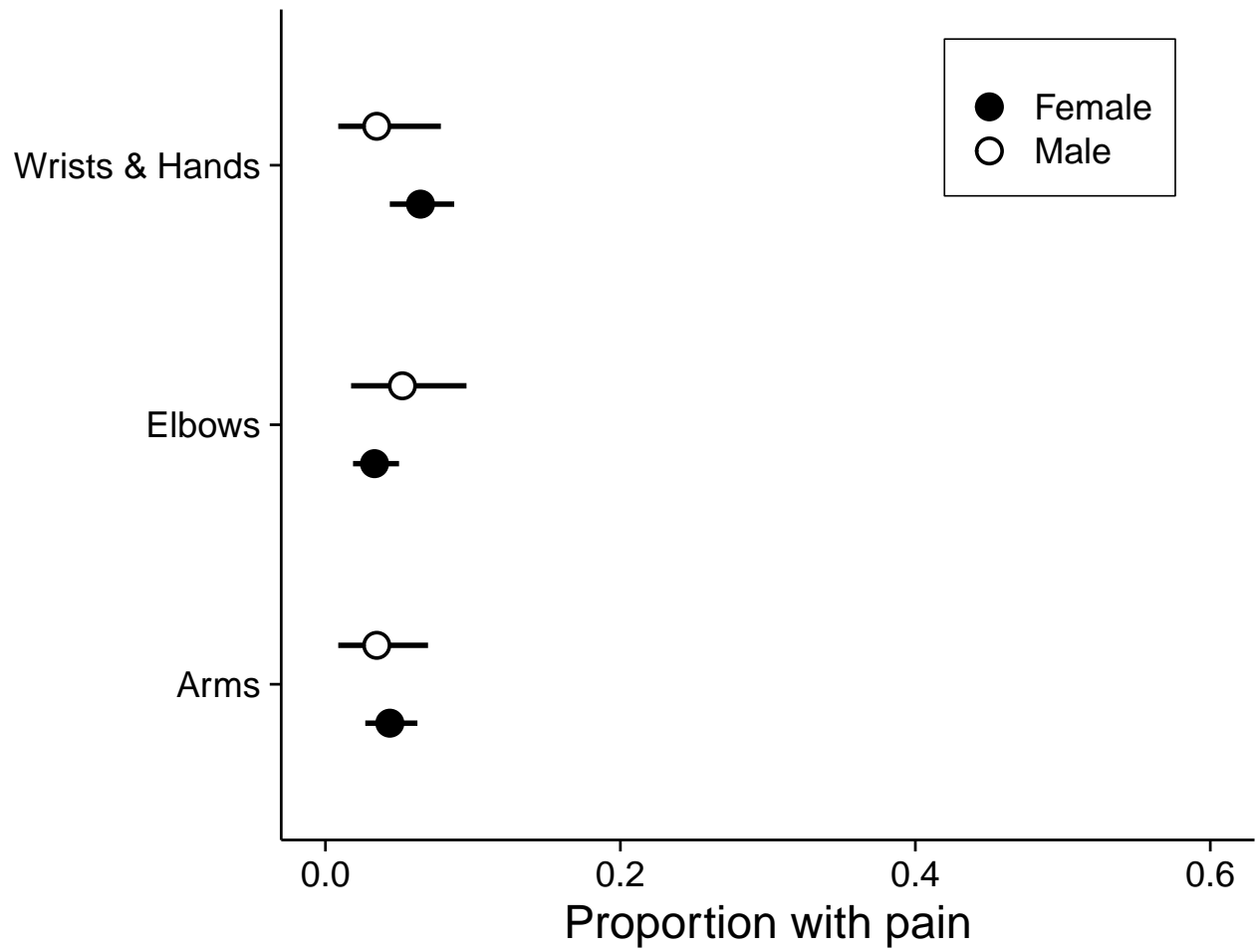

Head and upper torso

(Point estimate with 95%CI)



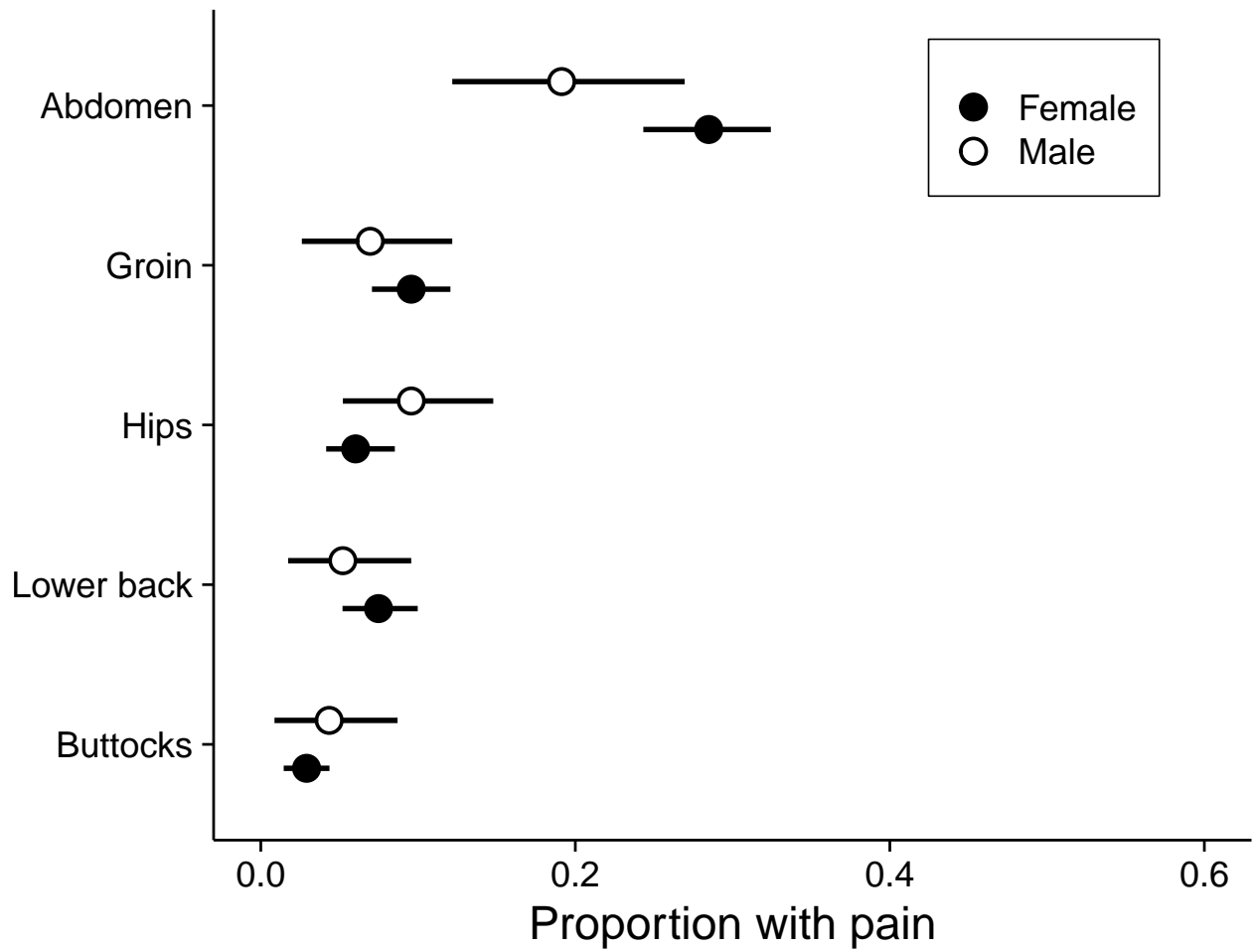
Upper limbs

(Point estimate with 95%CI)



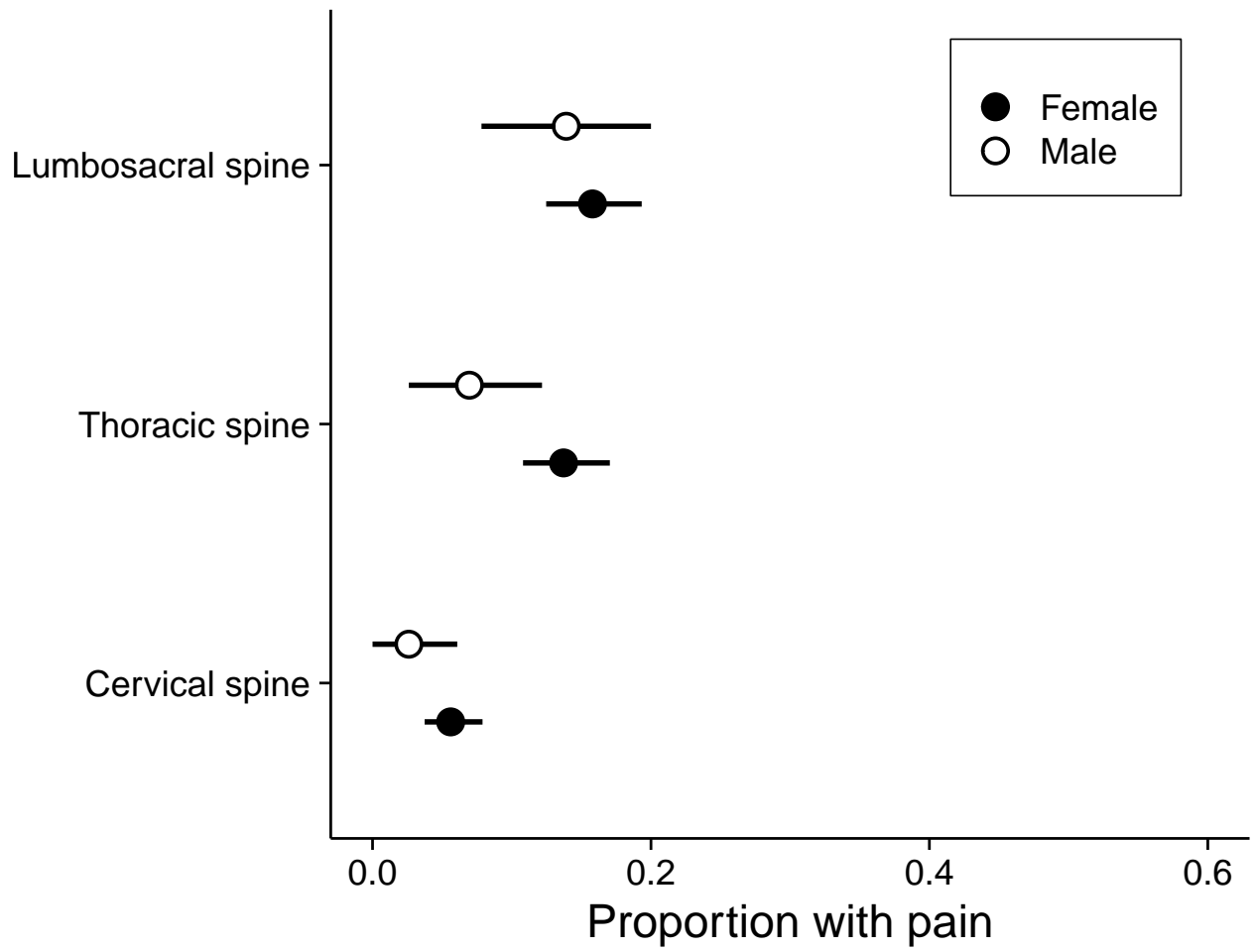
Lower torso

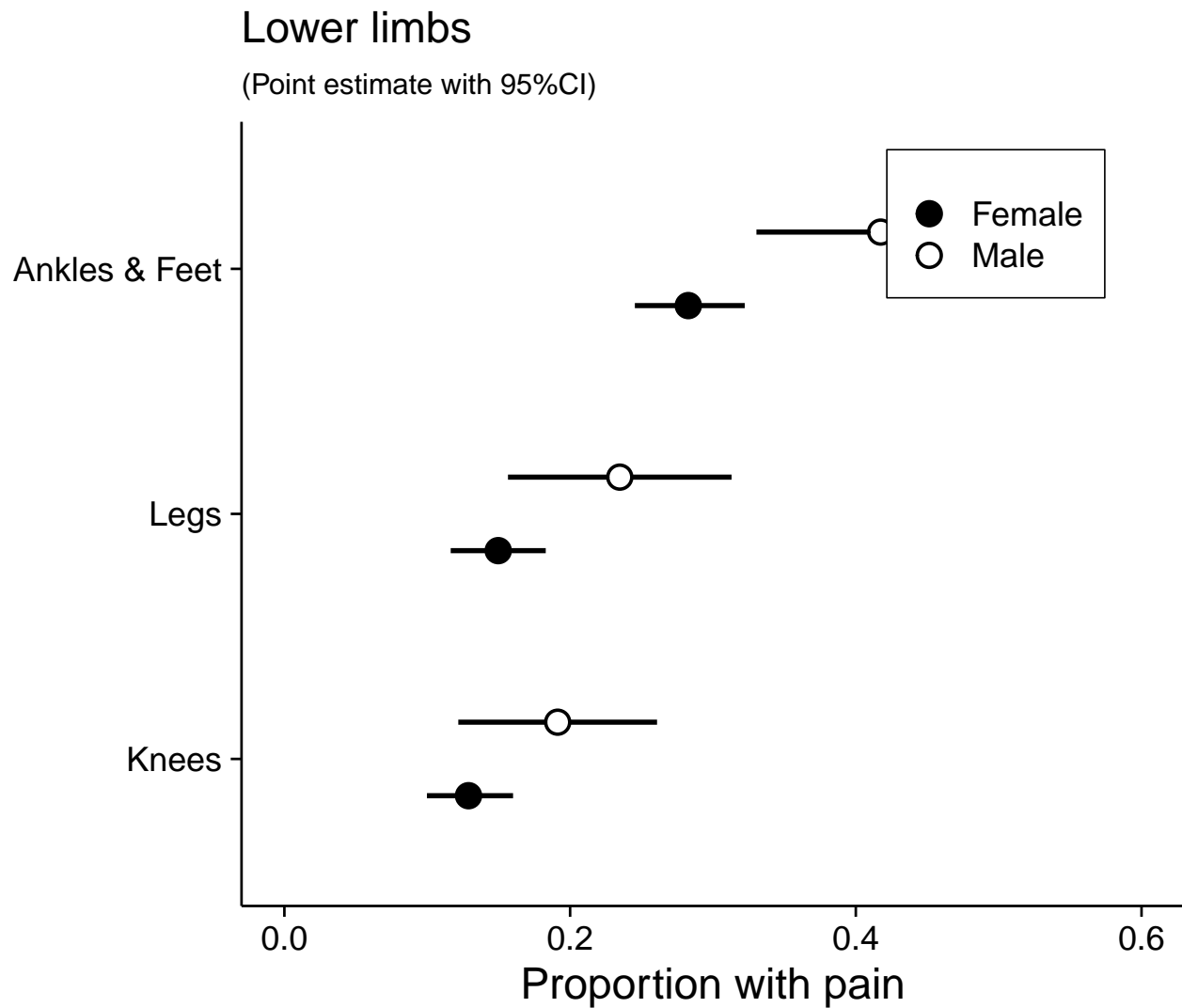
(Point estimate with 95%CI)



Spinal column

(Point estimate with 95%CI)





```
# Clean-up individual plots
sex_upper <- sex_boot2$plots[[1]] +
  theme(axis.title.x = element_blank(),
        plot.subtitle = element_blank(),
        legend.position = 'none')

sex_arm <- sex_boot2$plots[[2]] +
  theme(axis.title.x = element_blank(),
        plot.subtitle = element_blank(),
        legend.position = 'none')

sex_lower <- sex_boot2$plots[[3]] +
  theme(plot.subtitle = element_blank(),
        axis.title.x = element_blank())

sex_spine <- sex_boot2$plots[[4]] +
  theme(plot.subtitle = element_blank(),
        legend.position = 'none')

sex_leg <- sex_boot2$plots[[5]] +
```

```

  theme(plot.subtitle = element_blank(),
        legend.position = 'none')

# Organise individual plots
sex_prop <- sex_upper + sex_lower + sex_arm + sex_leg + sex_spine +
  plot_layout(ncol = 2)

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

```

5.4 Hypothesis testing

Fisher's Exact test for each body part. Family-wise error rate controlled using the Holm method.

```

sex_long <- sex %>%
  # Convert to long format
  pivot_longer(cols = -Sex,
               names_to = 'body_part',
               values_to = 'pain_present') %>%

  # Nest by body part
  group_by(body_part) %>%
  nest() %>%

  # Tabulate sex ~ pain_present
  mutate(tab = map(.x = data,
                   ~ xtabs(~ Sex + pain_present, data = .x))) %>%

  # Perform Fisher's Exact test
  mutate(fisher = map(.x = tab,
                     ~ tidy(fisher.test(.x)) %>%
                        select(estimate, conf.low, conf.high, p.value) %>%
                        rename(odds.ratio = estimate))) %>%

  # Extract p-values
  mutate(p_value = map(.x = fisher,
                      ~ .x$p.value))

# Get adjusted p-value (Holm method)
body_part <- sex_long$body_part
p_value <- unlist(sex_long$p_value)
names(p_value) <- body_part
adjusted_p <- data.frame(p.value_holm = p.adjust(p_value,
                                                method = 'holm'))

# Output
bind_rows(sex_long$fisher) %>%
  bind_cols(data.frame(body.part = sex_long$body_part)) %>%
  bind_cols(adjusted_p) %>%
  select(body.part, everything()) %>%
  arrange(p.value_holm) %>%
  kable(caption = "Fisher's Exact test (Holm family-wise p-value correction)",
       digits = 3)

```

Table 15: Fisher’s Exact test (Holm family-wise p-value correction)

body.part	odds.ratio	conf.low	conf.high	p.value	p.value_holm
Head	0.230	0.118	0.420	0.000	0.000
Ankles.Feet	1.815	1.163	2.821	0.007	0.116
Legs	1.741	1.014	2.931	0.036	0.573
Abdomen	0.594	0.341	1.001	0.046	0.689
Thoracic_spine	0.471	0.189	1.023	0.058	0.809
Throat	1.515	0.418	4.571	0.387	1.000
Shoulder	1.674	0.782	3.396	0.175	1.000
Arms	0.790	0.193	2.406	0.800	1.000
Elbows	1.598	0.500	4.426	0.405	1.000
Wrists.Hands	0.524	0.132	1.526	0.275	1.000
Chest	1.029	0.589	1.745	0.897	1.000
Lower_back	0.681	0.229	1.688	0.542	1.000
Cervical_spine	0.451	0.086	1.506	0.239	1.000
Lumbosacral_spine	0.861	0.449	1.571	0.669	1.000
Groin	0.707	0.280	1.573	0.471	1.000
Hips	1.647	0.718	3.531	0.211	1.000
Knees	1.597	0.888	2.794	0.100	1.000
Buttocks	1.515	0.418	4.571	0.387	1.000

6 Session information

```

sessionInfo()

## R version 4.0.4 (2021-02-15)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/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_1.1.1 broom_0.7.5    boot_1.3-27    knitr_1.31
##  [5] skimr_2.1.3     forcats_0.5.1 stringr_1.4.0  dplyr_1.0.5
##  [9] purrr_0.3.4     readr_1.4.0   tidyr_1.1.3   tibble_3.1.0
## [13] ggplot2_3.3.3   tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0 xfun_0.22      repr_1.1.3     haven_2.3.1
##  [5] colorspace_2.0-0 vctrs_0.3.6   generics_0.1.0 htmltools_0.5.1.1

```

## [9]	base64enc_0.1-3	yaml_2.2.1	utf8_1.2.1	rlang_0.4.10
## [13]	pillar_1.5.1	glue_1.4.2	withr_2.4.1	DBI_1.1.1
## [17]	dbplyr_2.1.0	modelr_0.1.8	readxl_1.3.1	lifecycle_1.0.0
## [21]	munsell_0.5.0	gtable_0.3.0	cellranger_1.1.0	rvest_1.0.0
## [25]	evaluate_0.14	labeling_0.4.2	parallel_4.0.4	fansi_0.4.2
## [29]	highr_0.8	Rcpp_1.0.6	scales_1.1.1	backports_1.2.1
## [33]	jsonlite_1.7.2	farver_2.1.0	fs_1.5.0	hms_1.0.0
## [37]	digest_0.6.27	stringi_1.5.3	grid_4.0.4	cli_2.3.1
## [41]	tools_4.0.4	magrittr_2.0.1	crayon_1.4.1	pkgconfig_2.0.3
## [45]	ellipsis_0.3.1	xml2_1.3.2	reprex_1.0.0	lubridate_1.7.10
## [49]	assertthat_0.2.1	rmarkdown_2.7	httr_1.4.2	rstudioapi_0.13
## [53]	R6_2.5.0	compiler_4.0.4		