Supplement 3: Primary analysis 1A

Mean pain rating of 6.2 (SD 1.7) at 0.26, 0.37, 0.51 correlation

Peter Kamerman

Last knitted: 30 June 2020

Contents

---

1 Introduction

1.1 Overview

The use of pain intensity cut-offs for study inclusion in clinical trials has two consequences. Firstly, the cut-off artificially raises the baseline mean pain score of the cohort being studied compared to the population the cohort was sampled from. Secondly, unless the correlation between two sequential measurements is 1, there should be a "flattening" of the relationship between the first and subsequent measurements. This "flattening" means that the cut-off has a disproportionate effect on the mean baseline pain intensity compared to subsequent measurements.

This script demonstrates the effect of this "flattening" of the relationship between two sequential pain measurements in a hypothetical placebo group in a clinical trial for the management of pain in the presence of various pain intensity cut-off values for trial inclusion.

1.2 Modelling specifics

1.2.1 Data simulation parameters

Baseline pain scores were extracted from papers listed in the supplementary materials of a systematic review of pharmacological treatments for neuropathic pain[1] and that did not include baseline pain threshold inclusion criteria. Using this information, the pooled mean pain intensity at baseline was calculated to be 6.2 on an 11-point numerical pain rating scale (NRS), and the pooled SD of pain intensity at baseline was 1.7. The correlation between measurements 1 (V1) and 2 (V2) was set at 0.51 and 0.37 based on aggregated data across five clinical trials from 788 patients with rheumatoid arthritis for baseline vs week 4 and week 12 of the trial, respectively[2], and 0.26 based on aggregated data across nine clinical trials from 2017 patients with knee or hip arthritis or chronic low back pain for baseline vs week 12 of the trial[3].

1.2.2 Data simulation and processing

Covariance matrices were generated for each of the three combinations of correlation (r = 0.26, 0.37, or 0.51) and the SD (1.7) using the `cor2cov` function from the `MBESS` package[4]. Then, each of the three covariance matrices was used to a generate random sample from a bivariate normal distribution (n = 1000 per sample) with a mean of 6.2. The samples were generated using the `mvrnorm` function from the `MASS` package[5].

To check the data once they had been generated, measurement 1 (V1) and measurement 2 (V2) were plotted against each other using scatterplots with marginal density plots, and the sample means and SDs were calculated, as was the correlation between the two samples (*unconstrained data*). Thereafter, V1 values were constrained between 1 and 10 on an 11-point NRS to model participants entering a placebo-controlled clinical trial who can be expected to have at least

[1] Finnerup NB, Attal N, Haroutounian S, McNicol E, Baron R, Dworkin RH, Gilron I, Haanpää M, Hansson P, Jensen TS, Kamerman PR, Lund K, Moore A, Raja SN, Rice ASC, Rowbotham M, Sena E, Siddall P, Smith BH, Wallace M. Pharmacotherapy for neuropathic pain in adults: a systematic review and meta-analysis. Lancet Neurol 2015;14:162–173. doi:10.1016/S1474-4422(14)70251-0

[2] Vollert J, Cook NR, Kaptchuk TJ, Sehra ST, Bowen EX, Yong F, Zhang L, Tobias DK, Hall KT. Favorable placebo responses in objective and subjective outcome measures in rheumatoid arthritis clinical trials – implications on new drug development. *[Unpublished data]*

[3] Vase L, Vollert J, Finnerup NB, Miao X, Atkinson G, Marshall S, Nemeth R, Lange B, Liss C, Price DD, Maier C, Jensen TS, Segerdahl M. Predictors of the placebo analgesia response in randomized controlled trials of chronic pain: a meta-analysis of the individual data from nine industrially sponsored trials. Pain 2015;156:1795–1802. doi:10.1097/j.pain.0000000000000217.

[4] Kelley K. MBESS: The MBESS R package. 2019. Available: https://CRAN.R-project.org/package=MBESS

[5] Venables WN, Ripley BD. Modern Applied Statistics with S. Fourth. New York: Springer, 2002. Available: http://www.stats.ox.ac.uk/pub/MASS4

some pain. In comparison, V2 values were unconstrained and could take any value from 0 to 10 because pain intensity may take values across the full range of the scale on follow-up. To check for changes in centrality, spread, and correlation after constraining V1 data, the data were once again assessed using plots and numeric summaries. **Based on inspection of these plots and numeric summaries, it was deemed that constraining the lower tail of V1 had no major effect on the distribution and correlation structure of the data.**

1.2.3 Modelling the effect of baseline pain inclusion thresholds

A baseline inclusion threshold of 0 on an 11-point NRS (i.e., no inclusion threshold) was used as a "control". In addition to the control threshold, three baseline pain intensity inclusion thresholds were selected based on the data listed in the supplementary materials of Finnerup and colleagues[1]. The three thresholds were: pain intensity ≥ 4 (71% of listed studies), pain intensity ≥ 3 (12% of listed studies), and pain intensity ≥ 5 (11% of listed studies). Data from the three simulated datasets (mean = 6.2, SD = 1.7, correlation = 0.26, 0.37, 0.51) were each modelled under each of the four baseline pain intensity inclusion thresholds.

Modelling the effect of the thresholds involved removing all pairs of V1 and V2 data where V1 was less than the threshold value, yielding new datasets V1* and V2*. To assess the effect of removing data pairs on the group means, we calculated the difference in group means between V1 and V1*, and V2 and V2*. To assess the magnitude of the effect, we calculated the mean difference in pain intensity scores between V1* and V2* (point estimate), and generated bias-corrected and accelerated 95% bootstrapped confidence intervals of the difference (2000 resamples, generated using the `groupwiseMean` function from the `rcompanion` package[6]).

For reproducibility, I set the random seed to *"2019"* for all random sampling.

Note: Because of random sampling the mean (SD) of the samples differ slightly from the population mean (SD).

[6] Mangiafico S. rcompanion: functions to support extension education program evaluation. 2018. Available: https://CRAN.R-project.org/package=rcompanion.

2 Generate 2x2 covariance matrices

Generate a covariance matrix using an SD of 1.7 and correlation of $r = 0.26$, 0.37, and 0.51.

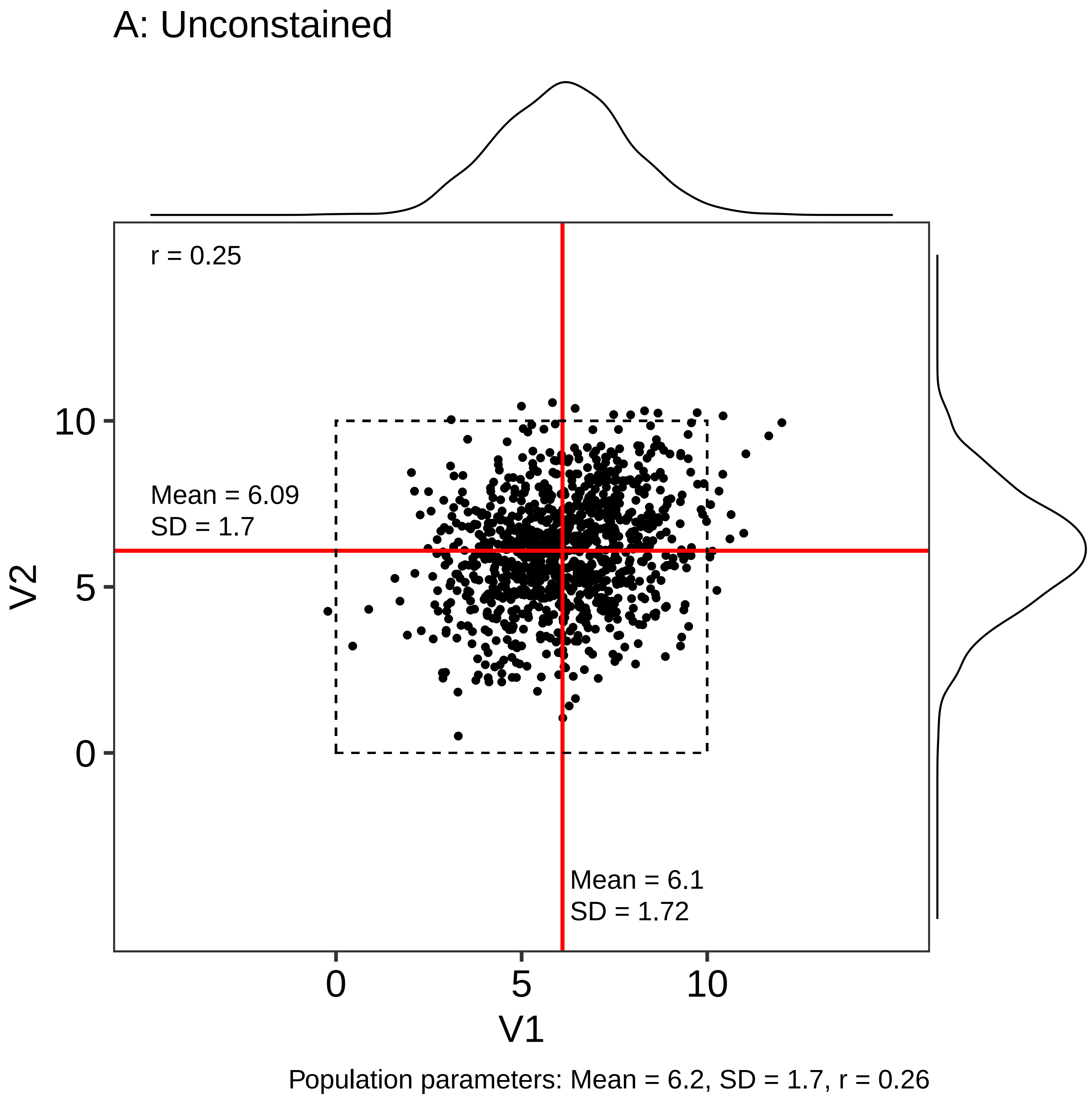
```
# Generate 2*2 correlation matrices
cor_026 <- matrix(c(1, 0.26, 0.26, 1), ncol = 2)
cor_037 <- matrix(c(1, 0.37, 0.37, 1), ncol = 2)
cor_051 <- matrix(c(1, 0.51, 0.51, 1), ncol = 2)

# Set the standard deviation
std <- c(1.7, 1.7)

# Generate a 2*2 covariance matrices
cov_026 <- cor2cov(cor.mat = cor_026,
                   sd = std)
cov_037 <- cor2cov(cor.mat = cor_037,
                   sd = std)
cov_051 <- cor2cov(cor.mat = cor_051,
                   sd = std)
```

3 Correlation: 0.26

3.1 Generate and summarise data

3.1.1 Unconstrained data

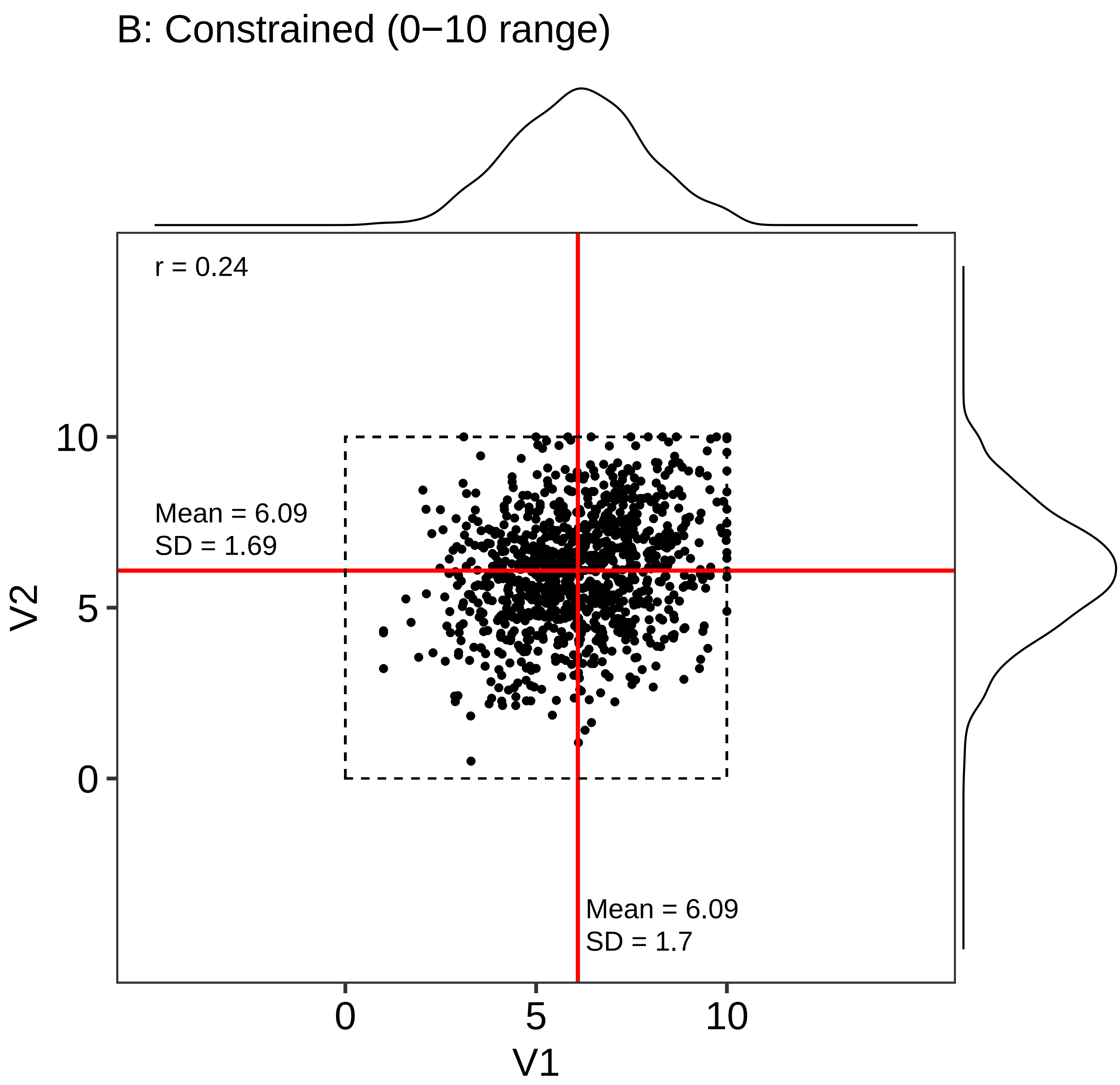
```
# Set the random seed for reproducibility
set.seed(2019)

# Generate the data
cor_026.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_026))

# Plot unconstrained data
ggMarginal(ggplot(data = cor_026.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_026.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_026.base$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_026.base$V1,
                                         cor_026.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_026.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_026.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_026.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_026.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_026.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_026.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_026.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_026.base$V1), 2)}")) +
               labs(title = 'A: Unconstained',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



3.1.2 Constrained data

```
# Constrain data
cor_026 <- cor_026.base %>%
    mutate(V1 = case_when(
                V1 < 1 ~ 1,
                V1 > 10 ~ 10,
                TRUE ~ V1)) %>%
    mutate(V2 = case_when(
                V2 < 0 ~ 0,
                V2 > 10 ~ 10,
                TRUE ~ V2)) %>%
```

```r
    mutate(group = 'No threshold')

# Plot constrained data
ggMarginal(ggplot(data = cor_026) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_026$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_026$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_026$V1,
                                         cor_026$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_026$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_026$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_026$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_026$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_026$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_026$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_026$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_026$V1), 2)}")) +
               labs(title = 'B: Constrained (0-10 range)',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



Population parameters: Mean = 6.2, SD = 1.7, r = 0.26

3.2 Effect of having a threshold on mean pain intensity scores

Constrained data only

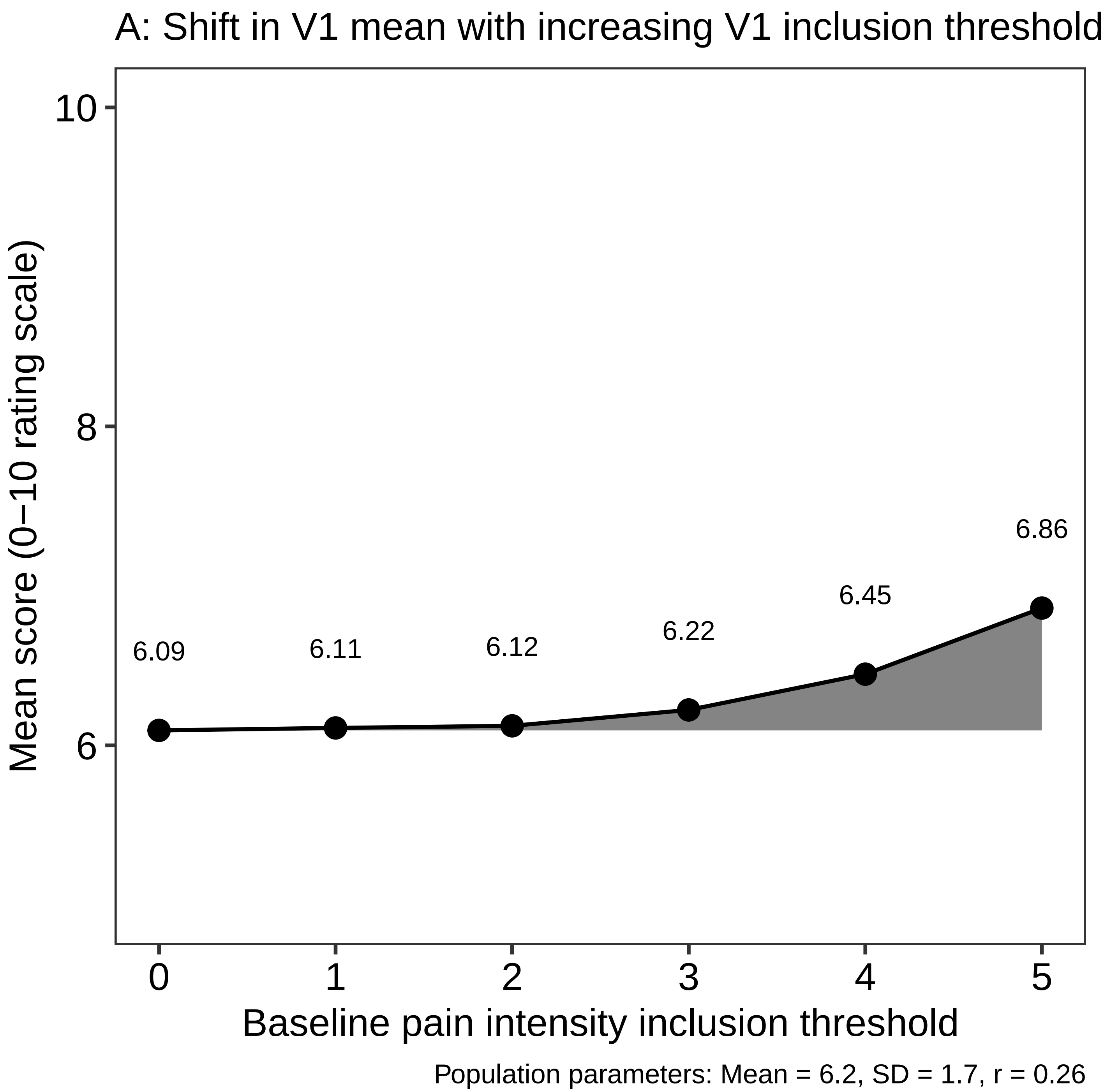
3.2.1 Model the mean of V1 with increasing pain inclusion thresholds from 0 to 5

```
# Extract visit 1 data
cor_026V1 <- cor_026$V1

# Generate a vector of threshold values to iterate over
cutoff <- 0:5
```

```r
# Generate a vector of V1 means at each V1 threshold
cor_026V1.shift <- sapply(cutoff, function(x){mean(cor_026V1[cor_026V1 > x])})

# Calculate deviation
(cor_026V1.df <- data.frame(time = 'V1',
                            cutoff = cutoff,
                            cutoff2 = cutoff - 0.15, # Offset for plotting purposes
                            mean = cor_026V1.shift) %>%
        mutate(deviation = mean - mean(cor_026V1),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean  deviation
## 1   V1      0   -0.15 6.094128 0.00000000
## 2   V1      1    0.85 6.109456 0.01532837
## 3   V1      2    1.85 6.122634 0.02850584
## 4   V1      3    2.85 6.222266 0.12813776
## 5   V1      4    3.85 6.446603 0.35247524
## 6   V1      5    4.85 6.861059 0.76693044
```

```r
# Plot data
ggplot(data = cor_026V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_026V1), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'A: Shift in V1 mean with increasing V1 inclusion threshold',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



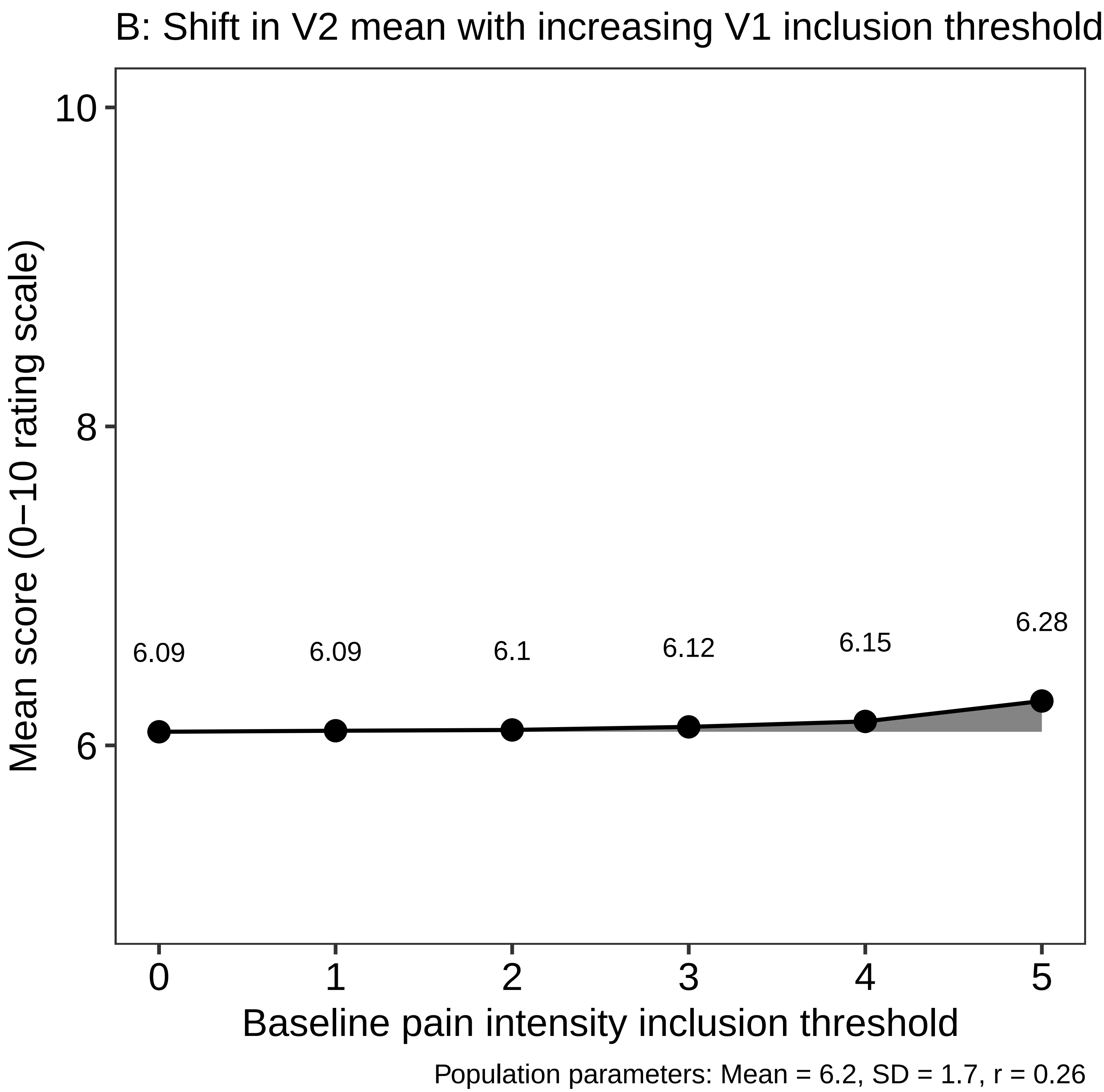
3.2.2 Model mean of V2 with increasing inclusion thresholds from 0 to 5

```
# Extract visit 2 data
cor_026V2 <- cor_026$V2

# Generate a vector of threshold values to iterate over
cutoff <- 0:5

# Generate a vector of V2 means at each V1 threshold
cor_026V2.shift <- map_dbl(.x = cutoff,
                           ~ cor_026 %>%
                             filter(V1 > .x) %>%
                             .$V2 %>%
```

```r
                              mean(.))

# Calculate deviation
(cor_026V2.df <- data.frame(time = 'V2',
                           cutoff = cutoff,
                           cutoff2 = cutoff + 0.15, # Offset for plotting purposes
                           mean = cor_026V2.shift) %>%
        mutate(deviation = mean - mean(cor_026V2),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean   deviation
## 1   V2      0    0.15 6.085393 0.000000000
## 2   V2      1    1.15 6.091863 0.006470213
## 3   V2      2    2.15 6.096795 0.011401991
## 4   V2      3    3.15 6.116158 0.030765461
## 5   V2      4    4.15 6.150265 0.064872588
## 6   V2      5    5.15 6.278327 0.192934328
```

```r
# Plot data
ggplot(data = cor_026V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_026V2), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'B: Shift in V2 mean with increasing V1 inclusion threshold',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



3.3 Distributional shifts caused by having a threshold

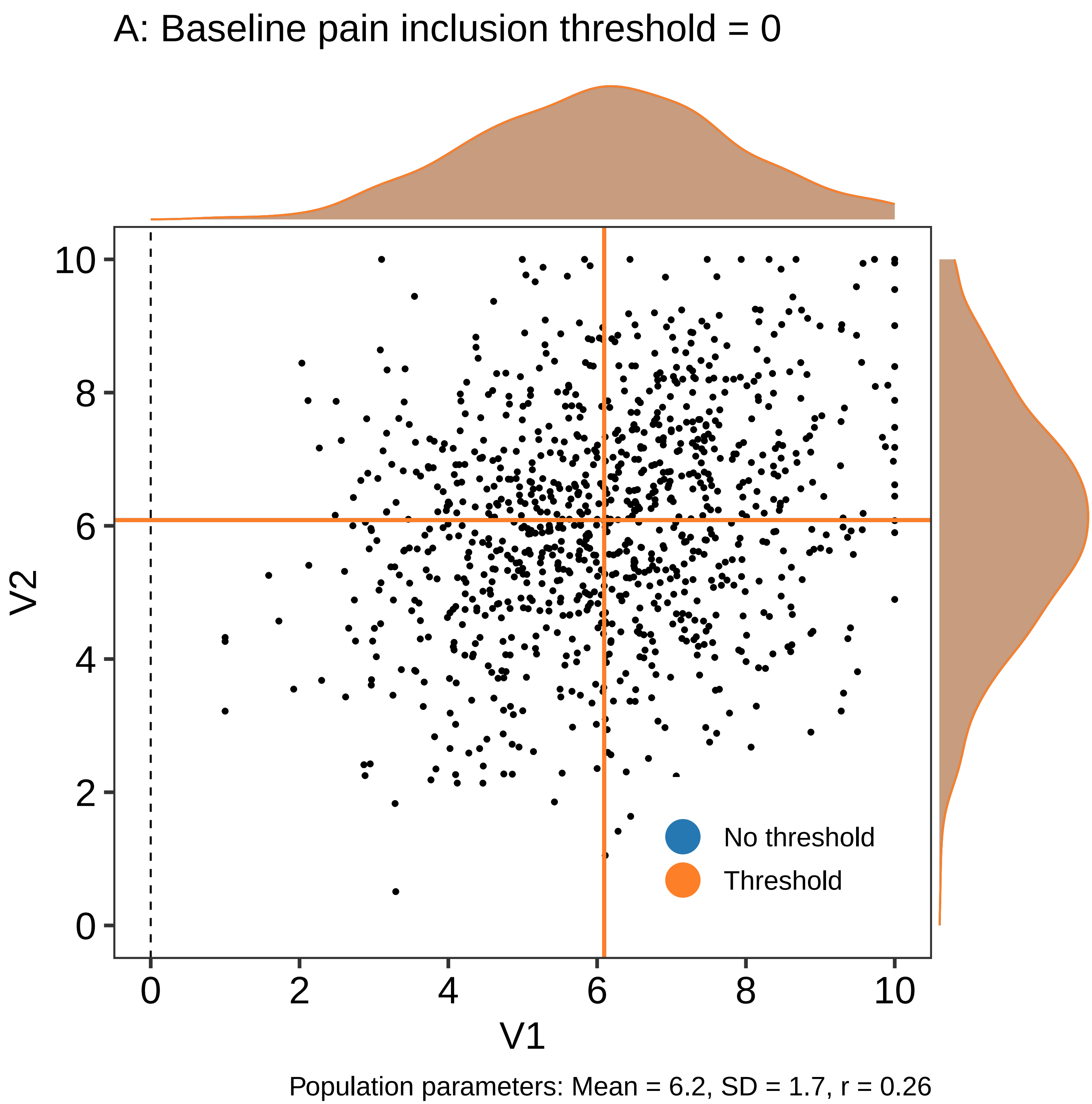
3.3.1 Threshold: 0

```
# Process data
placebo_1.0 <- cor_026 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.0 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.0,
```

```r
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.0$.id <- 0

kable(diff_1.0)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.00874	0.95	-0.126	0.137

```r
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(cor_026) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_026,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_026$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(cor_026$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.0$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'A: Baseline pain inclusion threshold = 0',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



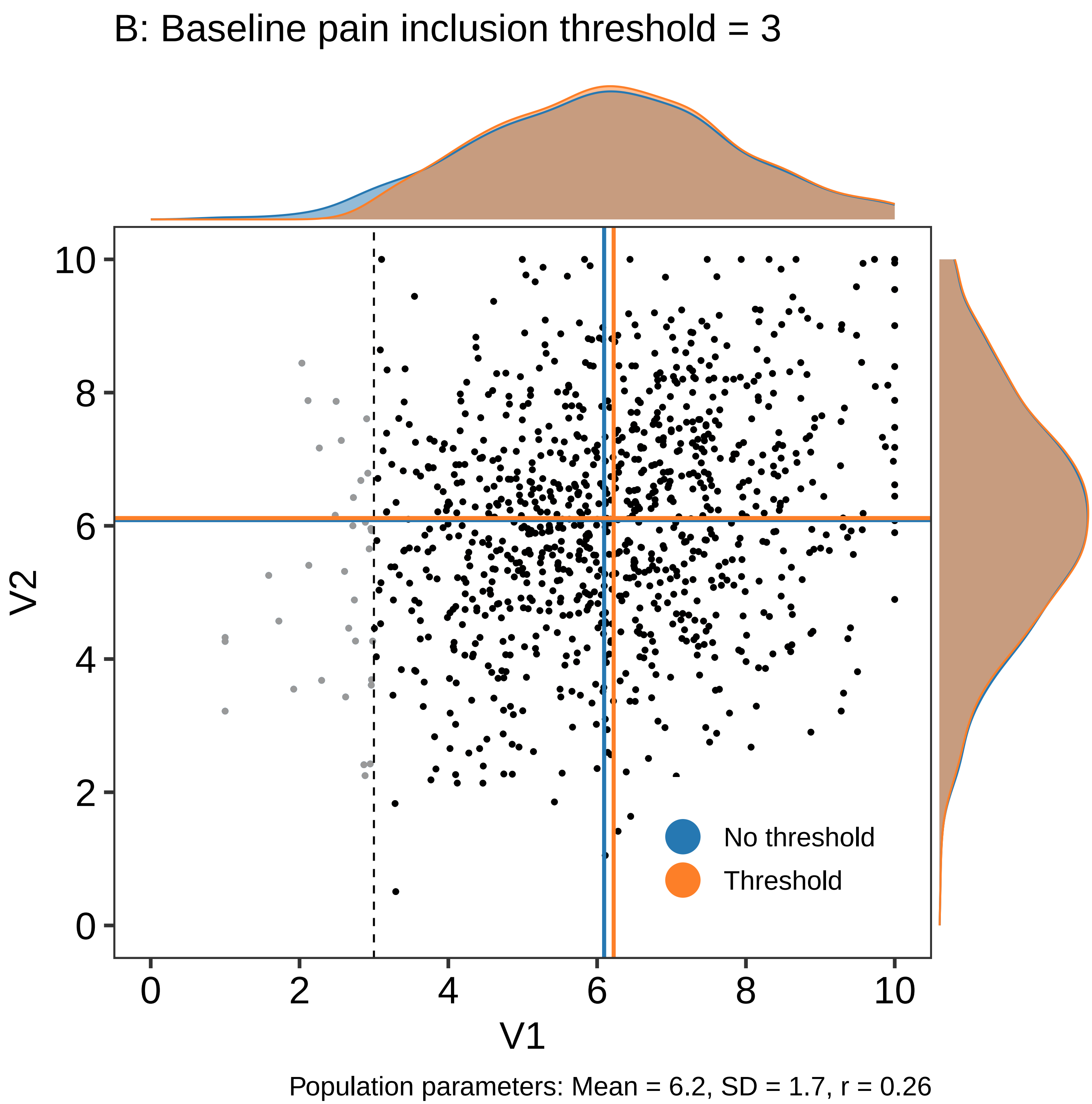
3.3.2 Threshold: 3

```
# Process data
placebo_1.3 <- cor_026 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.3 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.3,
                          R = 2000,
```

```r
                          traditional = FALSE,
                          bca = TRUE)

diff_1.3$.id <- 3

kable(diff_1.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	966	0.106	0.95	-0.0291	0.233

```r
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(cor_026) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_026,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_026$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(cor_026$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.3$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'B: Baseline pain inclusion threshold = 3',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



3.3.3 Threshold: 4

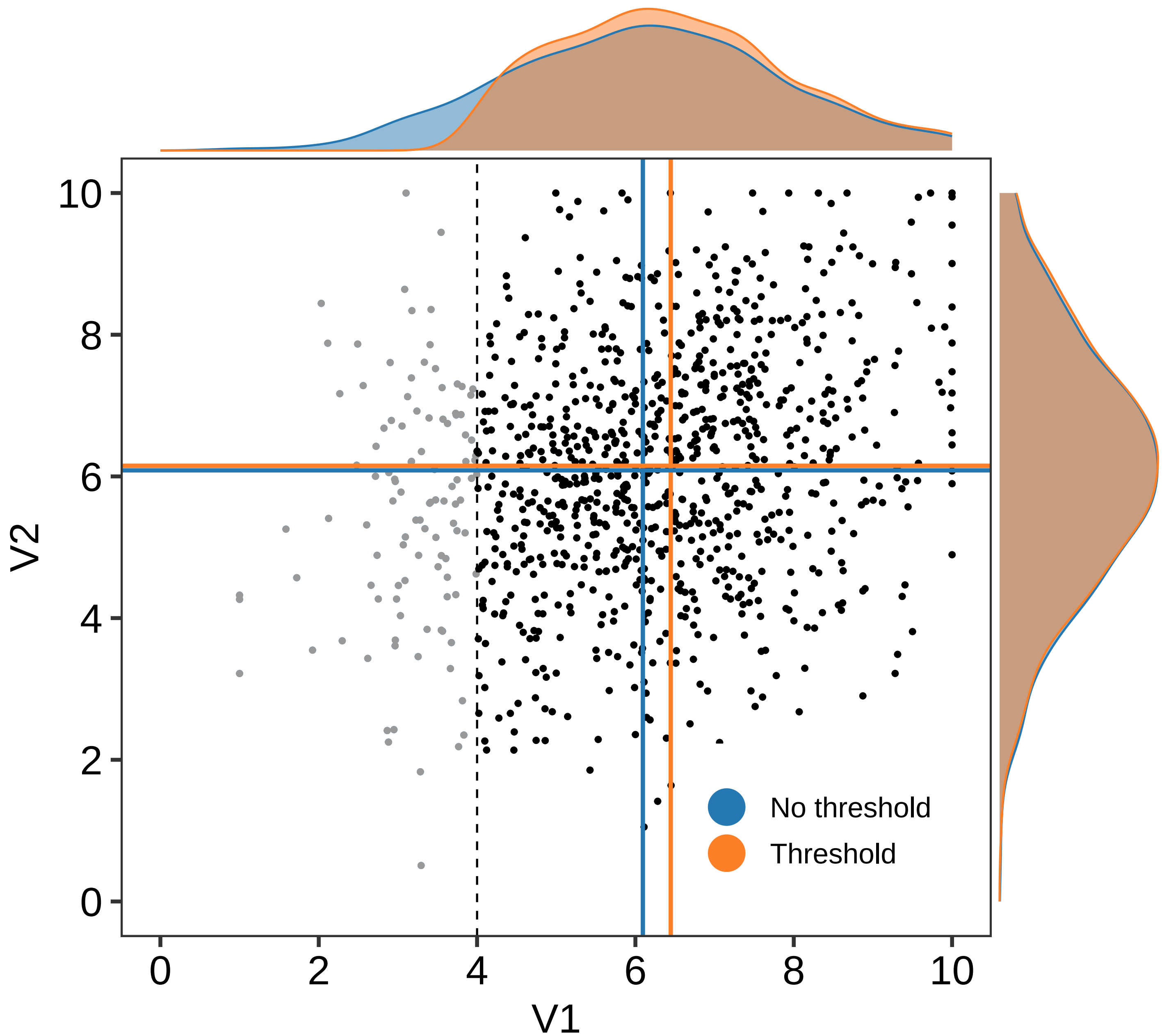
```
# Process that data
placebo_1.4 <- cor_026 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```r
diff_1.4 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.4,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.4$.id <- 4

kable(diff_1.4)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
4	892	0.296	0.95	0.163	0.417

```r
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
               bind_rows(cor_026) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_026,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_026$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(cor_026$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.4$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'C: Baseline pain inclusion threshold = 4',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

C: Baseline pain inclusion threshold = 4



Population parameters: Mean = 6.2, SD = 1.7, r = 0.26

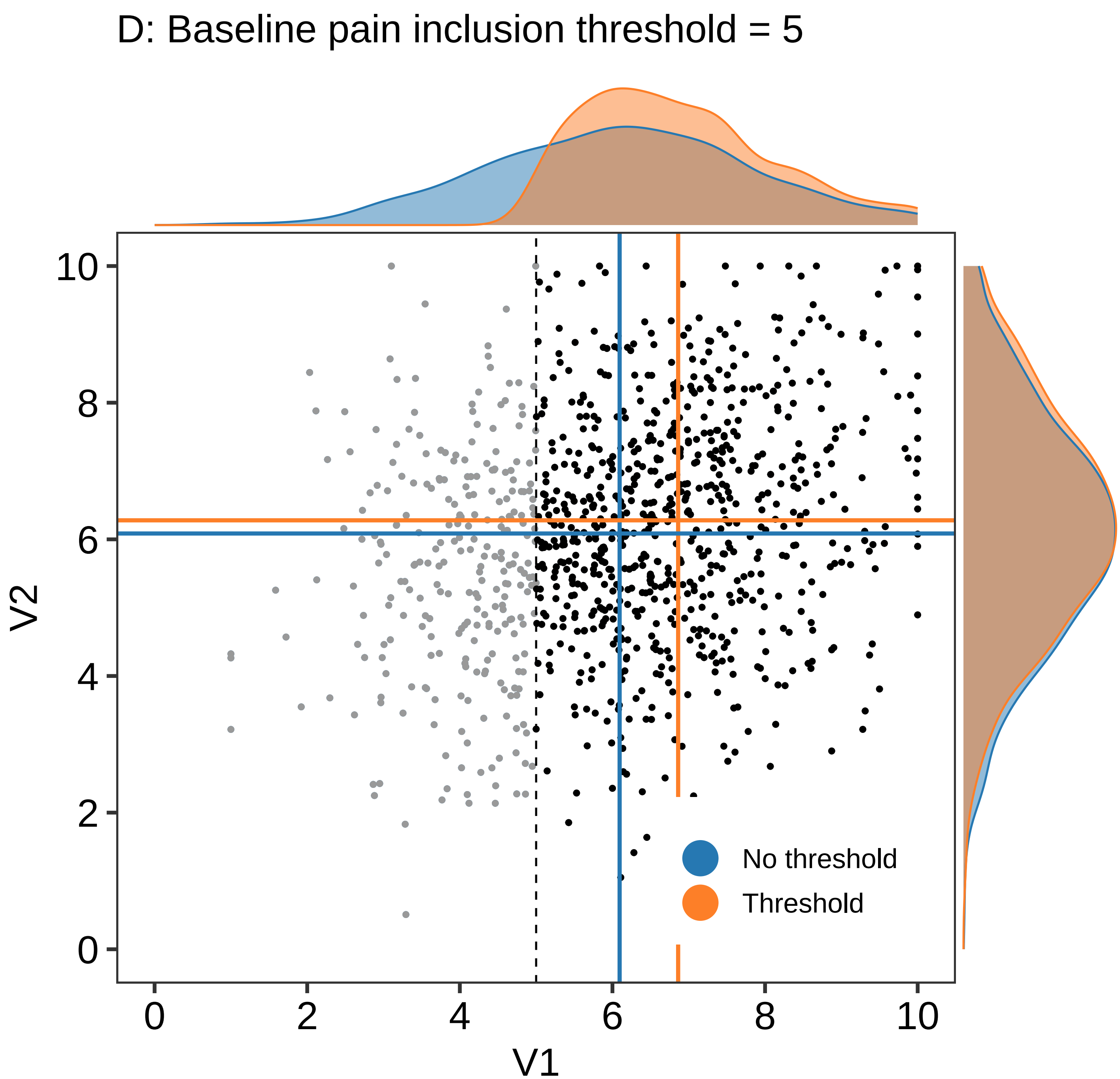
3.3.4 Threshold: 5

```
# Process that data
placebo_1.5 <- cor_026 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```r
diff_1.5 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.5,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.5$.id <- 5

kable(diff_1.5)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	734	0.583	0.95	0.454	0.723

```r
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(cor_026) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_026,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_026$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(cor_026$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.5$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'D: Baseline pain inclusion threshold = 5',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

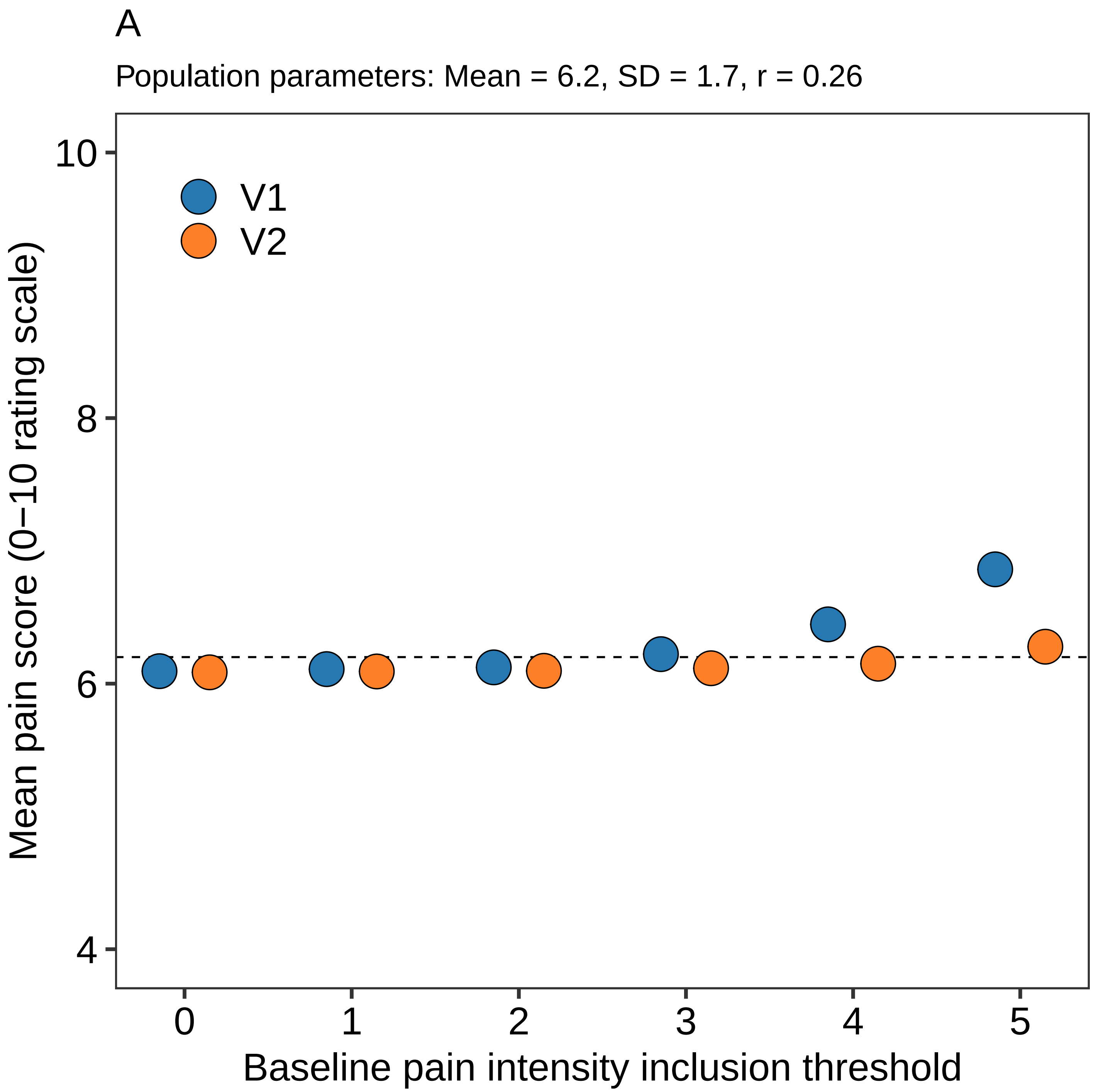


Population parameters: Mean = 6.2, SD = 1.7, r = 0.26

3.4 Summary plots

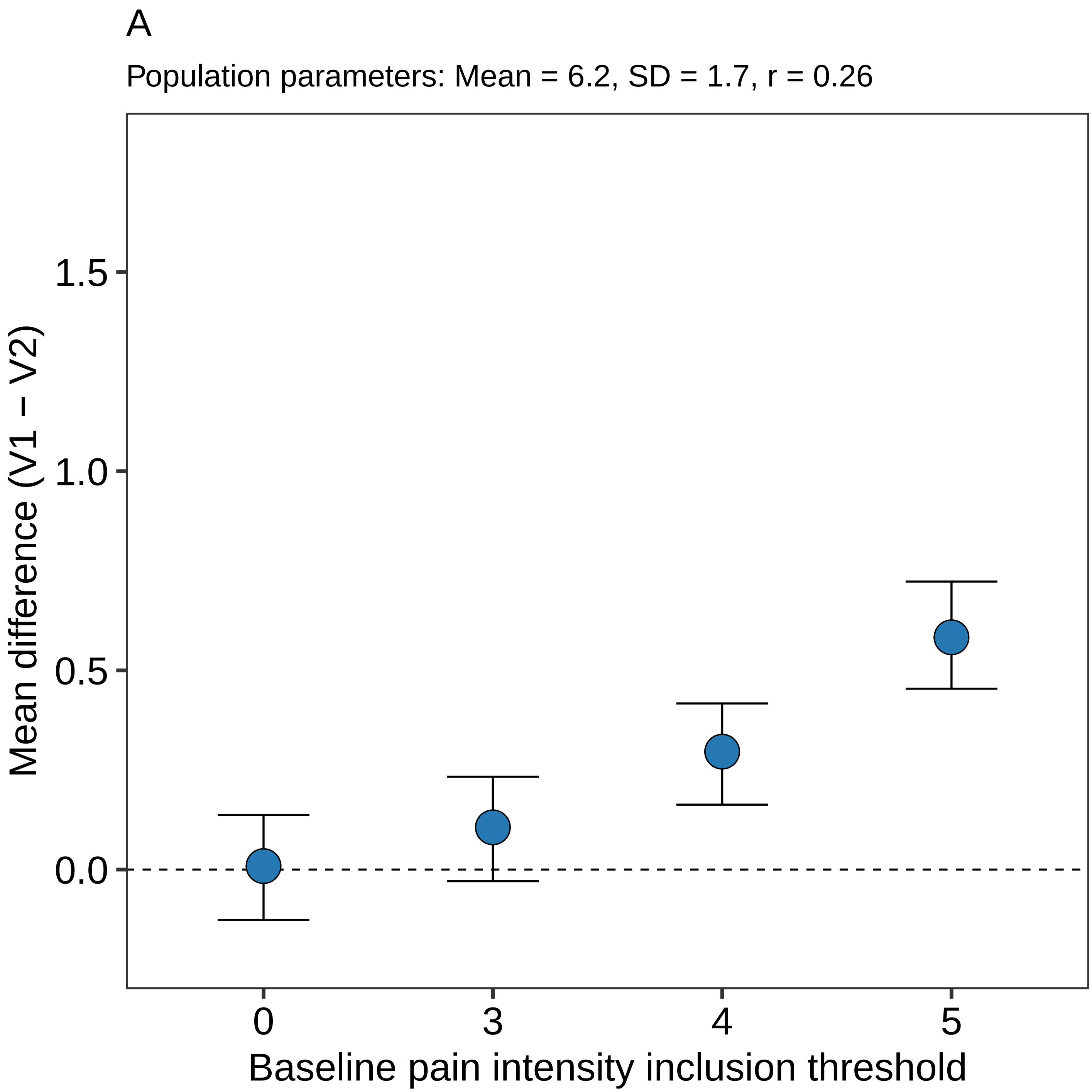
```
# Plot 1
shift_1 <- cor_026V1.df %>%
  bind_rows(cor_026V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
      x = cutoff2,
      fill= time) +
  geom_hline(yintercept = 6.2,
             linetype = 2) +
  geom_point(shape = 21,
```

```
             size = 8) +
  labs(title = 'A',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean pain score (0-10 rating scale)') +
  scale_x_continuous(breaks = 0:5) +
  scale_y_continuous(limits = c(4, 10)) +
  scale_fill_manual(values = pal) +
  theme(legend.title = element_blank(),
        legend.position = c(0.12, 0.89),
        legend.text = element_text(size = 20)); shift_1
```




```
# Plot 2
# Bind diff_*.* dataframes
```

```r
diff_all_1 <- diff_1.0 %>%
  bind_rows(diff_1.3, diff_1.4, diff_1.5)

pp_1 <- diff_all_1 %>%
  mutate(Threshold = factor(.id)) %>%
  ggplot(data = .) +
  aes(x = Threshold,
      y = Mean,
      ymin = Bca.lower,
      ymax = Bca.upper) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_errorbar(width = 0.4) +
  geom_point(shape = 21,
             fill = pal[[1]],
             size = 8) +
  labs(title = 'A',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.26',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_1
```

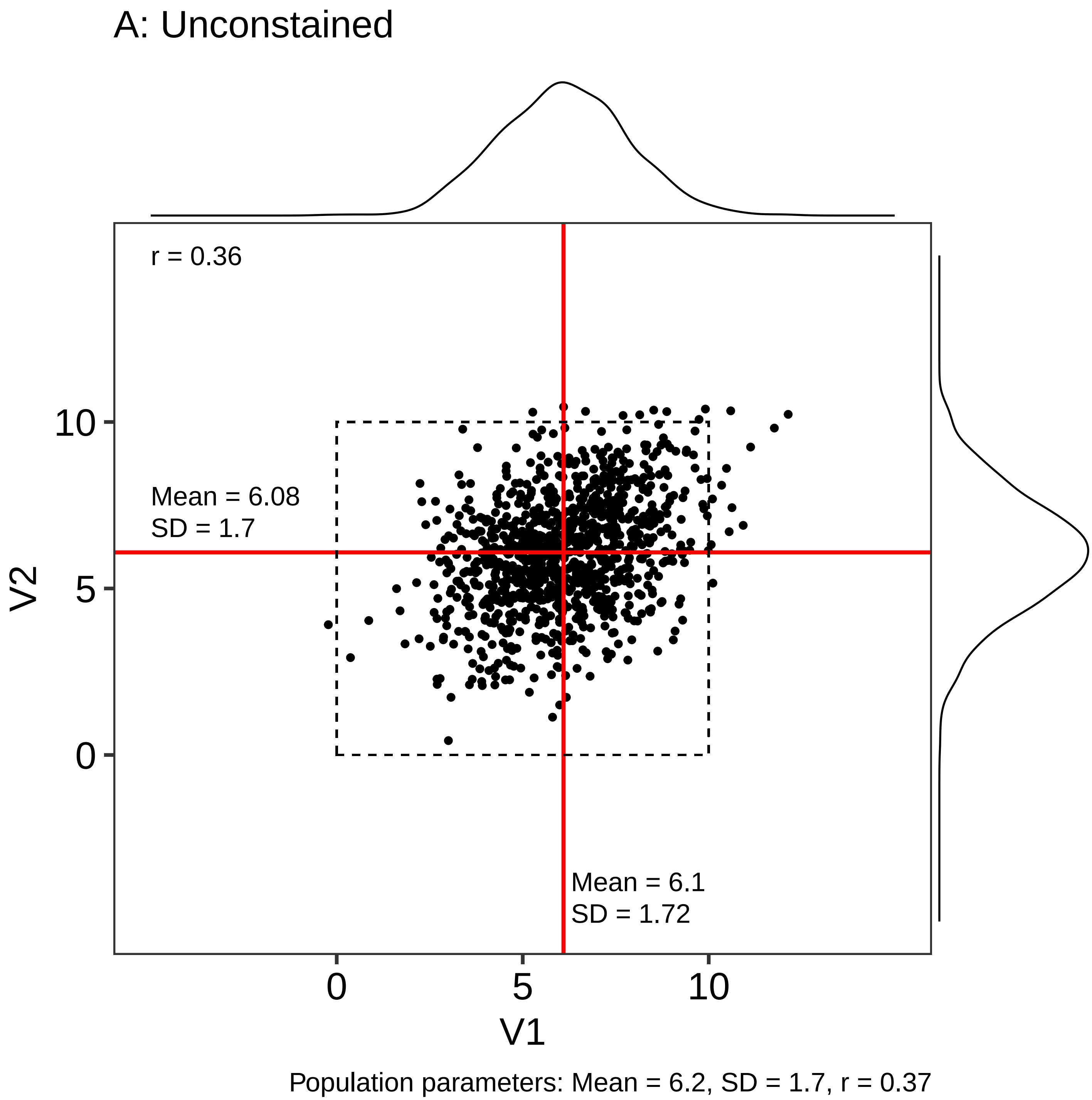

A
Population parameters: Mean = 6.2, SD = 1.7, r = 0.26
Mean difference (V1 − V2)
1.5
1.0
0.5
0.0
0
3
4
5
Baseline pain intensity inclusion threshold

4 Correlation: 0.37

4.1 Generate and summarise data

4.1.1 Unconstrained data

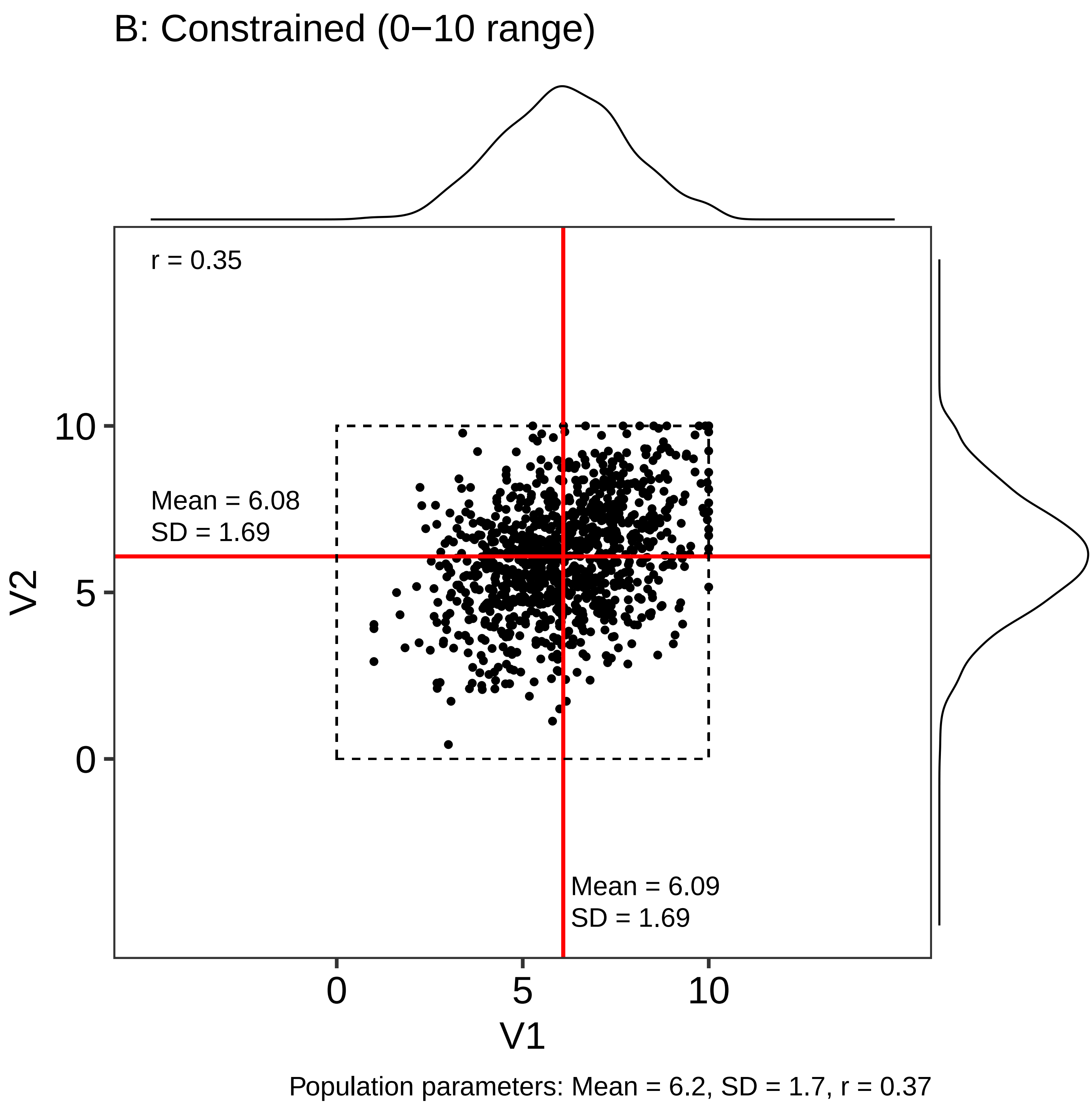
```
# Set the random seed for reproducibility
set.seed(2019)

# Generate the data
cor_037.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_037))

# Plot unconstrained data
ggMarginal(ggplot(data = cor_037.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_037.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_037.base$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_037.base$V1,
                                         cor_037.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_037.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_037.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_037.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_037.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_037.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_037.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_037.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_037.base$V1), 2)}")) +
               labs(title = 'A: Unconstained',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



4.1.2 Constrained data

```
# Constrain data
cor_037 <- cor_037.base %>%
    mutate(V1 = case_when(
                V1 < 1 ~ 1,
                V1 > 10 ~ 10,
                TRUE ~ V1)) %>%
    mutate(V2 = case_when(
                V2 < 0 ~ 0,
                V2 > 10 ~ 10,
                TRUE ~ V2)) %>%
```

```r
    mutate(group = 'No threshold')

# Plot constrained data
ggMarginal(ggplot(data = cor_037) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_037$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_037$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_037$V1,
                                         cor_037$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_037$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_037$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_037$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_037$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_037$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_037$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_037$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_037$V1), 2)}")) +
               labs(title = 'B: Constrained (0-10 range)',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



4.2 Effect of having a threshold on mean pain intensity scores

Constrained data only

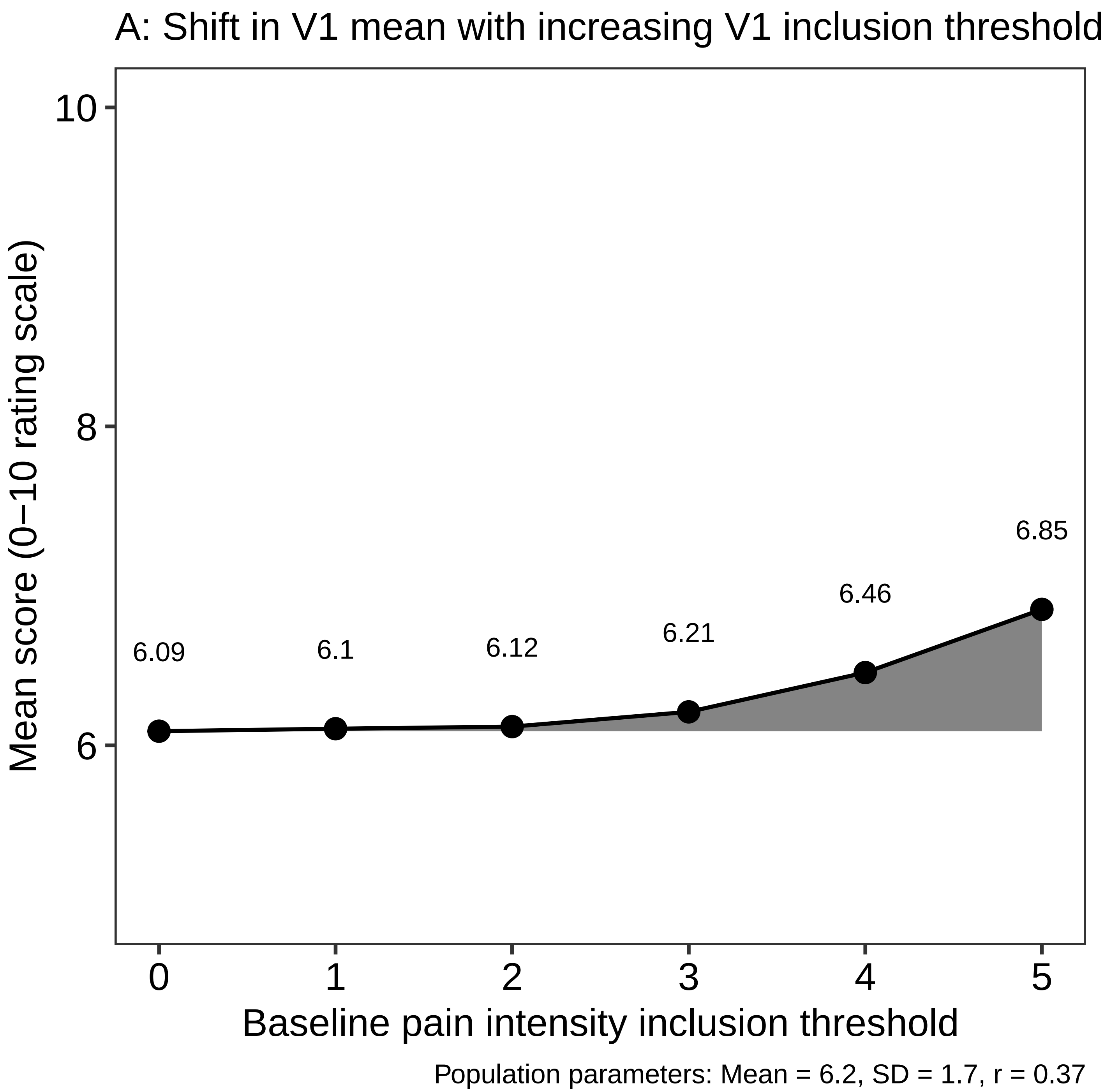
4.2.1 Model mean of V1 with increasing pain inclusion thresholds from 0 to 5

```
# Extract visit 1 data
cor_037V1 <- cor_037$V1

# Generate a vector of threshold values to iterate over
cutoff <- 0:5
```

```r
# Generate a vector of V1 means at each V1 threshold
cor_037V1.shift <- sapply(cutoff, function(x){mean(cor_037V1[cor_037V1 > x])})

# Calculate deviation
(cor_037V1.df <- data.frame(time = 'V1',
                            cutoff = cutoff,
                            cutoff2 = cutoff - 0.15, # Offset for plotting purposes
                            mean = cor_037V1.shift) %>%
        mutate(deviation = mean - mean(cor_037V1),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean deviation
## 1   V1      0   -0.15 6.089086 0.0000000
## 2   V1      1    0.85 6.104399 0.0153132
## 3   V1      2    1.85 6.117645 0.0285585
## 4   V1      3    2.85 6.210219 0.1211326
## 5   V1      4    3.85 6.456642 0.3675562
## 6   V1      5    4.85 6.852977 0.7638908
```

```r
# Plot data
ggplot(data = cor_037V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_037V1), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'A: Shift in V1 mean with increasing V1 inclusion threshold',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



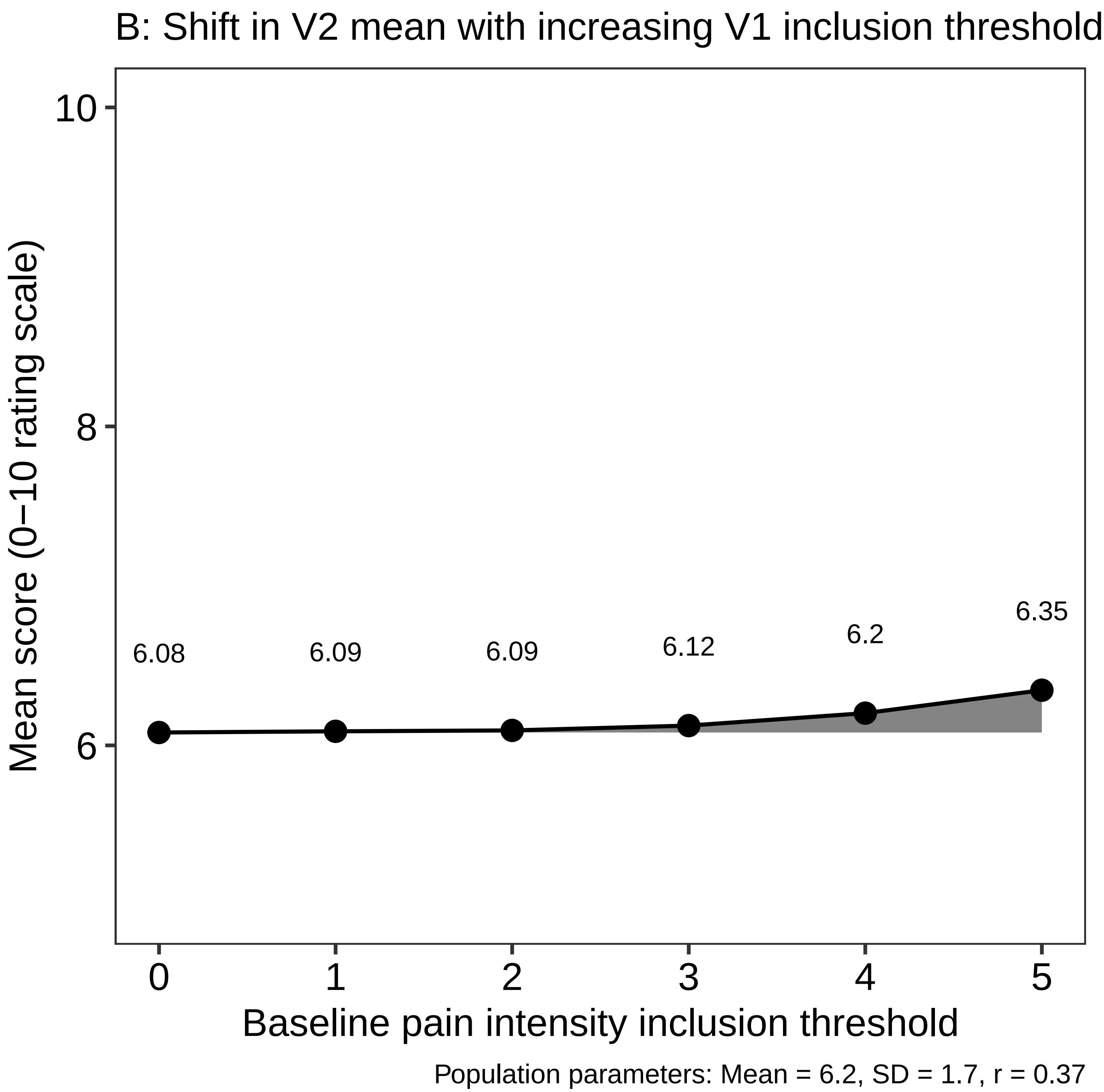
4.2.2 Model mean of V2 with increasing inclusion thresholds from 0 to 5

```
# Extract visit 2 data
cor_037V2 <- cor_037$V2

# Generate a vector of threshold values to iterate over
cutoff <- 0:5

# Generate a vector of V2 means at each V1 threshold
cor_037V2.shift <- map_dbl(.x = cutoff,
                           ~ cor_037 %>%
                             filter(V1 > .x) %>%
                             .$V2 %>%
```

```r
                              mean(.))

# Calculate deviation
(cor_037V2.df <- data.frame(time = 'V2',
                            cutoff = cutoff,
                            cutoff2 = cutoff + 0.15, # Offset for plotting purposes
                            mean = cor_037V2.shift) %>%
        mutate(deviation = mean - mean(cor_037V2),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean   deviation
## 1   V2      0    0.15 6.080904 0.000000000
## 2   V2      1    1.15 6.088296 0.007391845
## 3   V2      2    2.15 6.093934 0.013029342
## 4   V2      3    3.15 6.124168 0.043263676
## 5   V2      4    4.15 6.202257 0.121352537
## 6   V2      5    5.15 6.346071 0.265166275
```

```r
# Plot data
ggplot(data = cor_037V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_037V2), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'B: Shift in V2 mean with increasing V1 inclusion threshold',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



4.3 Distributional shifts caused by having a threshold

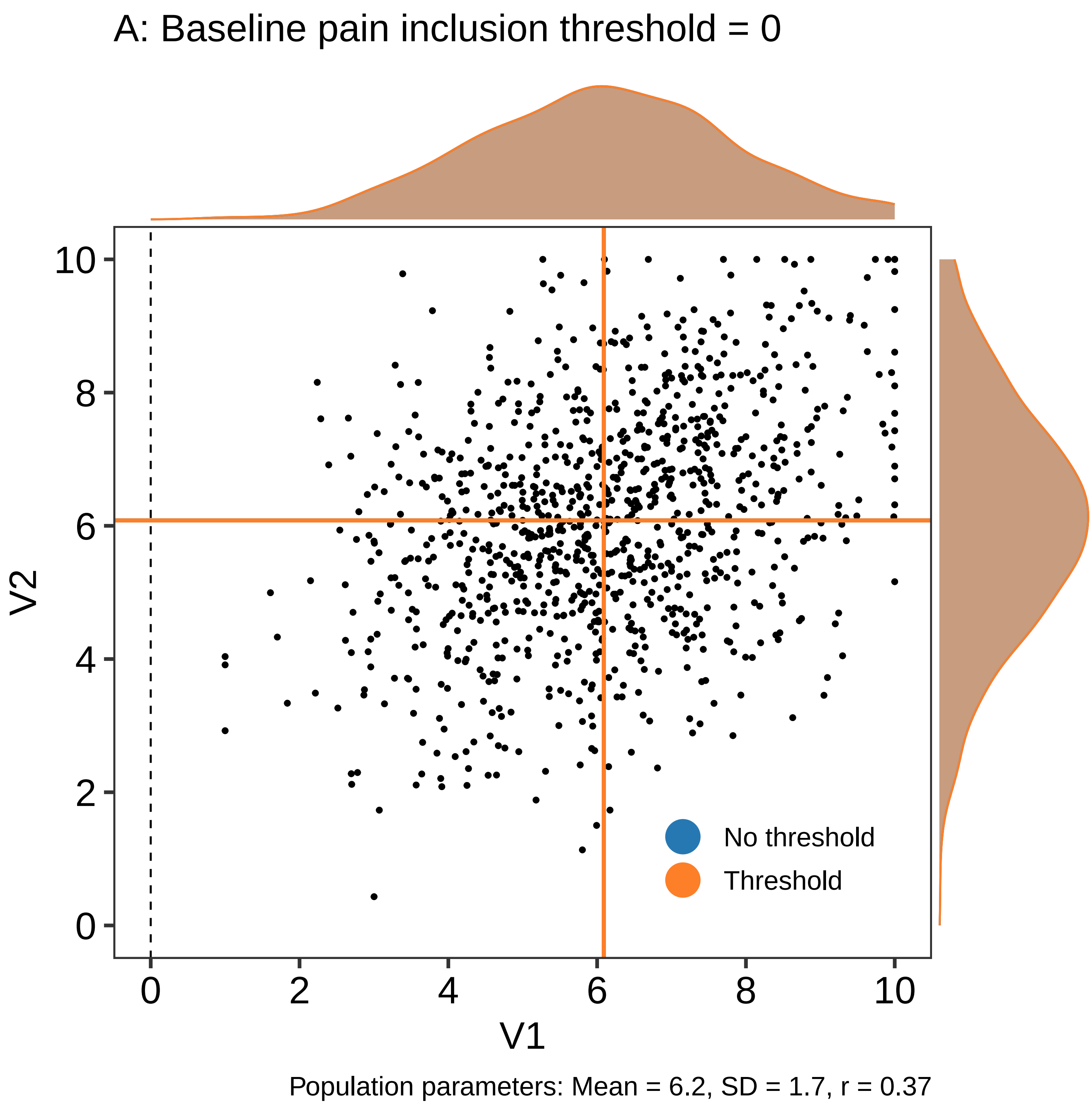
4.3.1 Threshold: 0

```
# Process data
placebo_1.0 <- cor_037 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.0 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.0,
```

```
                            R = 2000,
                            traditional = FALSE,
                            bca = TRUE)

diff_1.0$.id <- 0

kable(diff_1.0)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.00818	0.95	-0.116	0.126

```
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(cor_037) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_037,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_037$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(cor_037$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.0$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'A: Baseline pain inclusion threshold = 0',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



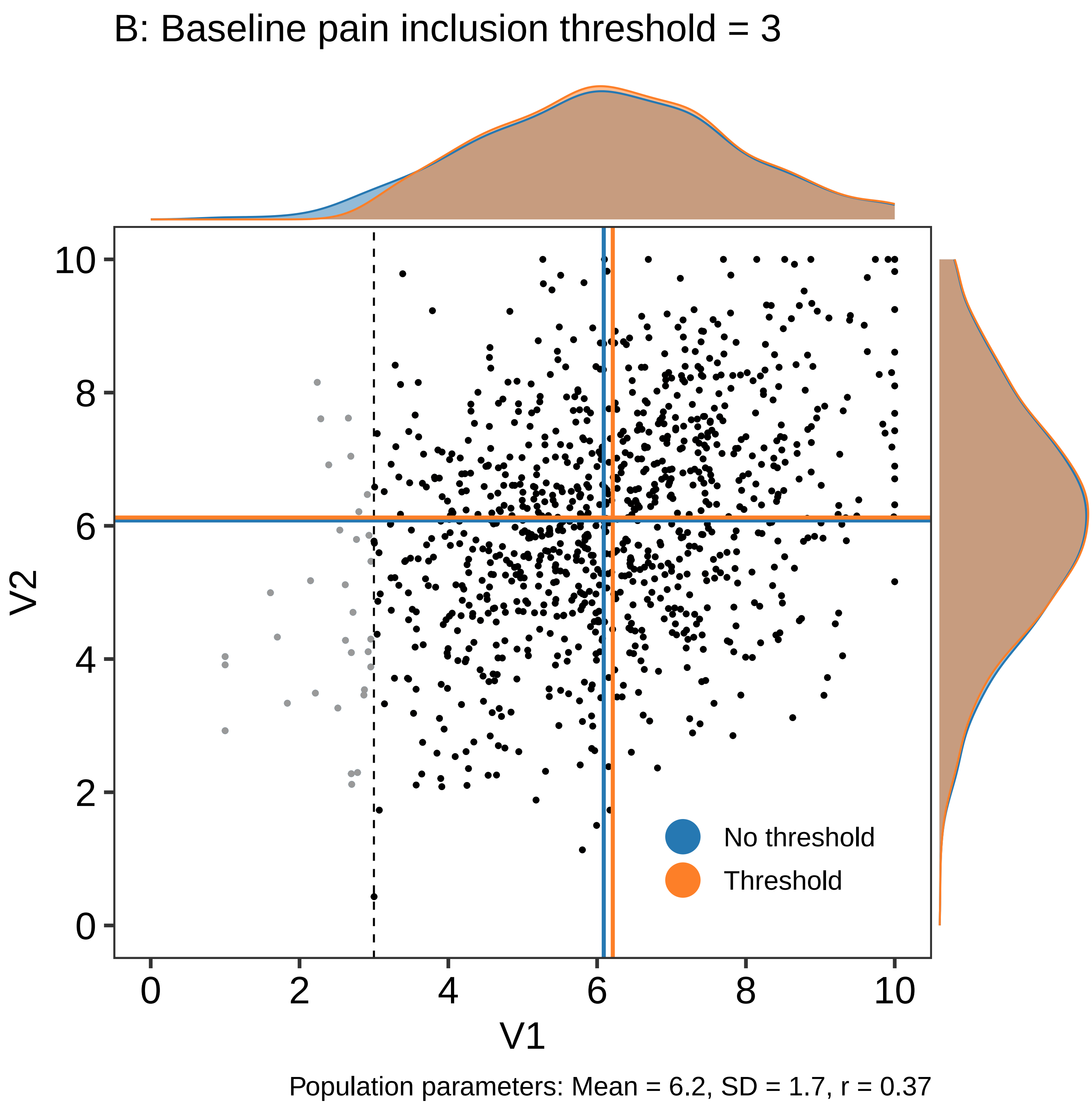
4.3.2 Threshold: 3

```
# Process data
placebo_1.3 <- cor_037 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.3 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.3,
                          R = 2000,
```

```r
                            traditional = FALSE,
                            bca = TRUE)

diff_1.3$.id <- 3

kable(diff_1.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	968	0.0861	0.95	-0.0351	0.207

```r
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(cor_037) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_037,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_037$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(cor_037$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.3$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'B: Baseline pain inclusion threshold = 3',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



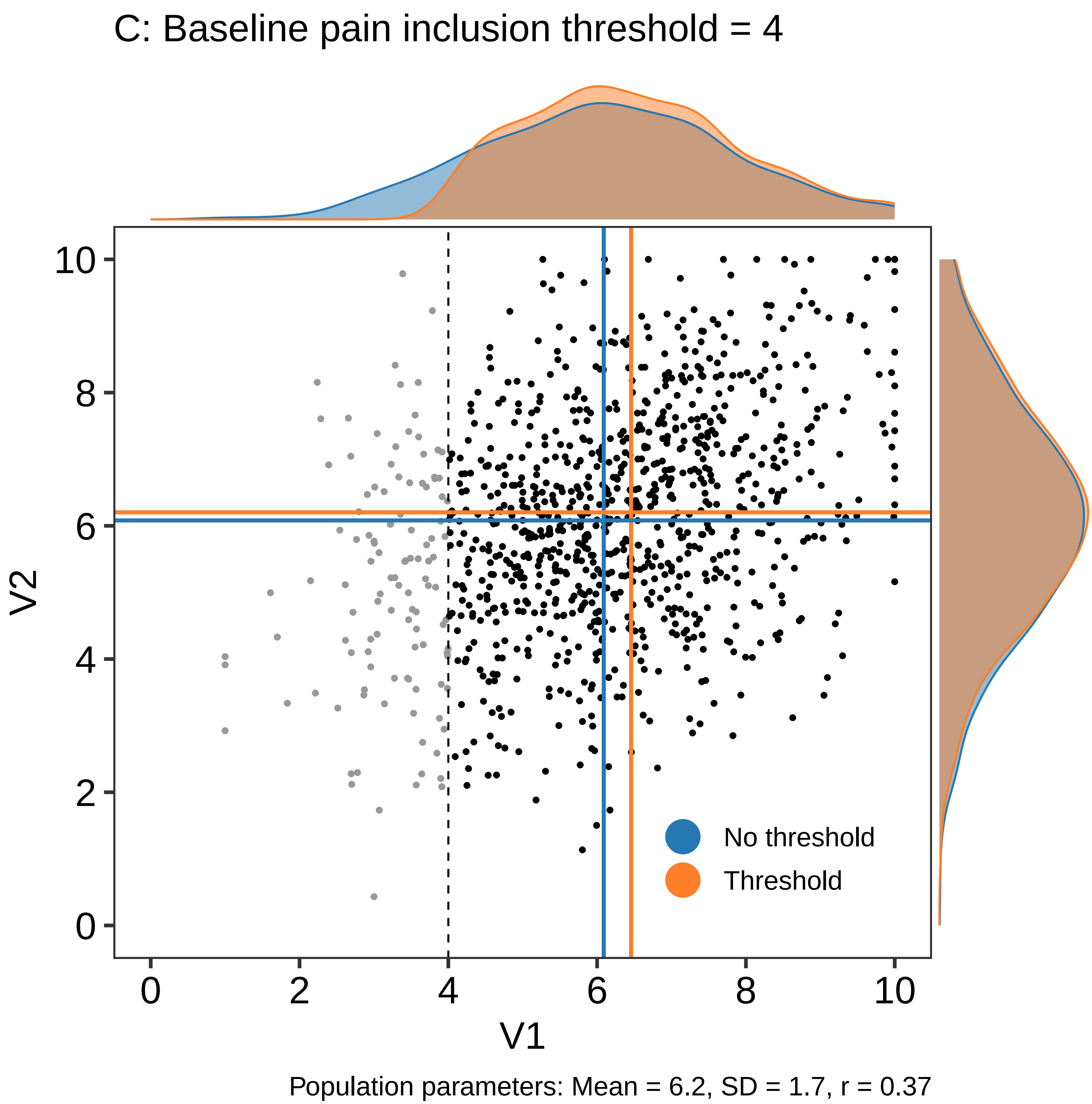
4.3.3 Threshold: 4

```
# Process that data
placebo_1.4 <- cor_037 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```
diff_1.4 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.4,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.4$.id <- 4

kable(diff_1.4)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
4	886	0.254	0.95	0.132	0.373

```
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
               bind_rows(cor_037) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_037,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_037$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(cor_037$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.4$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'C: Baseline pain inclusion threshold = 4',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



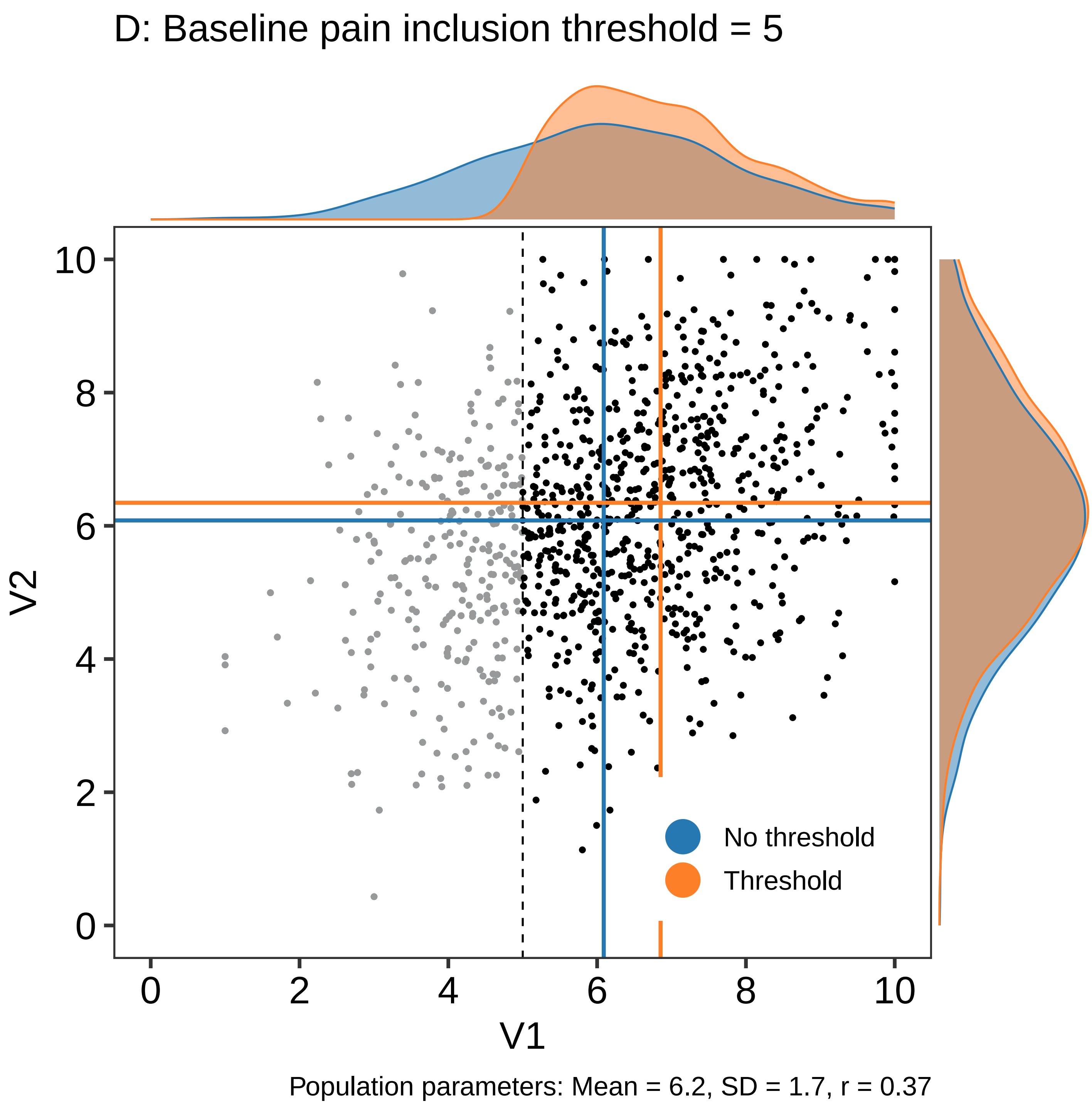
4.3.4 Threshold: 5

```
# Process that data
placebo_1.5 <- cor_037 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```r
diff_1.5 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.5,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.5$.id <- 5

kable(diff_1.5)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	735	0.507	0.95	0.379	0.625

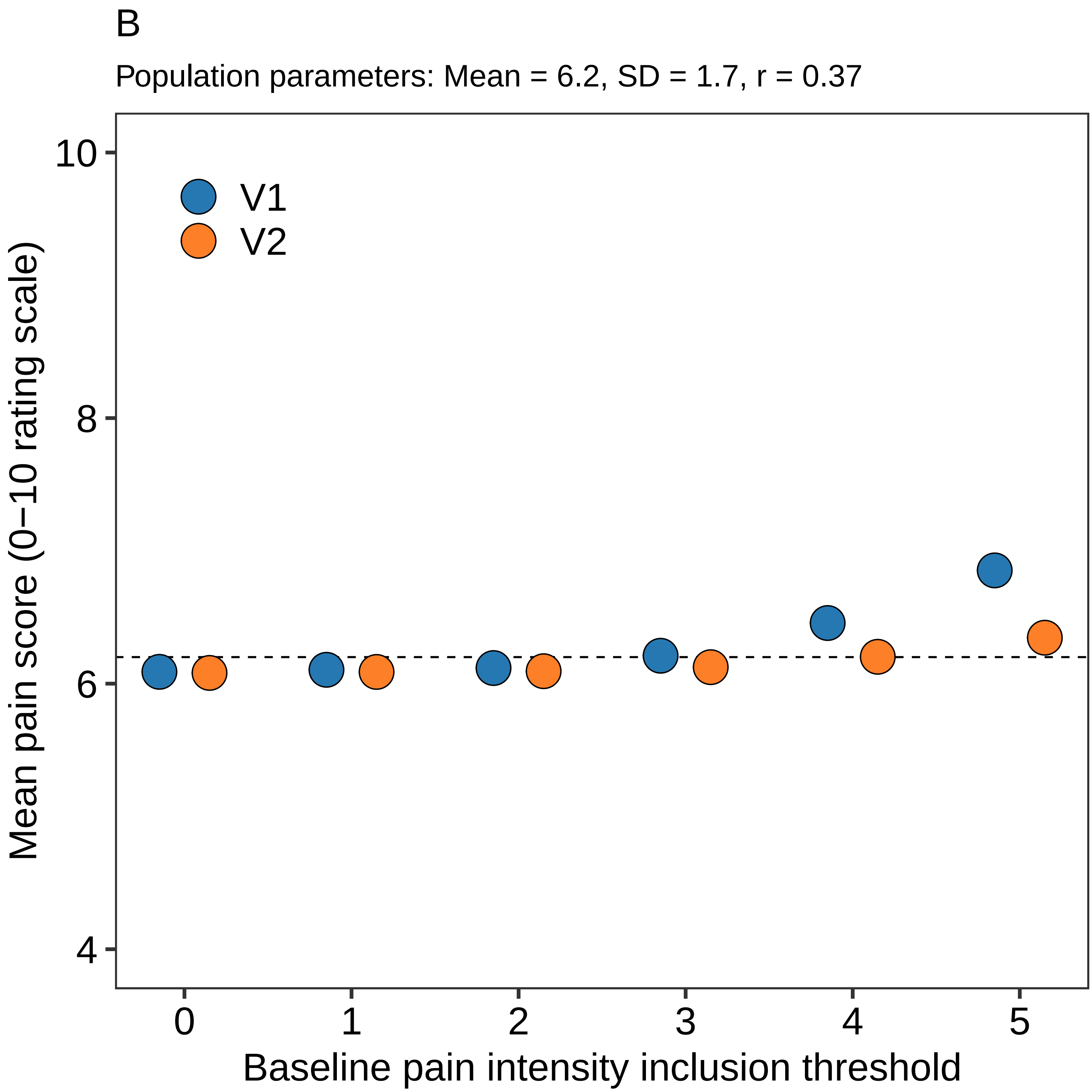
```r
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(cor_037) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_037,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_037$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(cor_037$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.5$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'D: Baseline pain inclusion threshold = 5',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



4.4 Summary plots

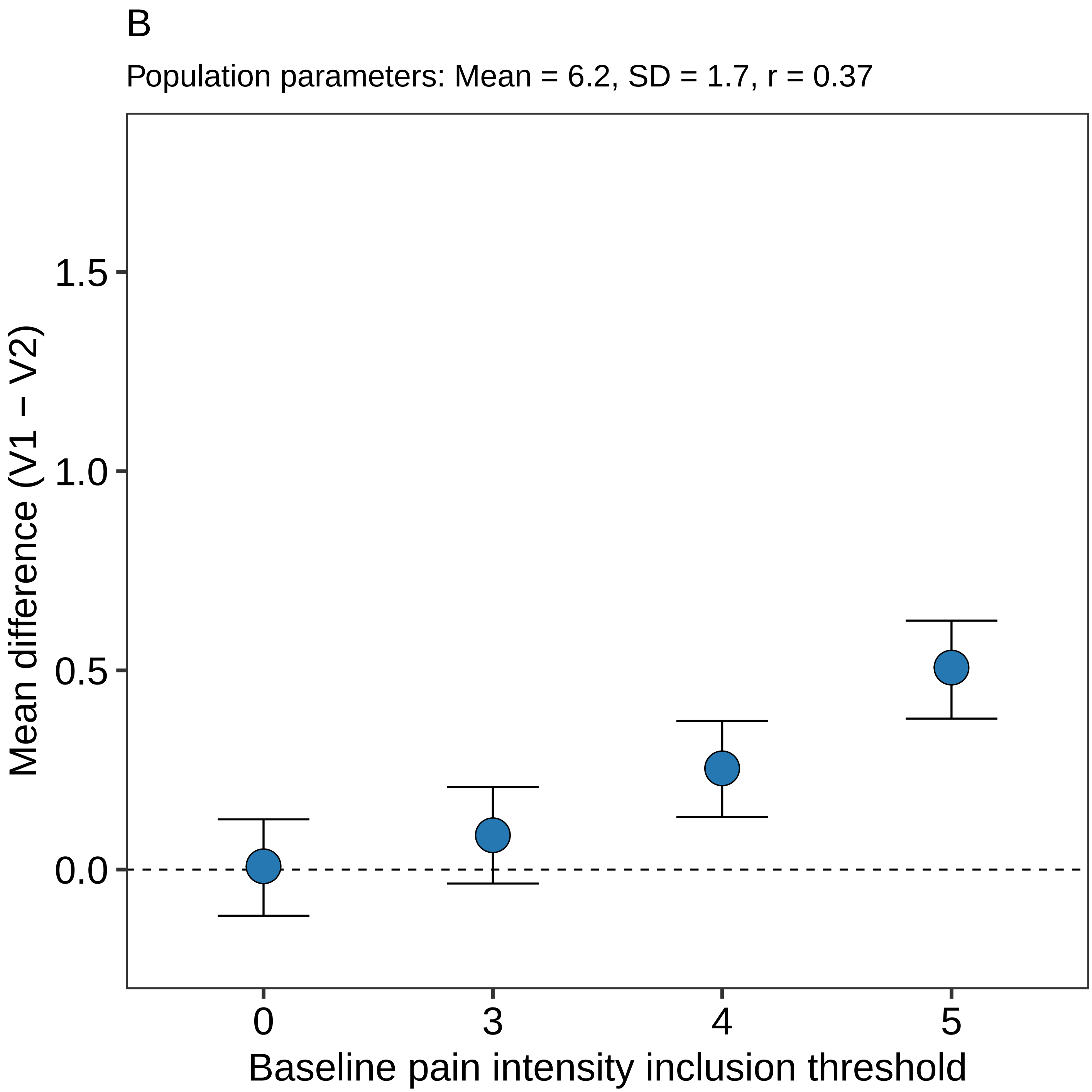
```
# Plot 1
shift_1 <- cor_037V1.df %>%
  bind_rows(cor_037V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
      x = cutoff2,
      fill= time) +
  geom_hline(yintercept = 6.2,
             linetype = 2) +
  geom_point(shape = 21,
```

```
              size = 8) +
  labs(title = 'B',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean pain score (0-10 rating scale)') +
  scale_x_continuous(breaks = 0:5) +
  scale_y_continuous(limits = c(4, 10)) +
  scale_fill_manual(values = pal) +
  theme(legend.title = element_blank(),
        legend.position = c(0.12, 0.89),
        legend.text = element_text(size = 20)); shift_1
```




```
# Plot 2
# Bind diff_*.* dataframes
```

```
diff_all_1 <- diff_1.0 %>%
  bind_rows(diff_1.3, diff_1.4, diff_1.5)

pp_1 <- diff_all_1 %>%
  mutate(Threshold = factor(.id)) %>%
  ggplot(data = .) +
  aes(x = Threshold,
      y = Mean,
      ymin = Bca.lower,
      ymax = Bca.upper) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_errorbar(width = 0.4) +
  geom_point(shape = 21,
             fill = pal[[1]],
             size = 8) +
  labs(title = 'B',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.37',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_1
```

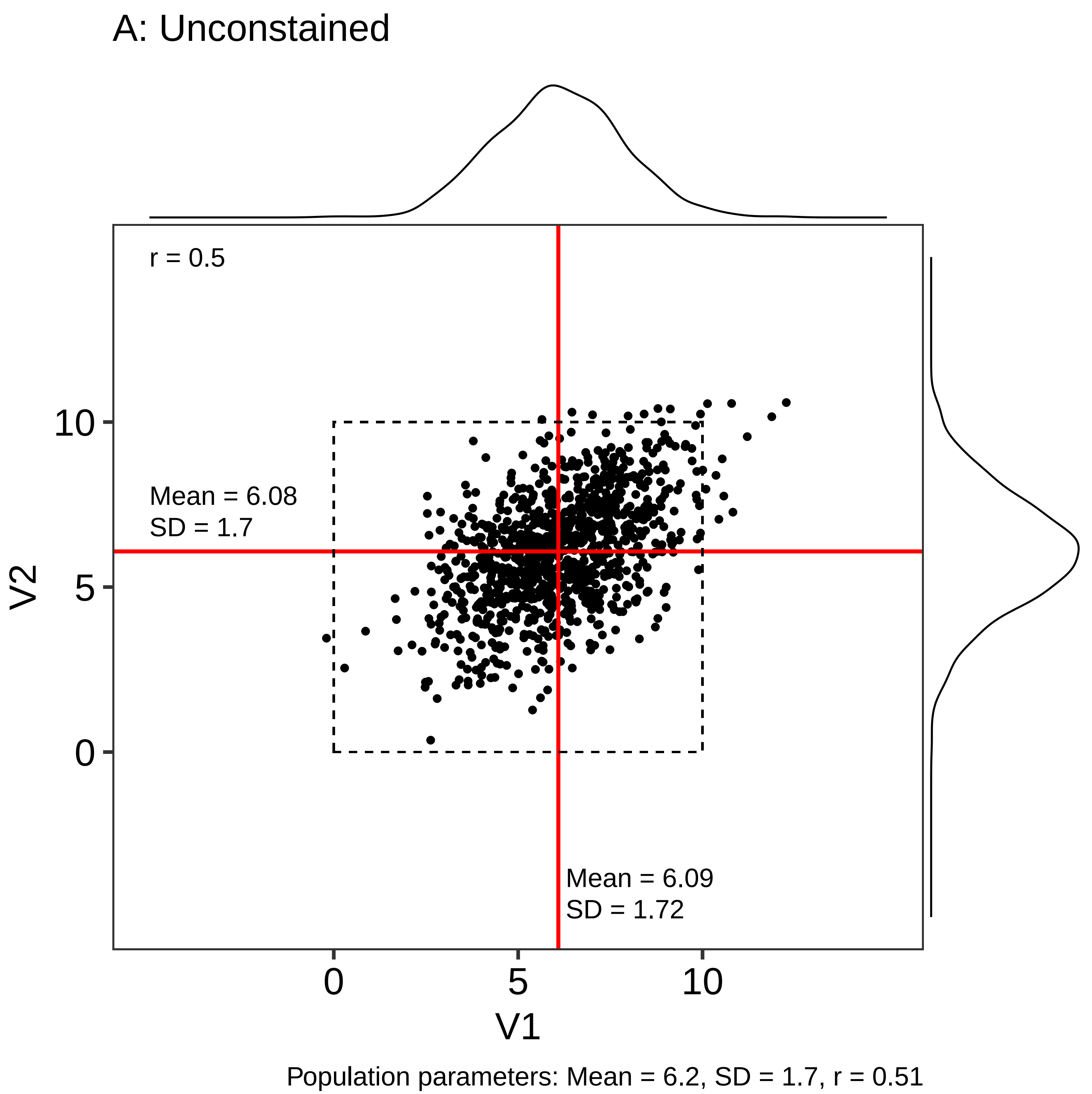

B
Population parameters: Mean = 6.2, SD = 1.7, r = 0.37
Mean difference (V1 – V2)
1.5
1.0
0.5
0.0
0
3
4
5
Baseline pain intensity inclusion threshold

5 Correlation: 0.51

5.1 Generate and summarise data

5.1.1 Unconstrained data

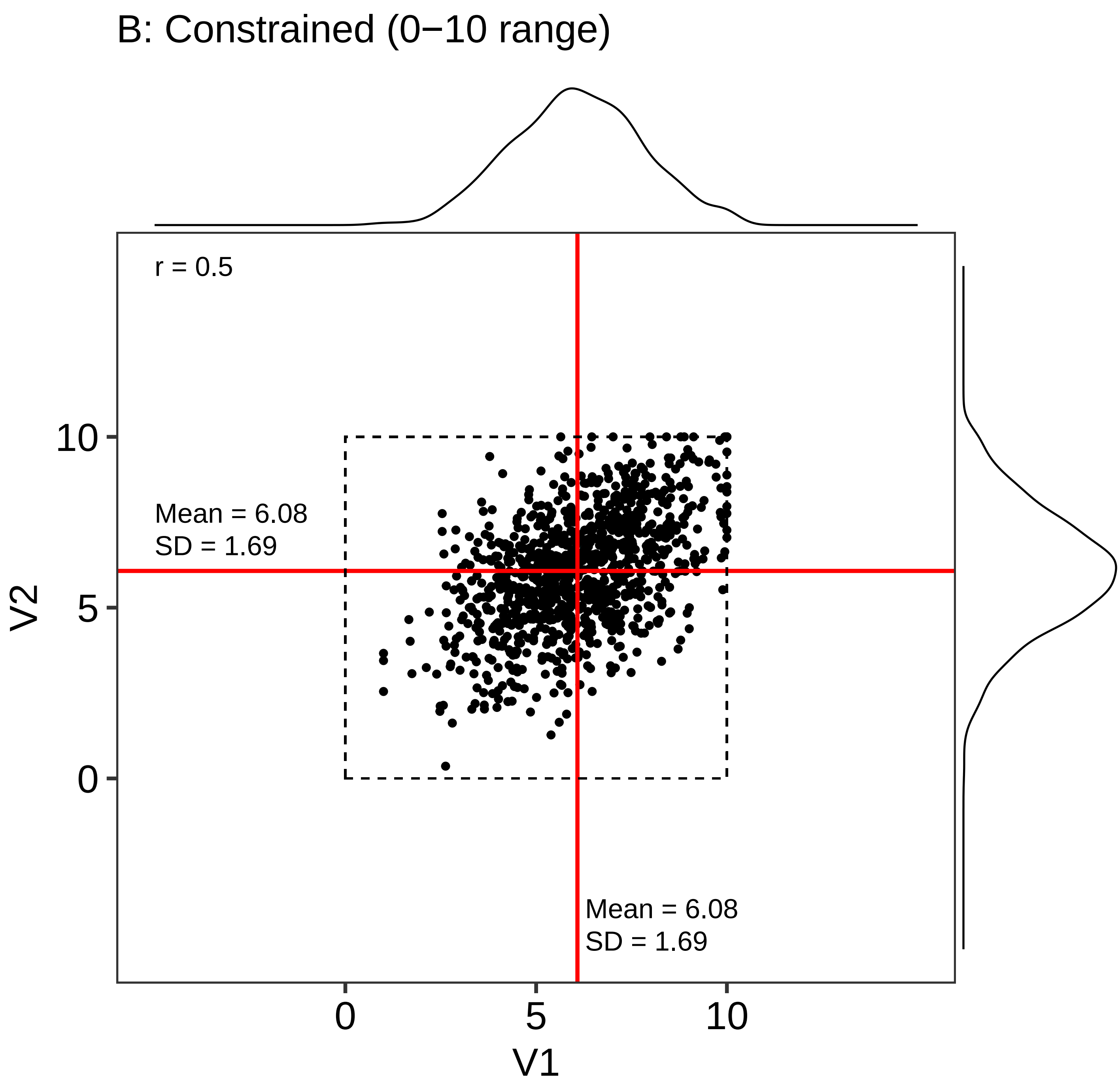
```
# Set the random seed for reproducibility
set.seed(2019)

# Generate the data
cor_051.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_051))

# Plot unconstrained data
ggMarginal(ggplot(data = cor_051.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_051.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_051.base$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_051.base$V1,
                                         cor_051.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_051.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_051.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_051.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_051.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_051.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_051.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_051.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_051.base$V1), 2)}")) +
               labs(title = 'A: Unconstained',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



5.1.2 Constrained data

```
# Constrain data
cor_051 <- cor_051.base %>%
    mutate(V1 = case_when(
                  V1 < 1 ~ 1,
                  V1 > 10 ~ 10,
                  TRUE ~ V1)) %>%
    mutate(V2 = case_when(
                  V2 < 0 ~ 0,
                  V2 > 10 ~ 10,
                  TRUE ~ V2)) %>%
```

```r
    mutate(group = 'No threshold')

# Plot constrained data
ggMarginal(ggplot(data = cor_051) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(cor_051$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(cor_051$V1),
                          colour = 'red', size = 1) +
               geom_rect(ymin = 0, ymax = 10,
                         xmin = 0, xmax = 10,
                         colour = '#000000',
                         alpha = 0,
                         linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0, size = 5,
                        label = str_glue("r = {round(cor(cor_051$V1,
                                         cor_051$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_051$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_051$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(cor_051$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_051$V2),2)}")) +
               annotate(geom = 'text', x = mean(cor_051$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(cor_051$V1), 2)}")) +
               annotate(geom = 'text', x = mean(cor_051$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(cor_051$V1), 2)}")) +
               labs(title = 'B: Constrained (0-10 range)',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               scale_y_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               scale_x_continuous(limits = c(-5, 15),
                                  breaks = c(0, 5, 10)) +
               theme(plot.caption = element_text(size = 14)))
```



Population parameters: Mean = 6.2, SD = 1.7, r = 0.51

5.2 Effect of having a threshold on mean pain intensity scores

Constrained data only

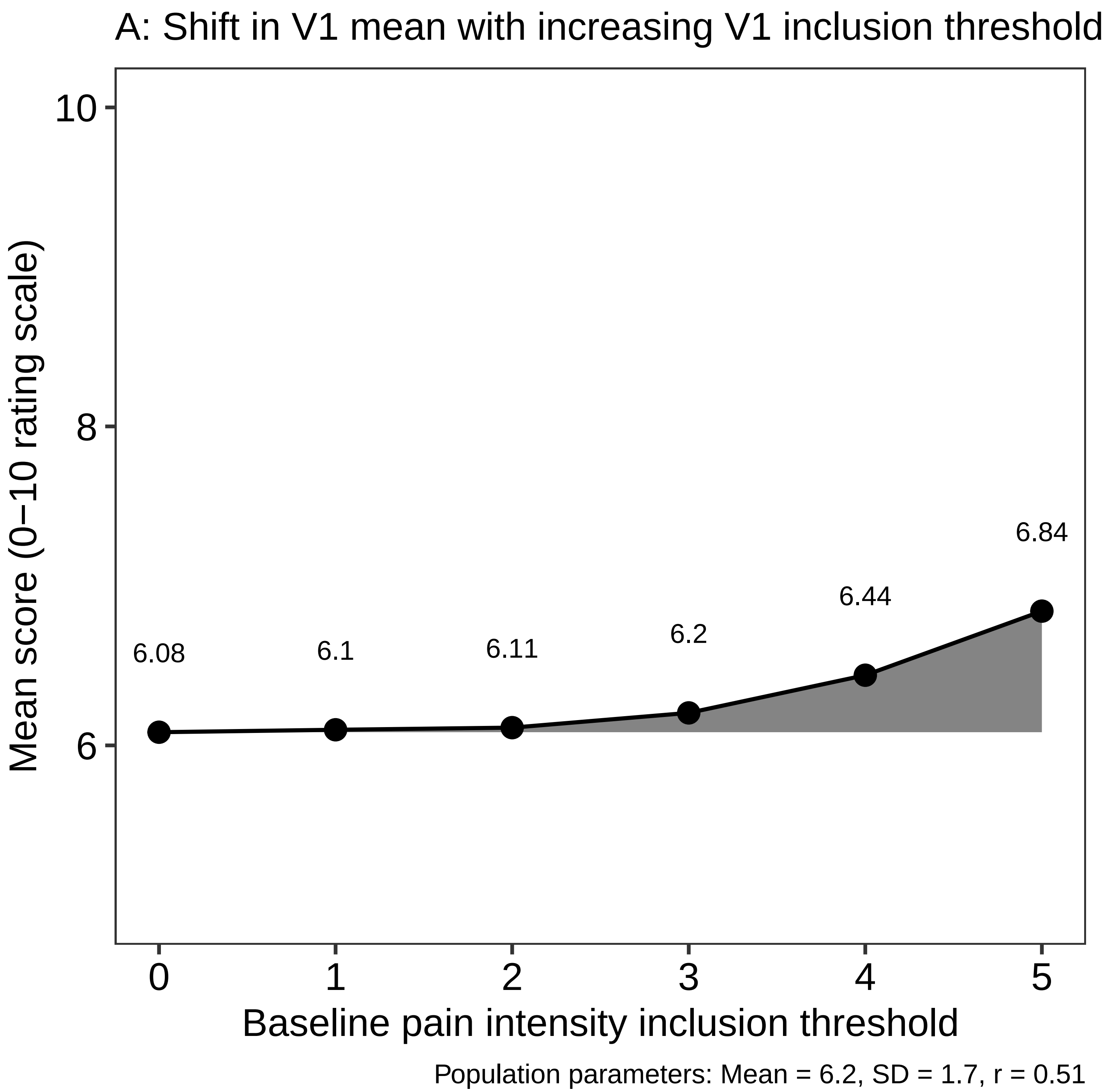
5.2.1 Model mean of V1 with increasing pain inclusion thresholds from 0 to 5

```
# Extract visit 1 data
cor_051V1 <- cor_051$V1

# Generate a vector of threshold values to iterate over
cutoff <- 0:5
```

```r
# Generate a vector of V1 means at each V1 threshold
cor_051V1.shift <- sapply(cutoff, function(x){mean(cor_051V1[cor_051V1 > x])})

# Calculate deviation
(cor_051V1.df <- data.frame(time = 'V1',
                            cutoff = cutoff,
                            cutoff2 = cutoff - 0.15, # Offset for plotting purposes
                            mean = cor_051V1.shift) %>%
        mutate(deviation = mean - mean(cor_051V1),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean  deviation
## 1   V1      0   -0.15 6.082645 0.00000000
## 2   V1      1    0.85 6.097939 0.01529382
## 3   V1      2    1.85 6.111204 0.02855958
## 4   V1      3    2.85 6.203815 0.12117013
## 5   V1      4    3.85 6.440019 0.35737385
## 6   V1      5    4.85 6.841812 0.75916762
```

```r
# Plot data
ggplot(data = cor_051V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_051V1), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'A: Shift in V1 mean with increasing V1 inclusion threshold',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



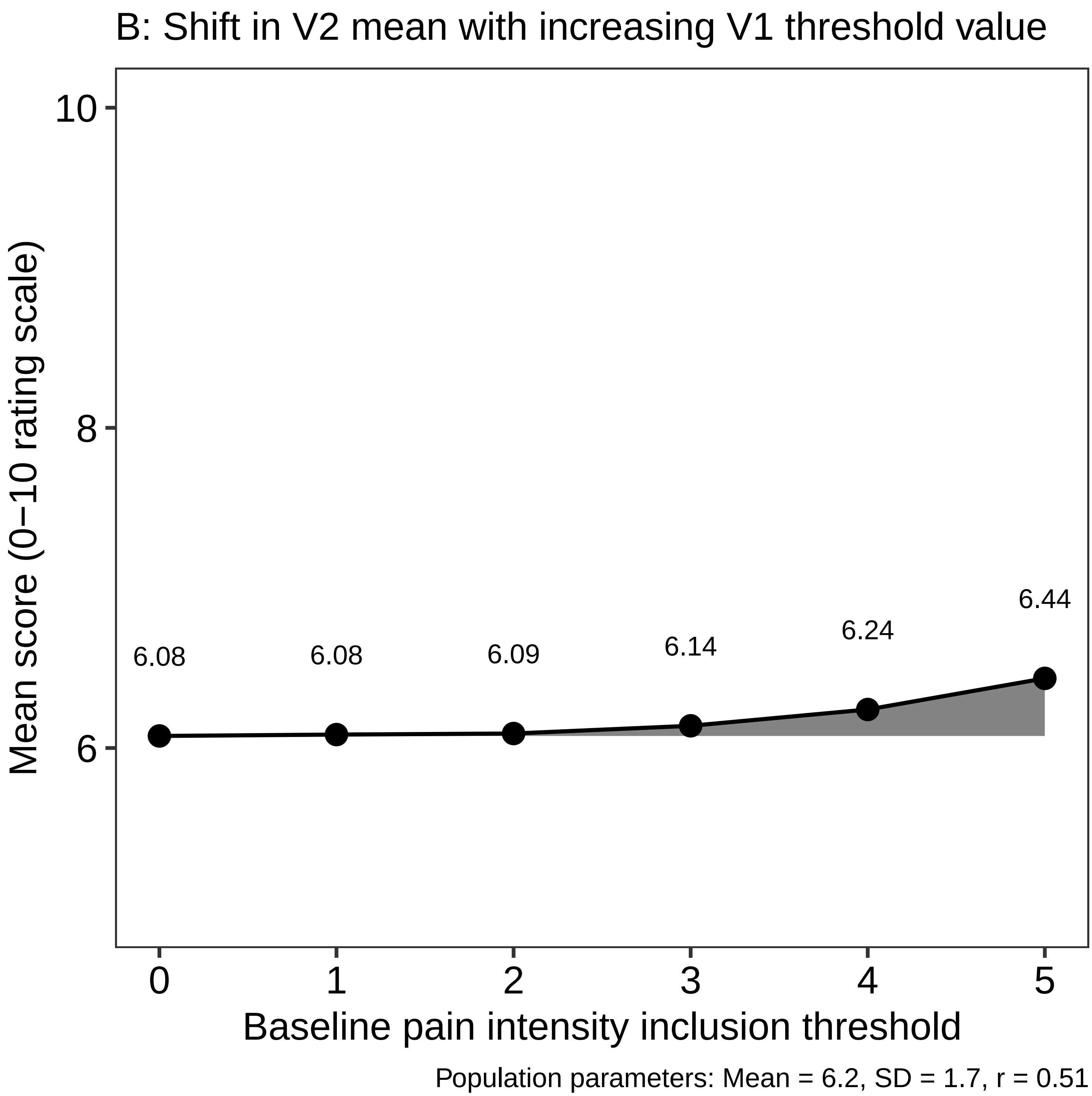
5.2.2 Model mean of V2 with increasing V1 thresholds from 0 to 5

```
# Extract visit 2 data
cor_051V2 <- cor_051$V2

# Generate a vector of threshold values to iterate over
cutoff <- 0:5

# Generate a vector of V2 means at each V1 threshold
cor_051V2.shift <- map_dbl(.x = cutoff,
                           ~ cor_051 %>%
                             filter(V1 > .x) %>%
                             .$V2 %>%
```

```r
                              mean(.))

# Calculate deviation
(cor_051V2.df <- data.frame(time = 'V2',
                            cutoff = cutoff,
                            cutoff2 = cutoff + 0.15, # Offset for plotting purposes
                            mean = cor_051V2.shift) %>%
        mutate(deviation = mean - mean(cor_051V2),
               time = as.character(time)))
```

```
##   time cutoff cutoff2     mean  deviation
## 1   V2      0    0.15 6.075349 0.00000000
## 2   V2      1    1.15 6.083941 0.00859256
## 3   V2      2    2.15 6.090498 0.01514949
## 4   V2      3    3.15 6.138678 0.06332909
## 5   V2      4    4.15 6.240444 0.16509523
## 6   V2      5    5.15 6.435127 0.35977857
```

```r
# Plot data
ggplot(data = cor_051V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(cor_051V2), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 5) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)),
              nudge_y = 0.5, size = 5) +
    scale_y_continuous(limits = c(5, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(title = 'B: Shift in V2 mean with increasing V1 threshold value',
         caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51',
         x = 'Baseline pain intensity inclusion threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



5.3 Distributional shifts caused by having a threshold

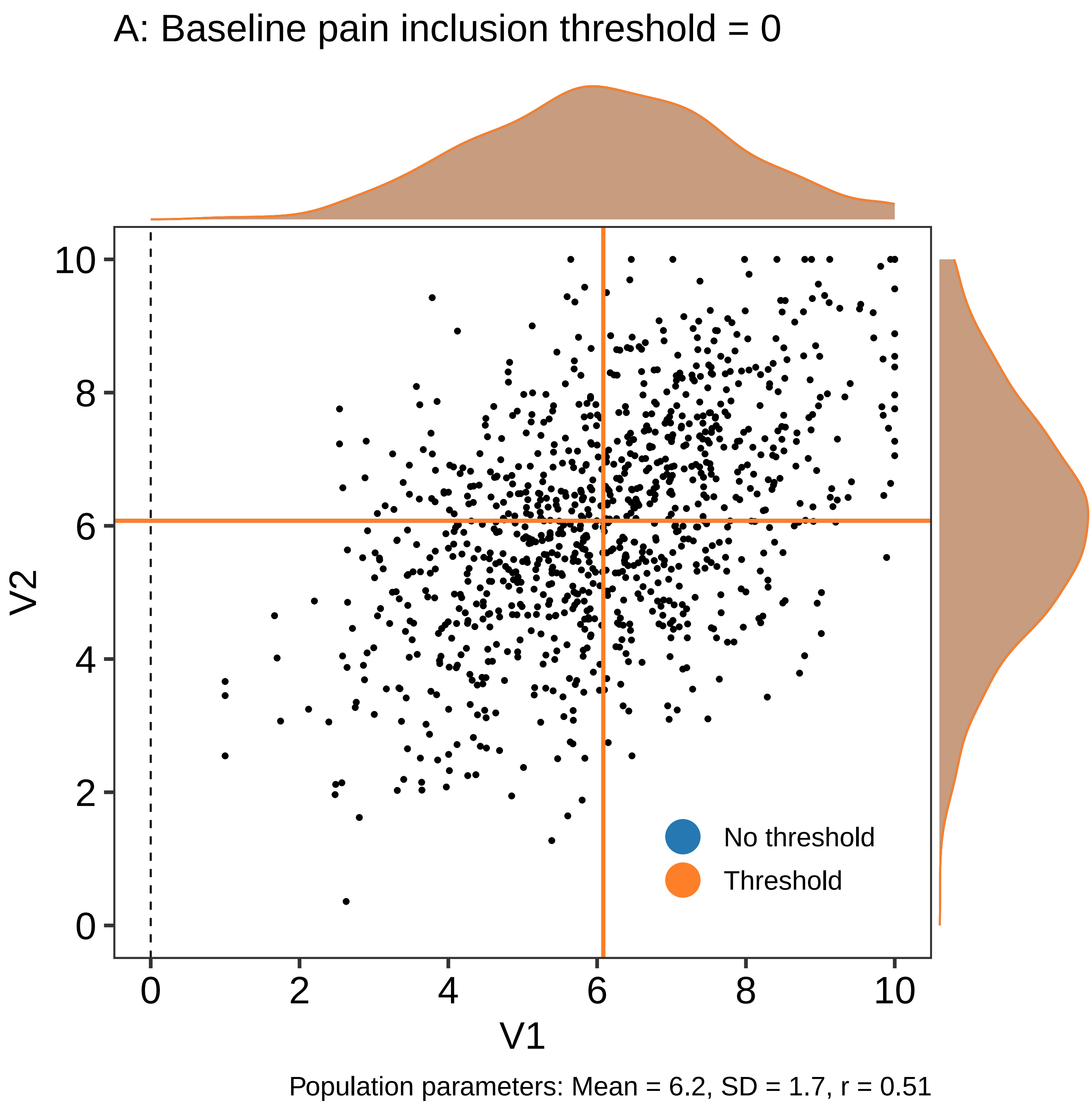
5.3.1 Threshold: 0

```
# Process data
placebo_1.0 <- cor_051 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.0 <- groupwiseMean(difference ~ 1,
```

```
                          data = placebo_1.0,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.0$.id <- 0

kable(diff_1.0)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.0073	0.95	-0.102	0.11

```
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(cor_051) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_051,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_051$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(cor_051$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.0$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'A: Baseline pain inclusion threshold = 0',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



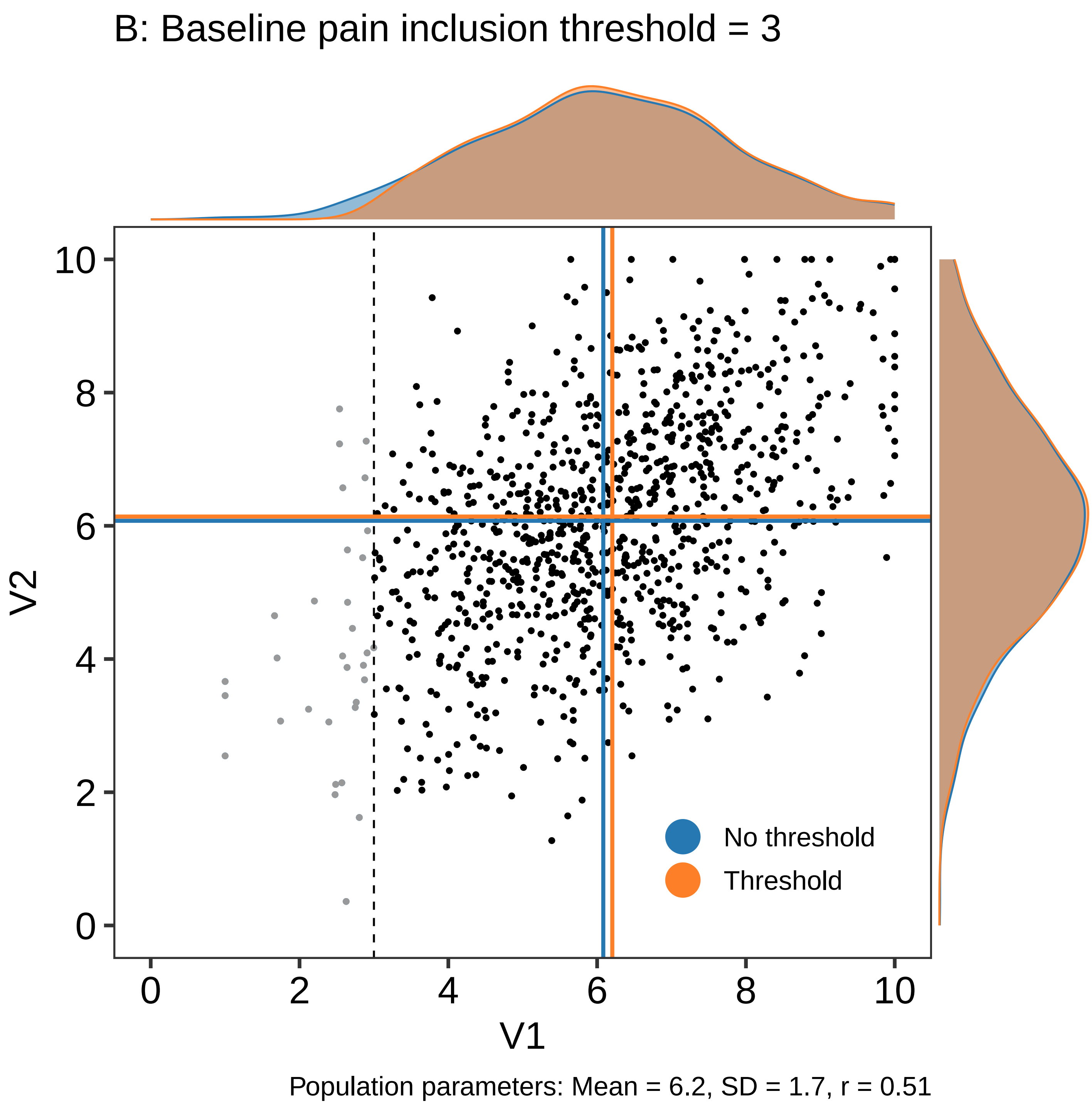
5.3.2 Threshold: 3

```
# Process data
placebo_1.3 <- cor_051 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_1.3 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.3,
                          R = 2000,
```

```r
                              traditional = FALSE,
                              bca = TRUE)

diff_1.3$.id <- 3

kable(diff_1.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	968	0.0651	0.95	-0.0355	0.173

```r
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(cor_051) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_051,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_051$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(cor_051$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.3$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'B: Baseline pain inclusion threshold = 3',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



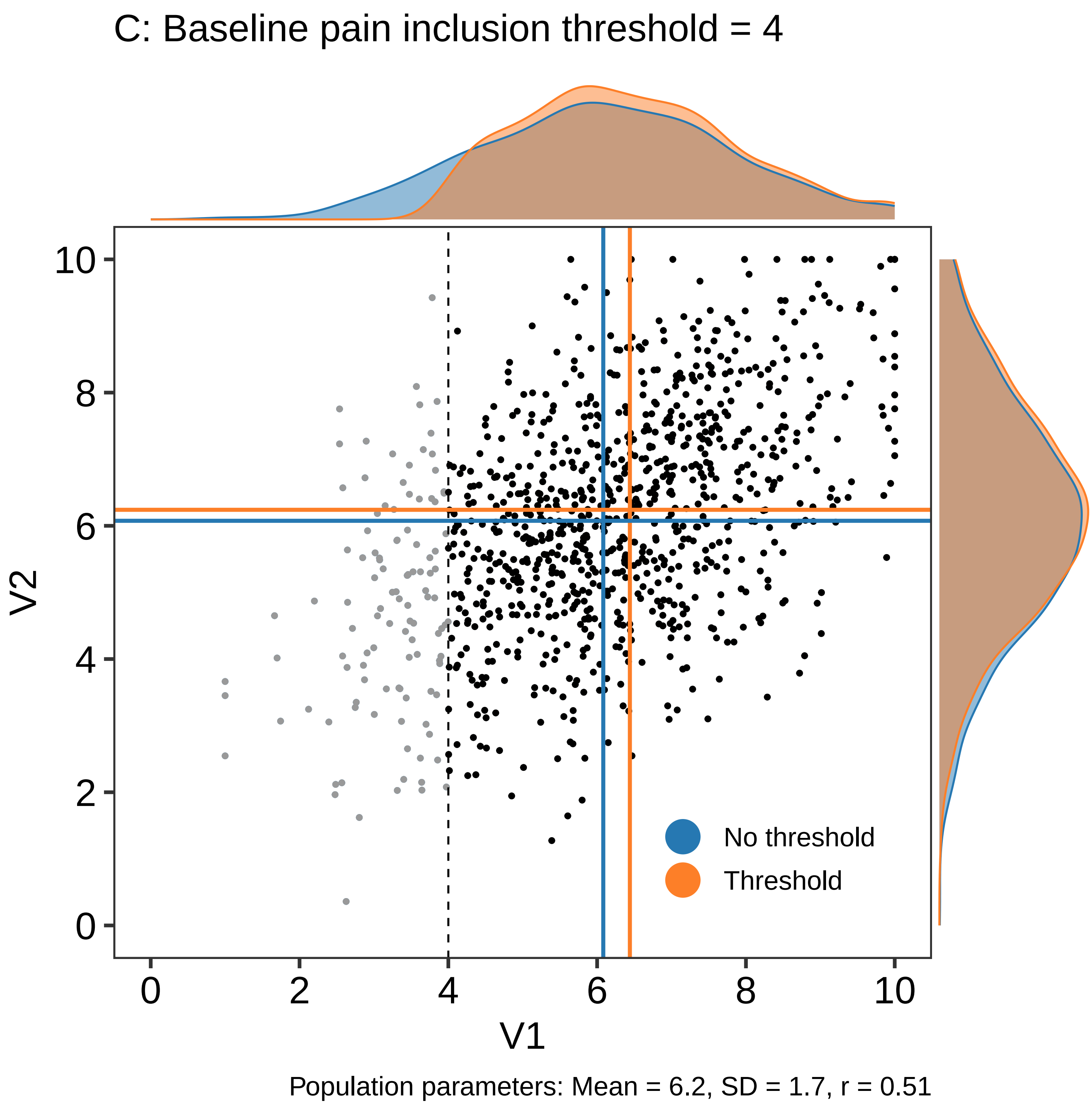
5.3.3 Threshold: 4

```
# Process that data
placebo_1.4 <- cor_051 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```r
diff_1.4 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.4,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.4$.id <- 4

kable(diff_1.4)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
4	889	0.2	0.95	0.0975	0.306

```r
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
               bind_rows(cor_051) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_051,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_051$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(cor_051$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.4$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'C: Baseline pain inclusion threshold = 4',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



5.3.4 Threshold: 5

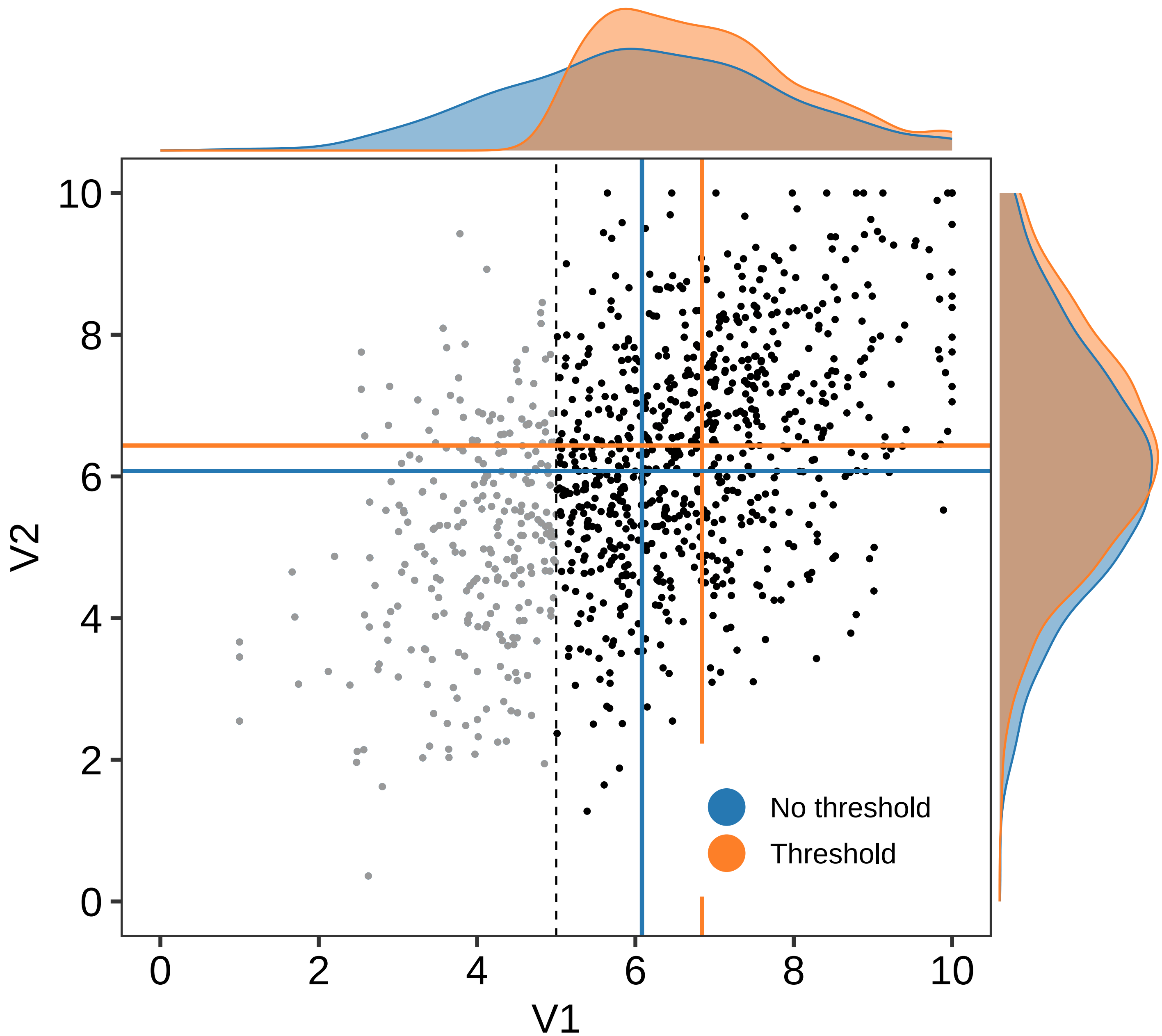
```
# Process that data
placebo_1.5 <- cor_051 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
```

```r
diff_1.5 <- groupwiseMean(difference ~ 1,
                          data = placebo_1.5,
                          R = 2000,
                          traditional = FALSE,
                          bca = TRUE)

diff_1.5$.id <- 5

kable(diff_1.5)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	736	0.407	0.95	0.299	0.517

```r
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(cor_051) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1) +
               guides(colour = guide_legend(override.aes = list(size = 8))) +
               geom_point(data = cor_051,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(cor_051$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(cor_051$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.5$V2),
                          colour = pal[2], size = 1) +
               scale_y_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_x_continuous(limits = c(0, 10),
                                  breaks = c(0, 2, 4, 6, 8, 10)) +
               scale_fill_manual(values = pal) +
               scale_colour_manual(values = pal) +
               labs(title = 'D: Baseline pain inclusion threshold = 5',
                    caption = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.8, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

D: Baseline pain inclusion threshold = 5

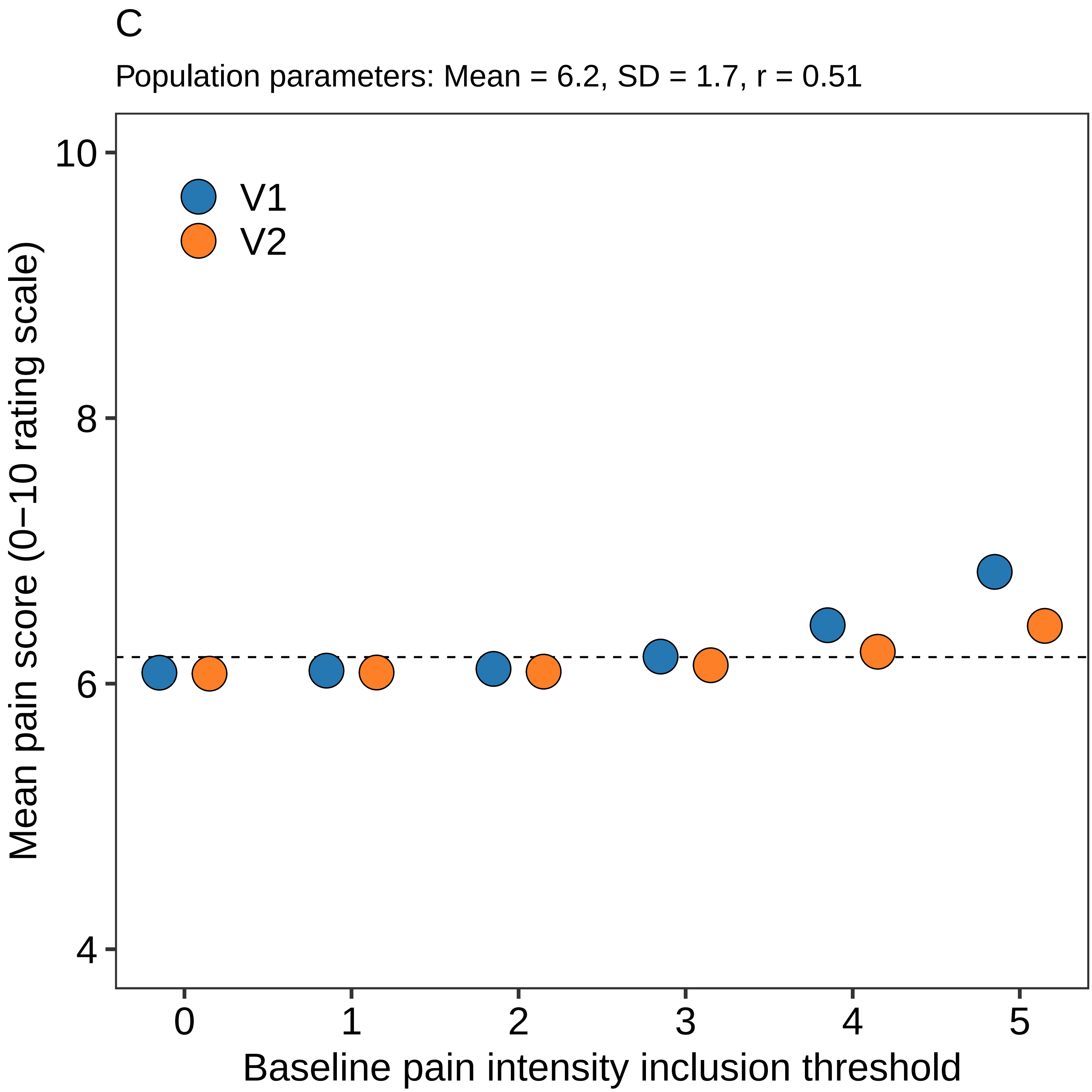


Population parameters: Mean = 6.2, SD = 1.7, r = 0.51

5.4 Summary plots

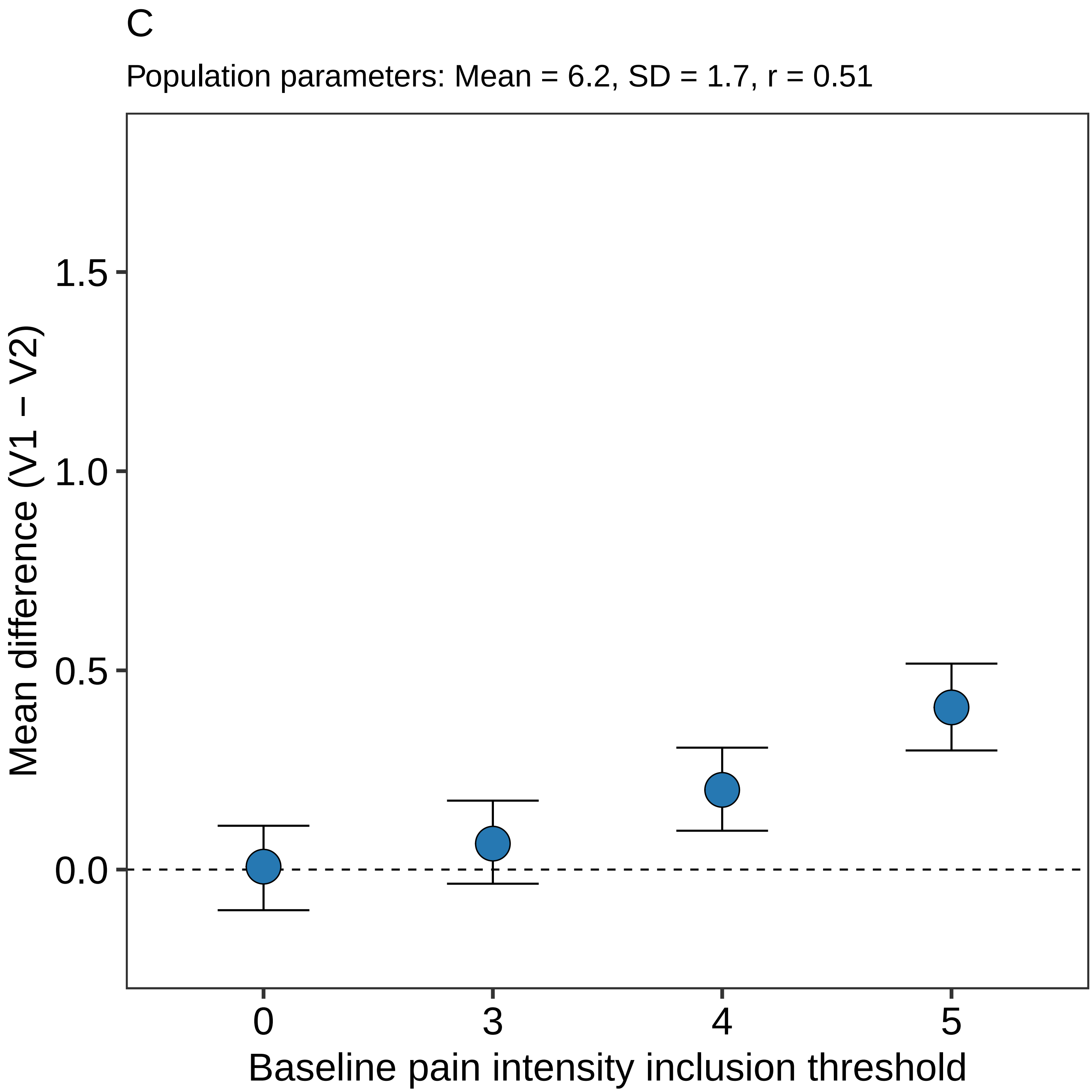
```
# Plot 1
shift_1 <- cor_051V1.df %>%
  bind_rows(cor_051V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
      x = cutoff2,
      fill= time) +
  geom_hline(yintercept = 6.2,
             linetype = 2) +
  geom_point(shape = 21,
```

```
             size = 8) +
  labs(title = 'C',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean pain score (0-10 rating scale)') +
  scale_x_continuous(breaks = 0:5) +
  scale_y_continuous(limits = c(4, 10)) +
  scale_fill_manual(values = pal) +
  theme(legend.title = element_blank(),
        legend.position = c(0.12, 0.89),
        legend.text = element_text(size = 20)); shift_1
```




```
# Plot 2
# Bind diff_*.* dataframes
```

```
diff_all_1 <- diff_1.0 %>%
  bind_rows(diff_1.3, diff_1.4, diff_1.5)

pp_1 <- diff_all_1 %>%
  mutate(Threshold = factor(.id)) %>%
  ggplot(data = .) +
  aes(x = Threshold,
      y = Mean,
      ymin = Bca.lower,
      ymax = Bca.upper) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_errorbar(width = 0.4) +
  geom_point(shape = 21,
             fill = pal[[1]],
             size = 8) +
  labs(title = 'C',
       subtitle = 'Population parameters: Mean = 6.2, SD = 1.7, r = 0.51',
       x = 'Baseline pain intensity inclusion threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_1
```



6 Session information

```r
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.5
##
## 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_1.0.1   knitr_1.29        MBESS_4.7.0       ggExtra_0.9
##  [5] rcompanion_2.3.25 MASS_7.3-51.6     magrittr_1.5      forcats_0.5.0
##  [9] stringr_1.4.0     dplyr_1.0.0       purrr_0.3.4       readr_1.3.1
## [13] tidyr_1.1.0       tibble_3.0.1      ggplot2_3.3.2     tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-148       matrixStats_0.56.0 fs_1.4.1           lubridate_1.7.9
##  [5] httr_1.4.1         tools_4.0.2        backports_1.1.8    R6_2.4.1
##  [9] nortest_1.0-4      DBI_1.1.0          colorspace_1.4-1   withr_2.2.0
## [13] tidyselect_1.1.0   compiler_4.0.2     cli_2.0.2          rvest_0.3.5
## [17] expm_0.999-4       xml2_1.3.2         sandwich_2.5-1     labeling_0.3
## [21] scales_1.1.1       lmtest_0.9-37      mvtnorm_1.1-1      multcompView_0.1-8
## [25] digest_0.6.25      rmarkdown_2.3      pkgconfig_2.0.3    htmltools_0.5.0
## [29] highr_0.8          fastmap_1.0.1      dbplyr_1.4.4       rlang_0.4.6
## [33] readxl_1.3.1       rstudioapi_0.11    shiny_1.5.0        farver_2.0.3
## [37] generics_0.0.2     zoo_1.8-8          jsonlite_1.6.1     modeltools_0.2-23
## [41] Matrix_1.2-18      Rcpp_1.0.4.6       DescTools_0.99.36  munsell_0.5.0
## [45] fansi_0.4.1        lifecycle_0.2.0    stringi_1.4.6      multcomp_1.4-13
## [49] yaml_2.2.1         plyr_1.8.6         grid_4.0.2         blob_1.2.1
## [53] promises_1.1.1     parallel_4.0.2     crayon_1.3.4       miniUI_0.1.1.1
## [57] lattice_0.20-41    haven_2.3.1        splines_4.0.2      hms_0.5.3
## [61] pillar_1.4.4       EMT_1.1            boot_1.3-25        codetools_0.2-16
## [65] stats4_4.0.2       reprex_0.3.0       glue_1.4.1         evaluate_0.14
## [69] modelr_0.1.8       vctrs_0.3.1        httpuv_1.5.4       cellranger_1.1.0
## [73] gtable_0.3.0       assertthat_0.2.1   xfun_0.15          mime_0.9
## [77] coin_1.3-1         xtable_1.8-4       libcoin_1.0-5      broom_0.5.6
## [81] later_1.1.0.1      survival_3.1-12    TH.data_1.0-10     ellipsis_0.3.1
```