Regression to the mean modeling

Mean pain rating of 5.2 at 0.2 correlation

Peter Kamerman

Last knitted: 15 August 2019

Contents

Generate 2x2 covariance matrix

Generate a covariance matrix using an SD of 1.2, 1.7, and 2.2, and correlation of 0.2.

```
cor <- matrix(c(1, 0.2, 0.2, 1), ncol = 2)
std_1 <- c(1.2, 1.2)
std_2 <- c(1.7, 1.7)
std_3 <- c(2.2, 2.2)
cov_1 <- cor2cov(cor.mat = cor,
                 sd = std_1)
cov_1
```

```
##       [,1]  [,2]
## [1,] 1.440 0.288
## [2,] 0.288 1.440
```

```
cov_2 <- cor2cov(cor.mat = cor,
                 sd = std_2)
cov_2
```

```
##       [,1]  [,2]
## [1,] 2.890 0.578
## [2,] 0.578 2.890
```

```
cov_3 <- cor2cov(cor.mat = cor,
                 sd = std_3)
cov_3
```

```
##       [,1]  [,2]
## [1,] 4.840 0.968
## [2,] 0.968 4.840
```

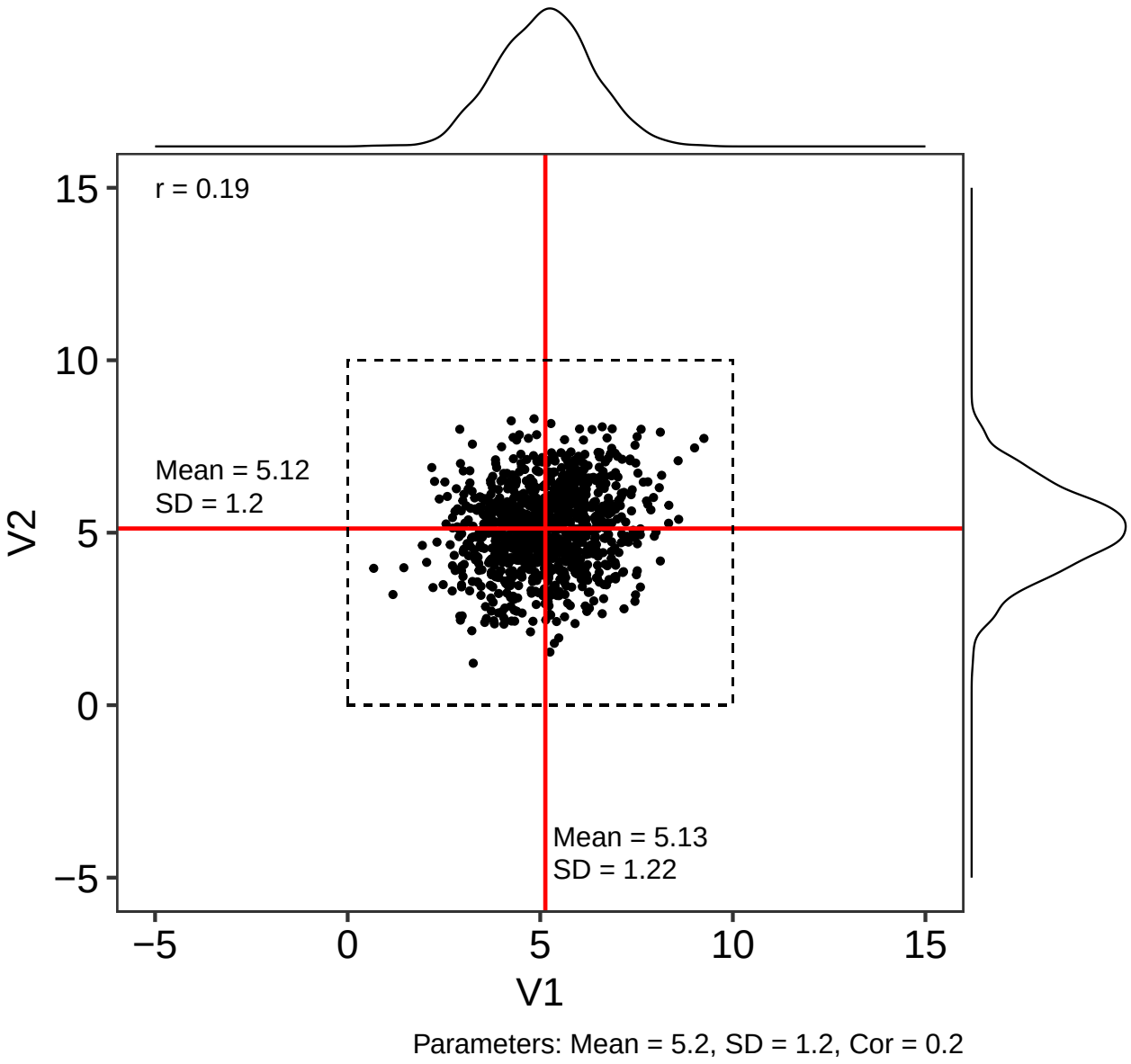
Mean = 5.2, SD = 1.2, Cor = 0.2

Generate and summarise data

Base data

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

# Generate the data
five_1.base <- as.data.frame(mvrnorm(n = 1000, mu = c(5.2, 5.2), Sigma = cov_1))

# Plot base data
ggMarginal(ggplot(data = five_1.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(five_1.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(five_1.base$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(five_1.base$V1, five_1.base$V2), 2)}")) +
annotate(geom = 'text', x = -5, y = mean(five_1.base$V2) + 1.7,
         hjust = 0, size = 5,
         label = str_glue("Mean = {round(mean(five_1.base$V2), 2)}")) +
annotate(geom = 'text', x = -5, y = mean(five_1.base$V2) + 0.75,
         hjust = 0, size = 5,
         label = str_glue("SD = {round(sd(five_1.base$V2),2)}")) +
annotate(geom = 'text', x = mean(five_1.base$V1) + 0.2, y = -3.8,
         hjust = 0, size = 5,
         label = str_glue("Mean = {round(mean(five_1.base$V1), 2)}")) +
annotate(geom = 'text', x = mean(five_1.base$V1) + 0.2, y = -4.75,
         hjust = 0, size = 5,
         label = str_glue("SD = {round(sd(five_1.base$V1), 2)}")) +
labs(title = 'A: Unconstained',
     caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
scale_y_continuous(limits = c(-5, 15)) +
scale_x_continuous(limits = c(-5, 15)) +
theme(plot.caption = element_text(size = 14)))
```




```
# Linear regression
summary(lm(V2 ~ V1, data = five_1.base))
```

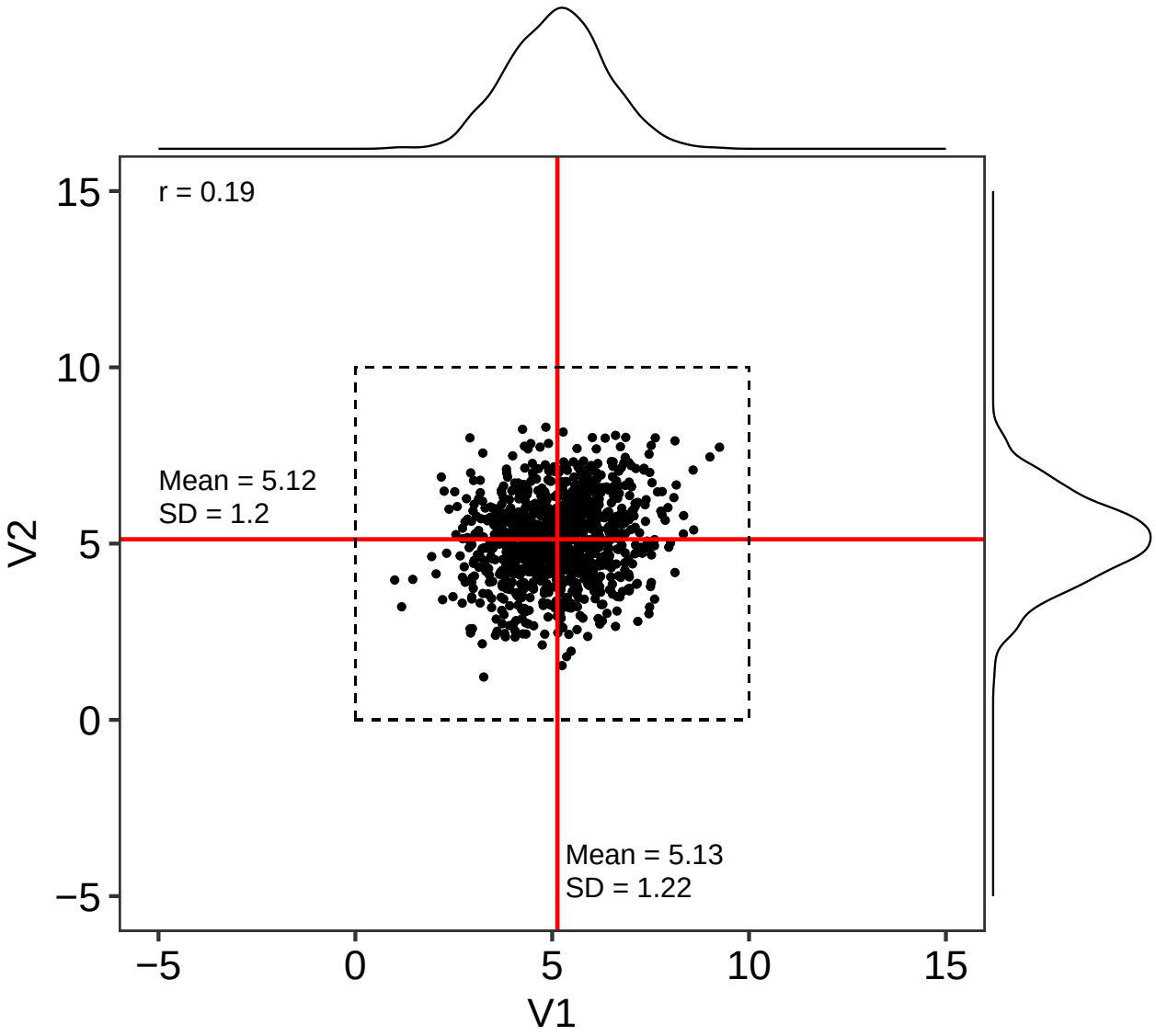
```
##
## Call:
## lm(formula = V2 ~ V1, data = five_1.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6055 -0.7584  0.0394  0.8105  3.2859
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.17617    0.16154  25.852  < 2e-16 ***
```

```
## V1           0.18442    0.03063   6.021 2.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.179 on 998 degrees of freedom
## Multiple R-squared:  0.03506,    Adjusted R-squared:  0.03409
## F-statistic: 36.26 on 1 and 998 DF,  p-value: 2.427e-09
```

Constrain values to 0-10 range

```
# Process data
five_1 <- five_1.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)) %>%
    mutate(group = 'No threshold')

# Plot processed data
ggMarginal(ggplot(data = five_1) +
                aes(x = V1, y = V2) +
                geom_point() +
                geom_hline(yintercept = mean(five_1$V2),
                           colour = 'red', size = 1) +
                geom_vline(xintercept = mean(five_1$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(five_1$V1, five_1$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_1$V2) + 1.7,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_1$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_1$V2) + 0.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_1$V2),2)}")) +
                annotate(geom = 'text', x = mean(five_1$V1) + 0.2, y = -3.8,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_1$V1), 2)}")) +
                annotate(geom = 'text', x = mean(five_1$V1) + 0.2, y = -4.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_1$V1), 2)}")) +
                labs(title = 'B: Constrained (0-10 range)',
                     caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
                scale_y_continuous(limits = c(-5, 15)) +
                scale_x_continuous(limits = c(-5, 15)) +
```

```
  theme(plot.caption = element_text(size = 14)))
```




```
# Linear regression
summary(lm(V2 ~ V1, data = five_1))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = five_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6055 -0.7584  0.0395  0.8105  3.2861
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

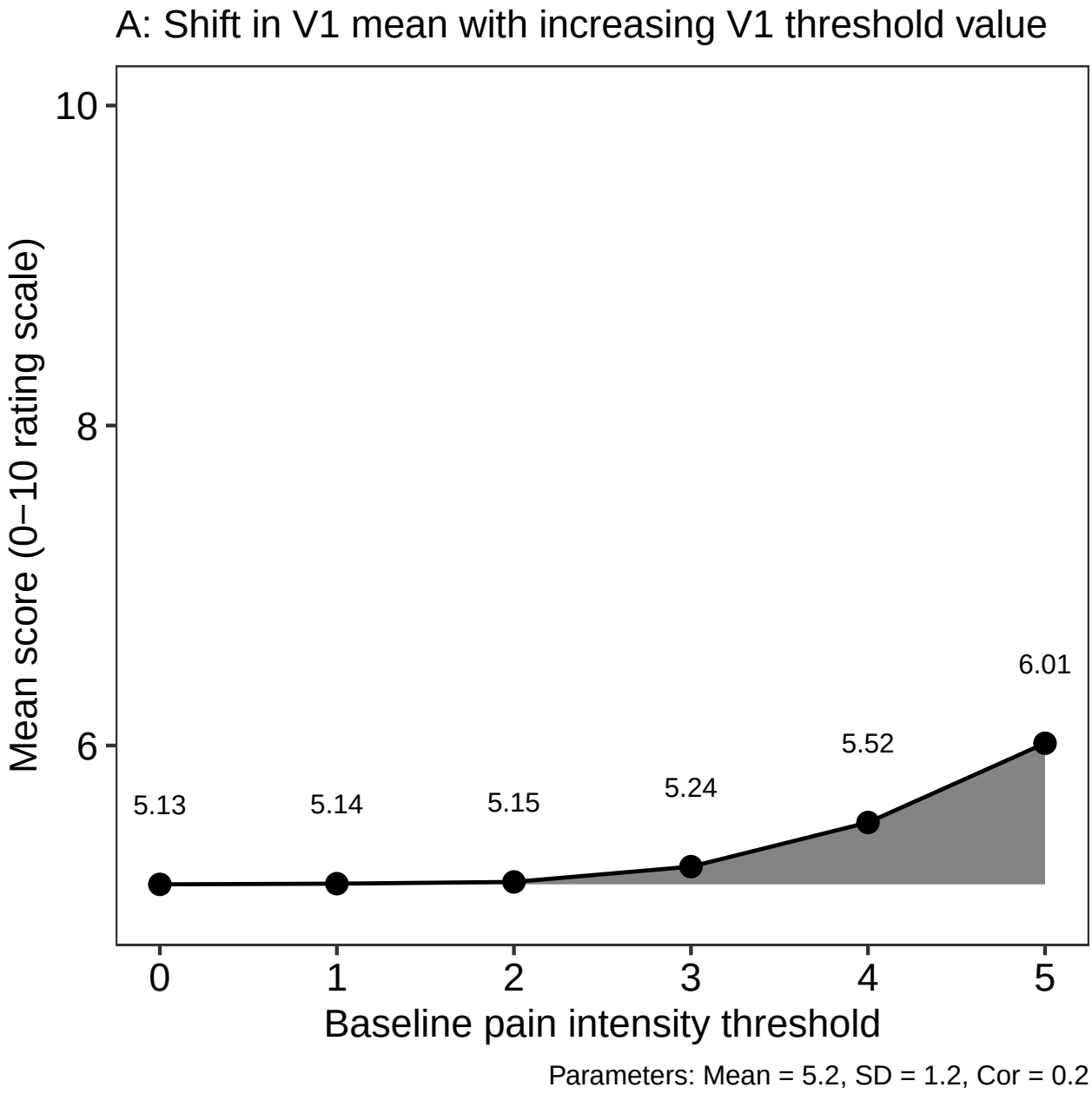
```
## (Intercept)  4.17563    0.16170  25.824  < 2e-16 ***
## V1           0.18451    0.03066   6.019 2.47e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.179 on 998 degrees of freedom
## Multiple R-squared:  0.03503,    Adjusted R-squared:  0.03406
## F-statistic: 36.23 on 1 and 998 DF,  p-value: 2.466e-09
```

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

```
# Extract visit 1 data
five_1V1 <- five_1$V1

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

# Generate a vector of V1 means at each V1 threshold
five_1V1.shift <- sapply(cutoff, function(x){mean(five_1V1[five_1V1 > x])})

# Calculate deviation
(five_1V1.df <- data.frame(cutoff = cutoff,
                           mean = five_1V1.shift) %>%
        mutate(deviation = mean - mean(five_1V1)))
```

```
##   cutoff     mean   deviation
## 1      0 5.132359 0.000000000
## 2      1 5.136496 0.004136496
## 3      2 5.147375 0.015015587
## 4      3 5.242943 0.110584082
## 5      4 5.518841 0.386482235
## 6      5 6.014179 0.881819446
```

```
# Plot data
ggplot(data = five_1V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_1V1), 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 threshold value',
         caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



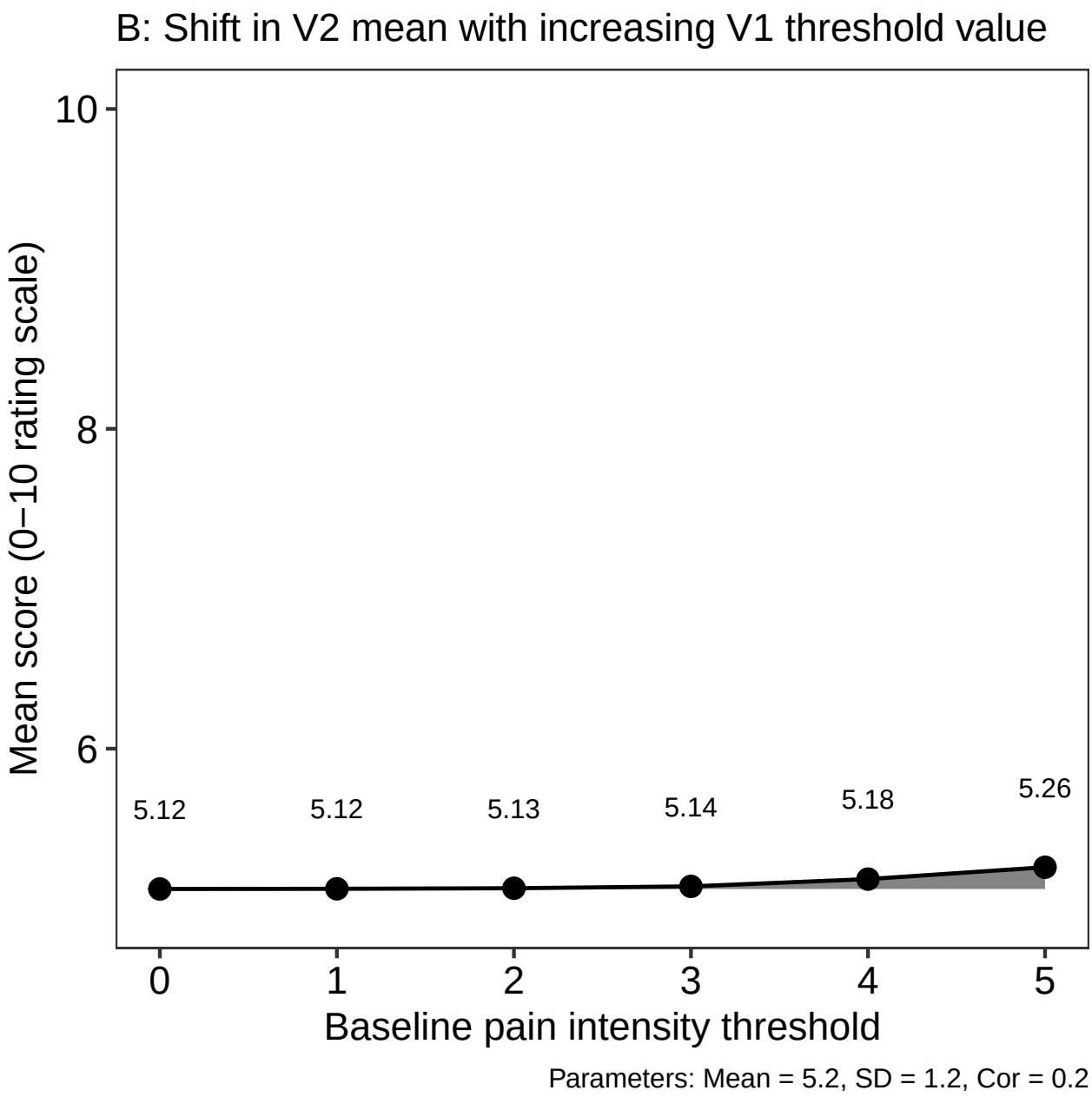
Model mean of V2 with increasing V1 thresholds from 0 to 5

```
# Extract visit 2 data
five_1V2 <- five_1$V2

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

# Generate a vector of V2 means at each V1 threshold
five_1V2.shift <- map_dbl(.x = cutoff,
                          ~ five_1 %>%
                              filter(V1 > .x) %>%
                              .$V2 %>%
```

```
                              mean(.))

# Calculate deviation
(five_1V2.df <- data.frame(cutoff = cutoff,
                           mean = five_1V2.shift) %>%
        mutate(deviation = mean - mean(five_1V2)))
```

```
##   cutoff     mean   deviation
## 1      0 5.122621 0.000000000
## 2      1 5.123778 0.001156347
## 3      2 5.127338 0.004716806
## 4      3 5.138913 0.016291210
## 5      4 5.184385 0.061763129
## 6      5 5.258636 0.136014079
```

```
# Plot data
ggplot(data = five_1V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_1V2), 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 = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



Placebo response

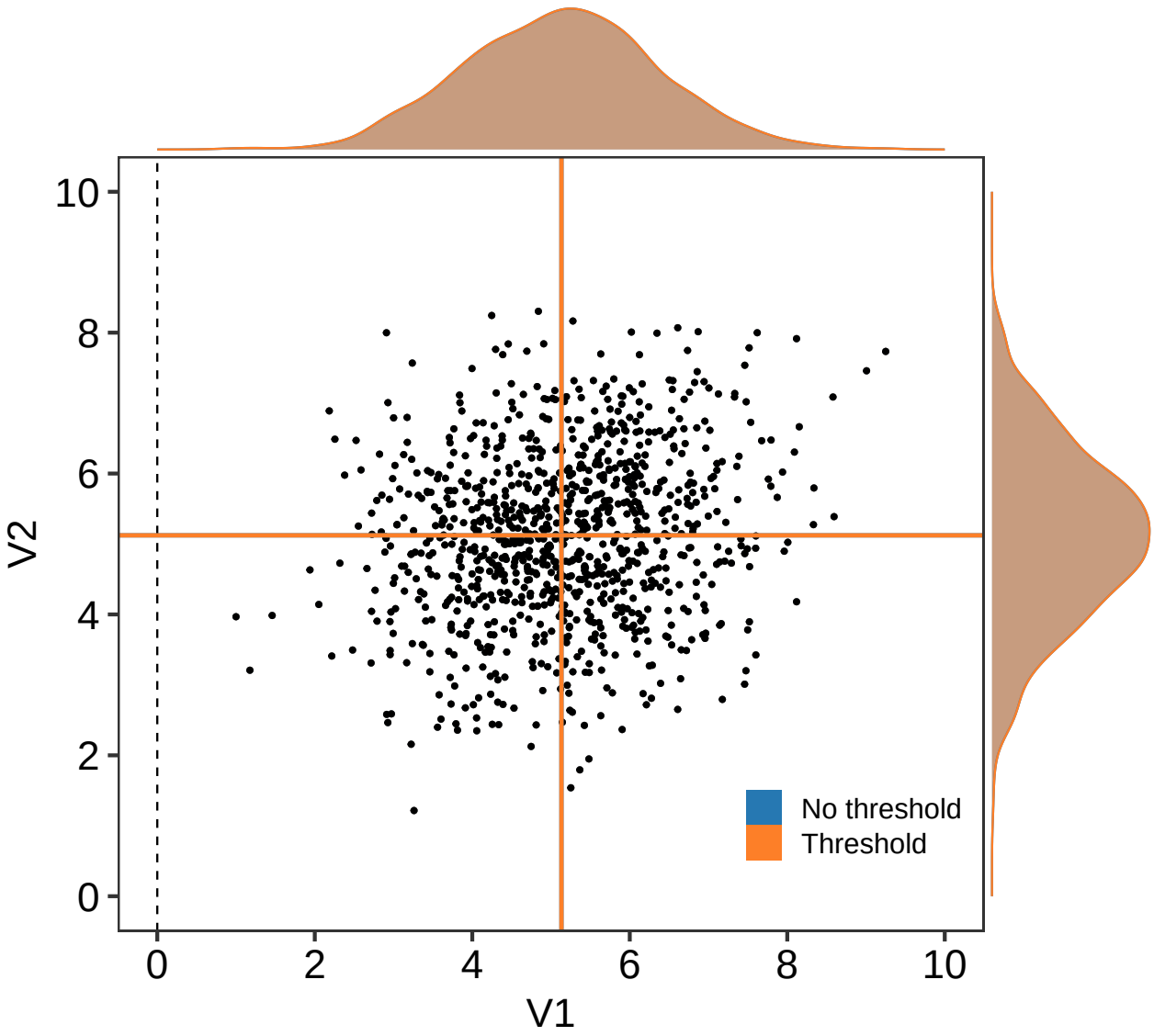
threshold: 0

```
# Process data
placebo_1.0 <- five_1 %>%
    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	0.00974	0.95	-0.0893	0.104

```
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(five_1) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_1$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(five_1$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.0$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 threshold = 0',
                    caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



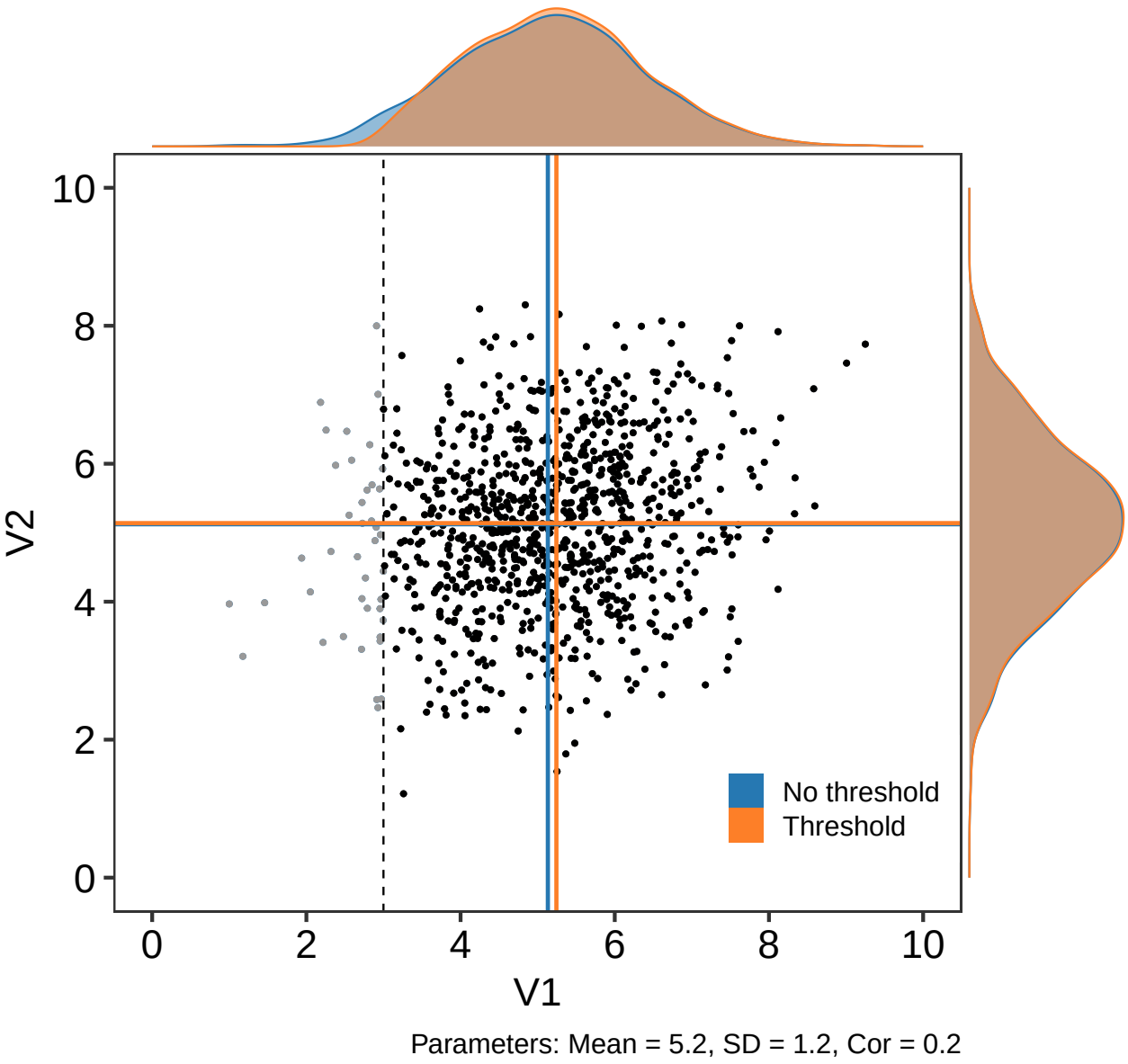
threshold: 3

```
# Process data
placebo_1.3 <- five_1 %>%
    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,
                          traditional = FALSE,
```

```
                          bca = TRUE)

diff_1.3$.id <- 3

kable(diff_1.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	958	0.104	0.95	0.00219	0.202

```
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(five_1) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_1$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(five_1$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.3$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 threshold = 3',
                    caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



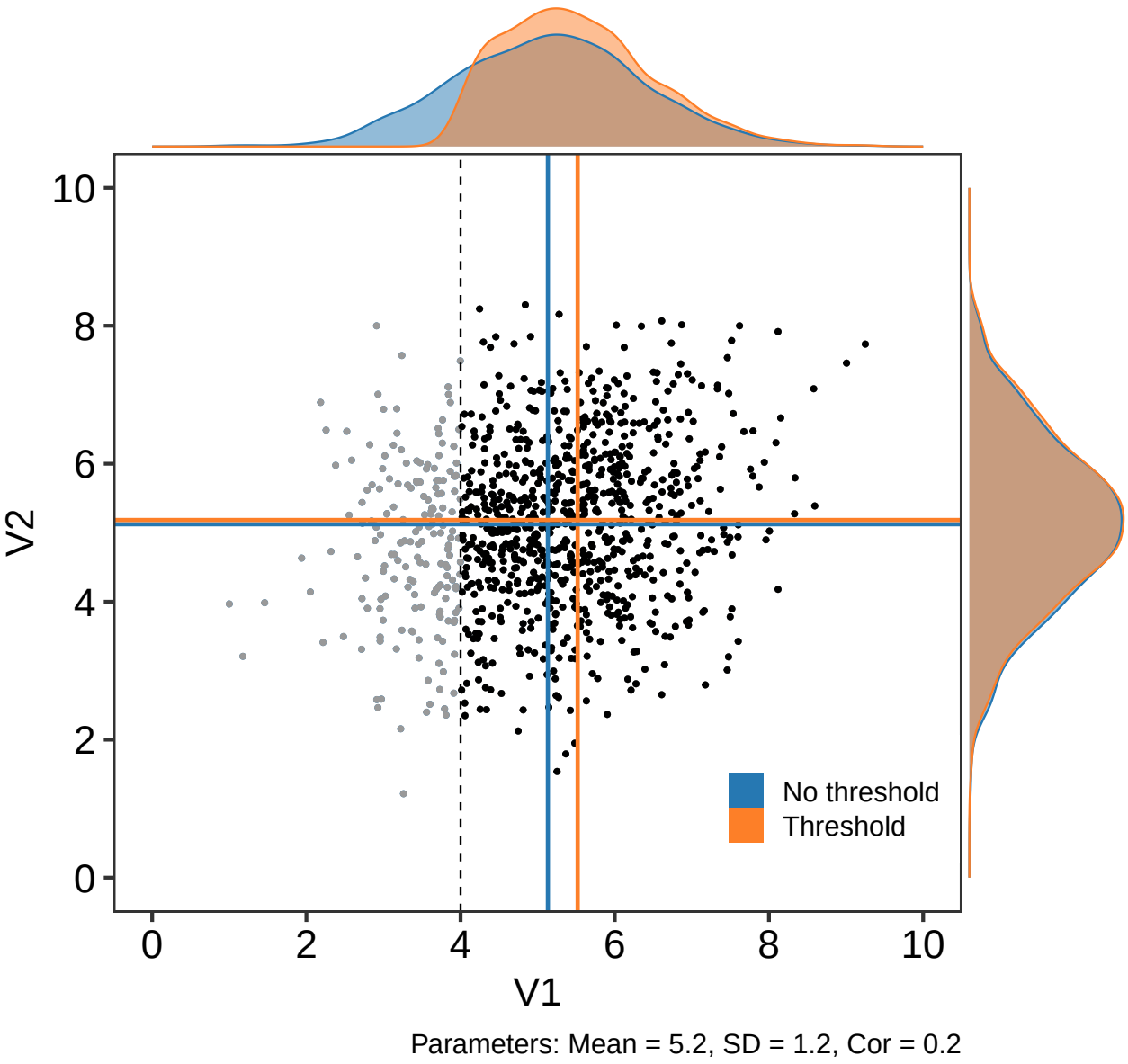
threshold: 4

```
# Process that data
placebo_1.4 <- five_1 %>%
    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	821	0.334	0.95	0.238	0.43

```
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
               bind_rows(five_1) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_1$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(five_1$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.4$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 threshold = 4',
                    caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



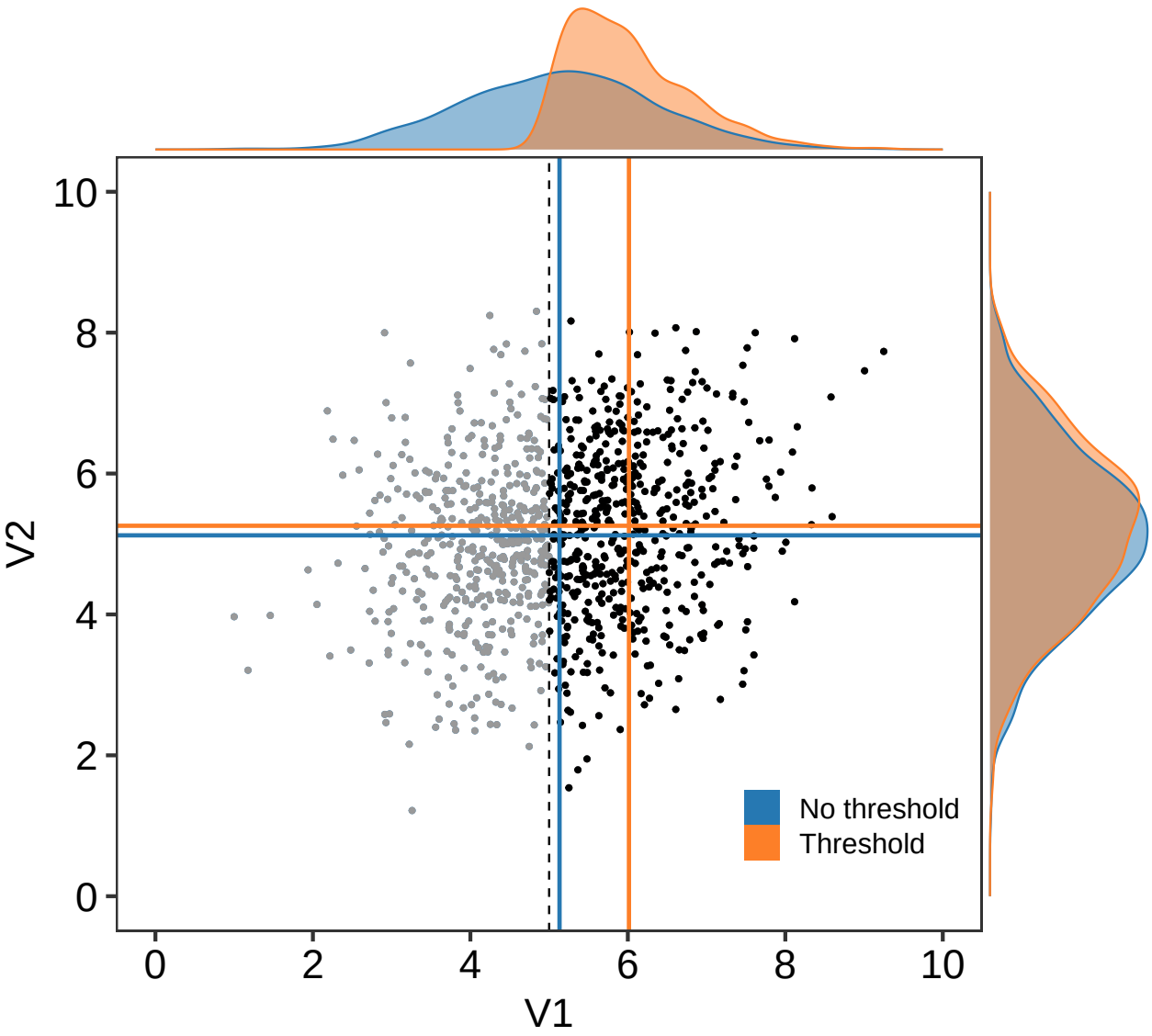
threshold: 5

```
# Process that data
placebo_1.5 <- five_1 %>%
    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
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	547	0.756	0.95	0.644	0.868

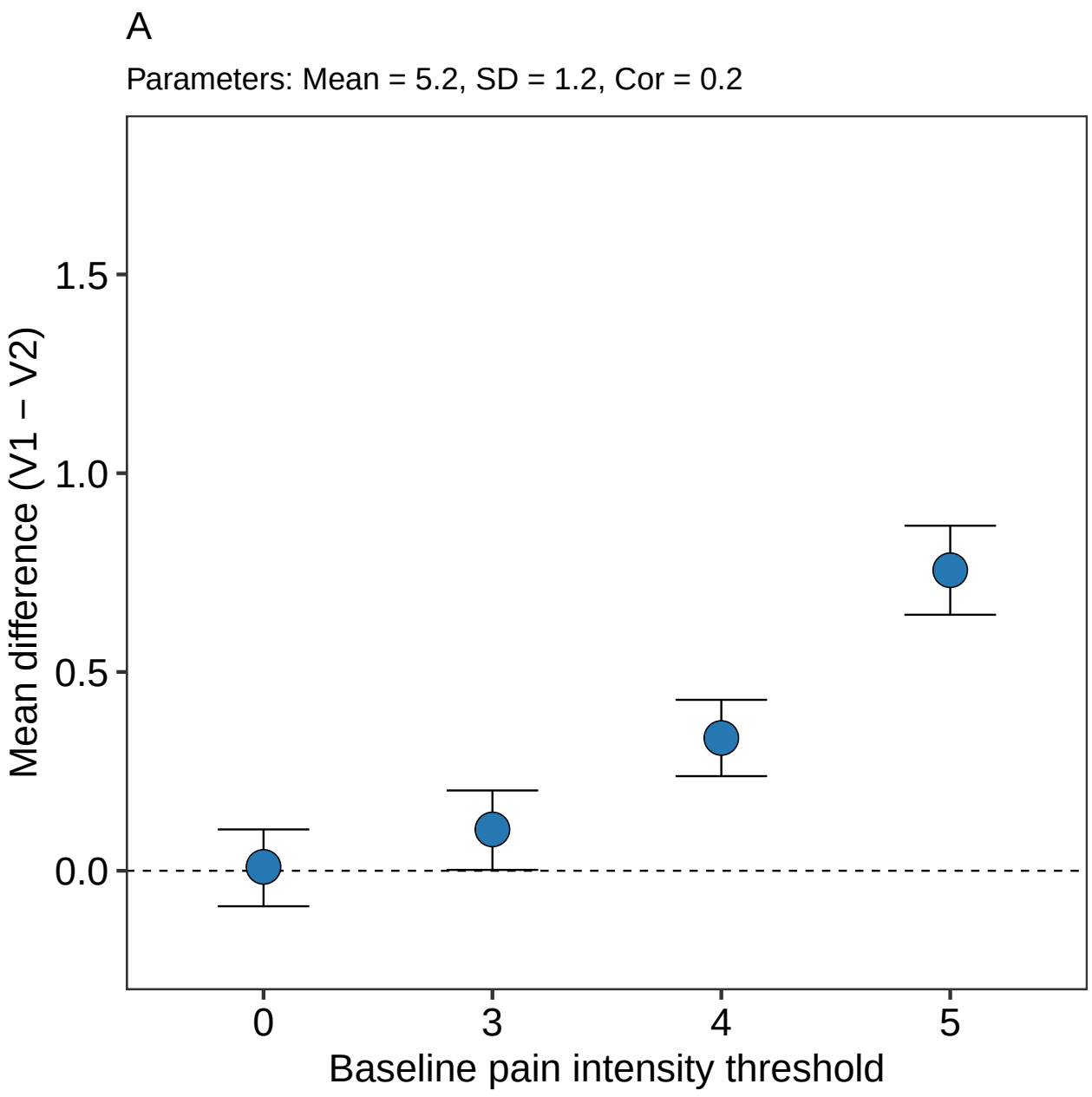
```
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(five_1) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_1$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_1.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(five_1$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_1.5$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 threshold = 5',
                    caption = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



Difference plot

```
# 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 = 'A',
     subtitle = 'Parameters: Mean = 5.2, SD = 1.2, Cor = 0.2',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)') +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_1
```

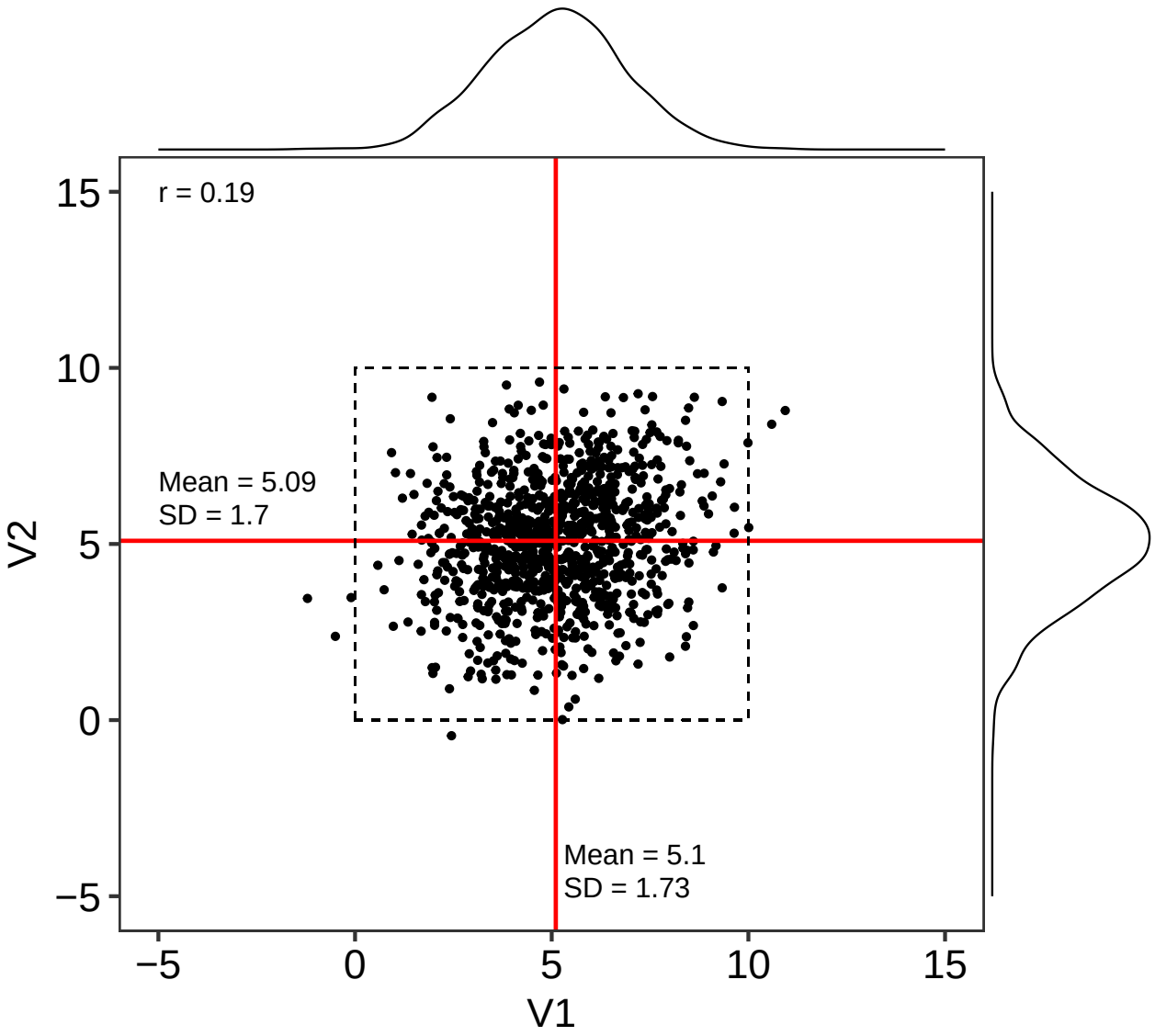


Mean = 5.2, SD = 1.7, Cor = 0.2

Generate and summarise data

Base data

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

# Generate the data
five_2.base <- as.data.frame(mvrnorm(n = 1000, mu = c(5.2, 5.2), Sigma = cov_2))

# Plot base data
ggMarginal(ggplot(data = five_2.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(five_2.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(five_2.base$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(five_2.base$V1, five_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(five_2.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(five_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(five_2.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(five_2.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(five_2.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(five_2.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(five_2.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(five_2.base$V1), 2)}")) +
               labs(title = 'A: Unconstrained',
                    caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(plot.caption = element_text(size = 14)))
```



Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2

```
# Linear regression
summary(lm(V2 ~ V1, data = five_2.base))
```

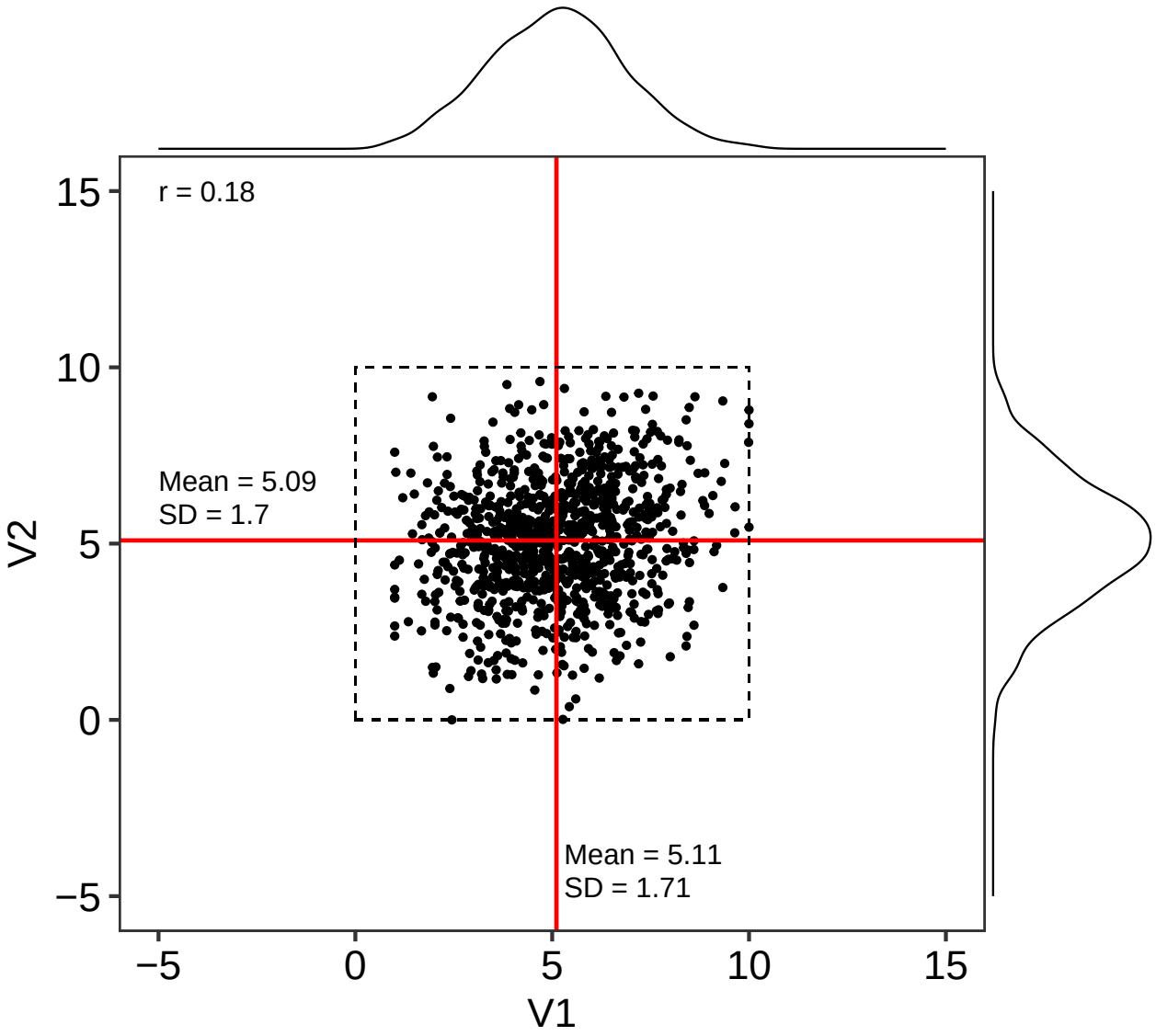
```
##
## Call:
## lm(formula = V2 ~ V1, data = five_2.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1078 -1.0744  0.0558  1.1482  4.6550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.14916    0.16500  25.147  < 2e-16 ***
```

```
## V1           0.18442    0.03063   6.021 2.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.67 on 998 degrees of freedom
## Multiple R-squared:  0.03506,    Adjusted R-squared:  0.03409
## F-statistic: 36.26 on 1 and 998 DF,  p-value: 2.427e-09
```

Constrain values to 0-10 range

```
# Process data
five_2 <- five_2.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)) %>%
    mutate(group = 'No threshold')

# Plot processed data
ggMarginal(ggplot(data = five_2) +
                aes(x = V1, y = V2) +
                geom_point() +
                geom_hline(yintercept = mean(five_2$V2),
                           colour = 'red', size = 1) +
                geom_vline(xintercept = mean(five_2$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(five_2$V1, five_2$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_2$V2) + 1.7,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_2$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_2$V2) + 0.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_2$V2),2)}")) +
                annotate(geom = 'text', x = mean(five_2$V1) + 0.2, y = -3.8,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_2$V1), 2)}")) +
                annotate(geom = 'text', x = mean(five_2$V1) + 0.2, y = -4.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_2$V1), 2)}")) +
                labs(title ='B: Constrained (0-10 range)',
                     caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
                scale_y_continuous(limits = c(-5, 15)) +
                scale_x_continuous(limits = c(-5, 15)) +
```

```
  theme(plot.caption = element_text(size = 14)))
```




```
# Linear regression
summary(lm(V2 ~ V1, data = five_2))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = five_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1073 -1.0738  0.0545  1.1486  4.6516
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

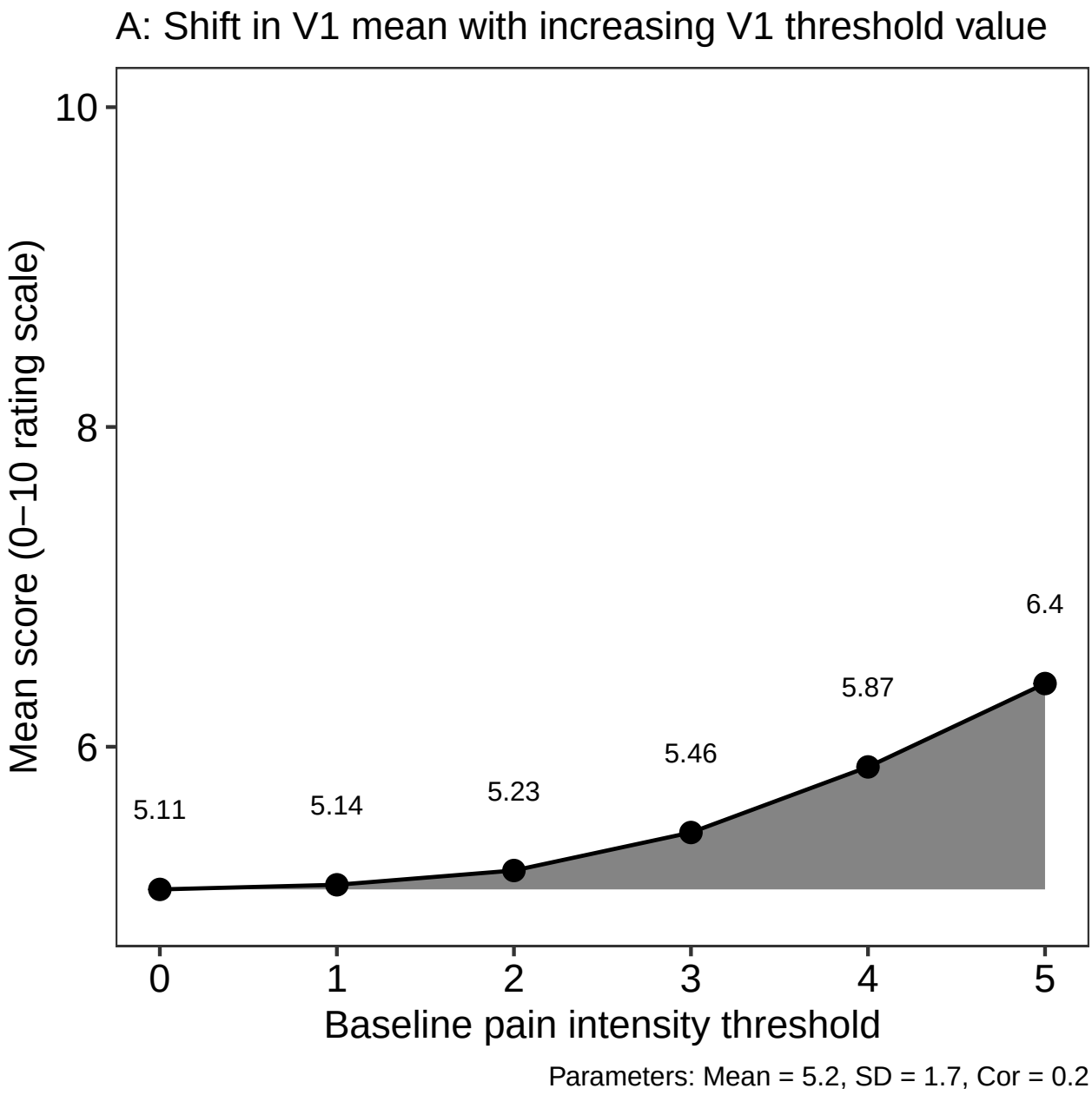
```
## (Intercept)  4.15511    0.16688  24.899  < 2e-16 ***
## V1           0.18320    0.03099   5.911 4.67e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.67 on 998 degrees of freedom
## Multiple R-squared:  0.03382,    Adjusted R-squared:  0.03286
## F-statistic: 34.94 on 1 and 998 DF,  p-value: 4.665e-09
```

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

```
# Extract visit 1 data
five_2V1 <- five_2$V1

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

# Generate a vector of V1 means at each V1 threshold
five_2V1.shift <- sapply(cutoff, function(x){mean(five_2V1[five_2V1 > x])})

# Calculate deviation
(five_2V1.df <- data.frame(cutoff = cutoff,
                           mean = five_2V1.shift) %>%
        mutate(deviation = mean - mean(five_2V1)))
```

```
##   cutoff     mean  deviation
## 1      0 5.107766 0.00000000
## 2      1 5.136723 0.02895706
## 3      2 5.225986 0.11822051
## 4      3 5.463707 0.35594105
## 5      4 5.873839 0.76607343
## 6      5 6.395213 1.28744682
```

```
# Plot data
ggplot(data = five_2V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_2V1), 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 threshold value',
         caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



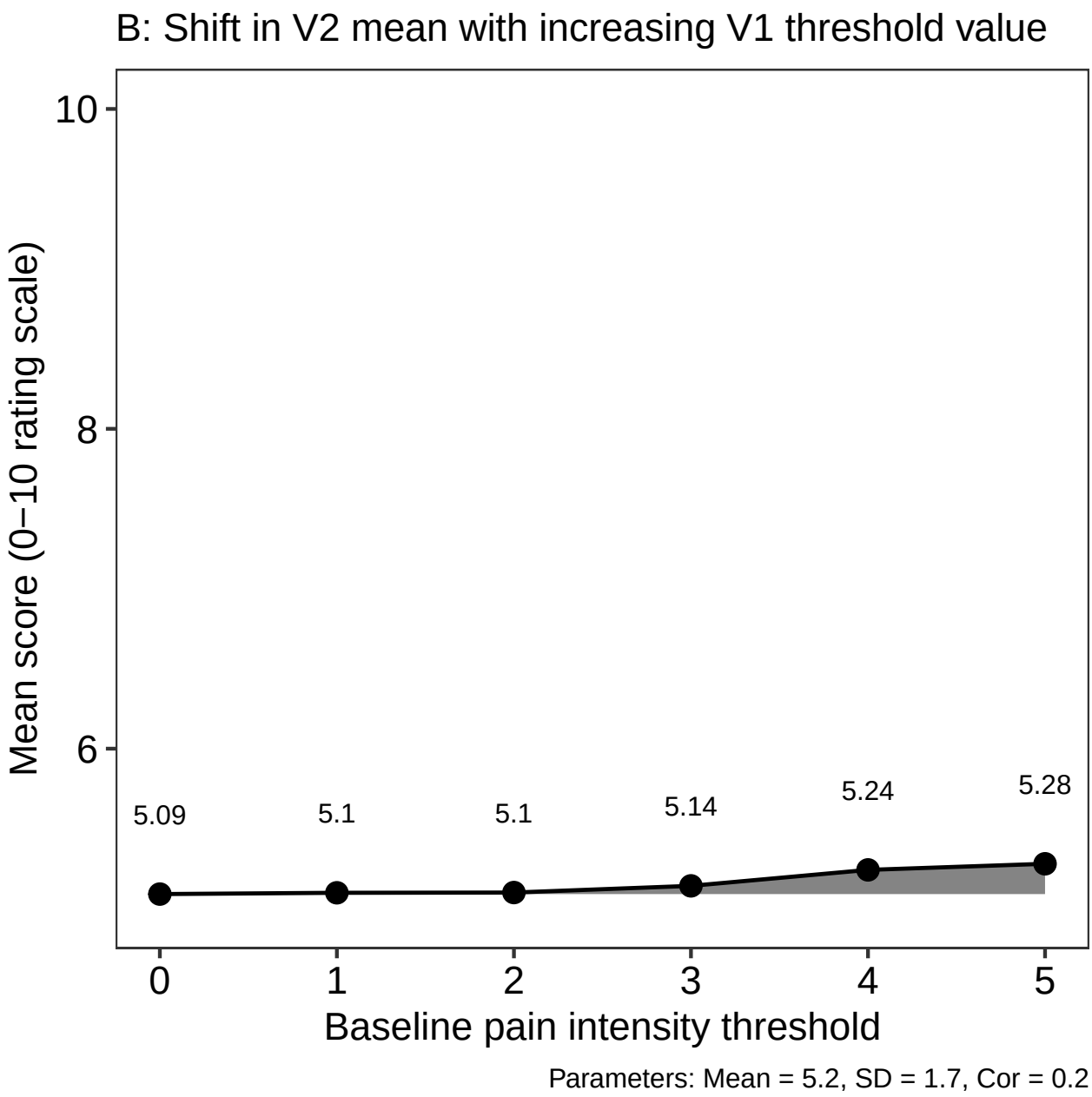
Model mean of V2 with increasing V1 thresholds from 0 to 5

```
# Extract visit 2 data
five_2V2 <- five_2$V2

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

# Generate a vector of V2 means at each V1 threshold
five_2V2.shift <- map_dbl(.x = cutoff,
                          ~ five_2 %>%
                              filter(V1 > .x) %>%
                              .$V2 %>%
```

```
                              mean(.))

# Calculate deviation
(five_2V2.df <- data.frame(cutoff = cutoff,
                           mean = five_2V2.shift) %>%
        mutate(deviation = mean - mean(five_2V2)))
```

```
##   cutoff     mean   deviation
## 1      0 5.090824 0.000000000
## 2      1 5.098853 0.008028823
## 3      2 5.100386 0.009562021
## 4      3 5.142141 0.051317210
## 5      4 5.241978 0.151153729
## 6      5 5.280114 0.189290126
```

```
# Plot data
ggplot(data = five_2V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_2V2), 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 = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



Placebo response

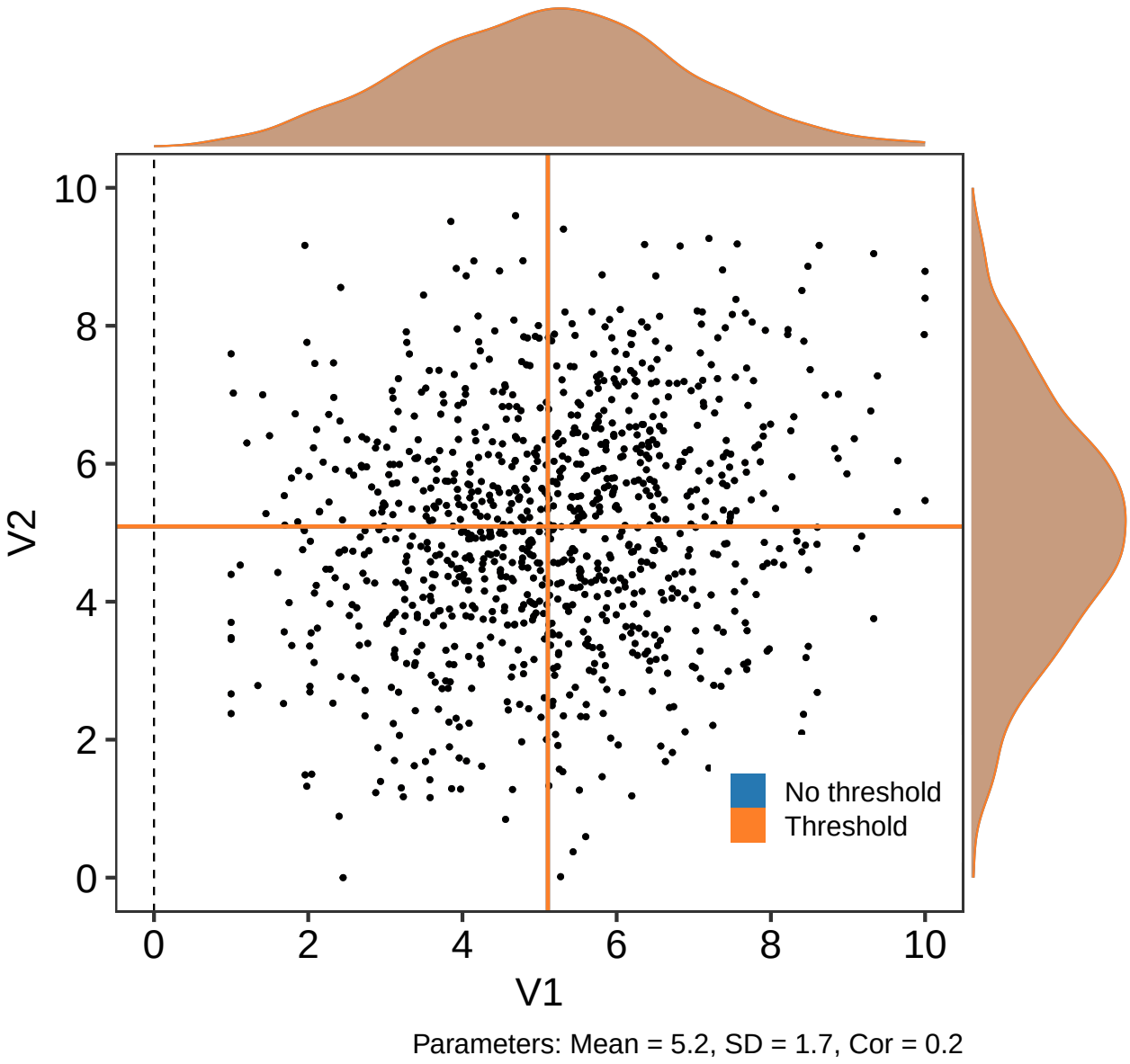
threshold: 0

```
# Process data
placebo_2.0 <- five_2 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_2.0 <- groupwiseMean(difference ~ 1,
                          data = placebo_2.0,
```

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

diff_2.0$.id <- 0

kable(diff_2.0)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.0169	0.95	-0.123	0.15

```
# Plot the data
ggMarginal(placebo_2.0[, 1:3] %>%
               bind_rows(five_2) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_2$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_2.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(five_2$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_2.0$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 threshold = 0',
                    caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



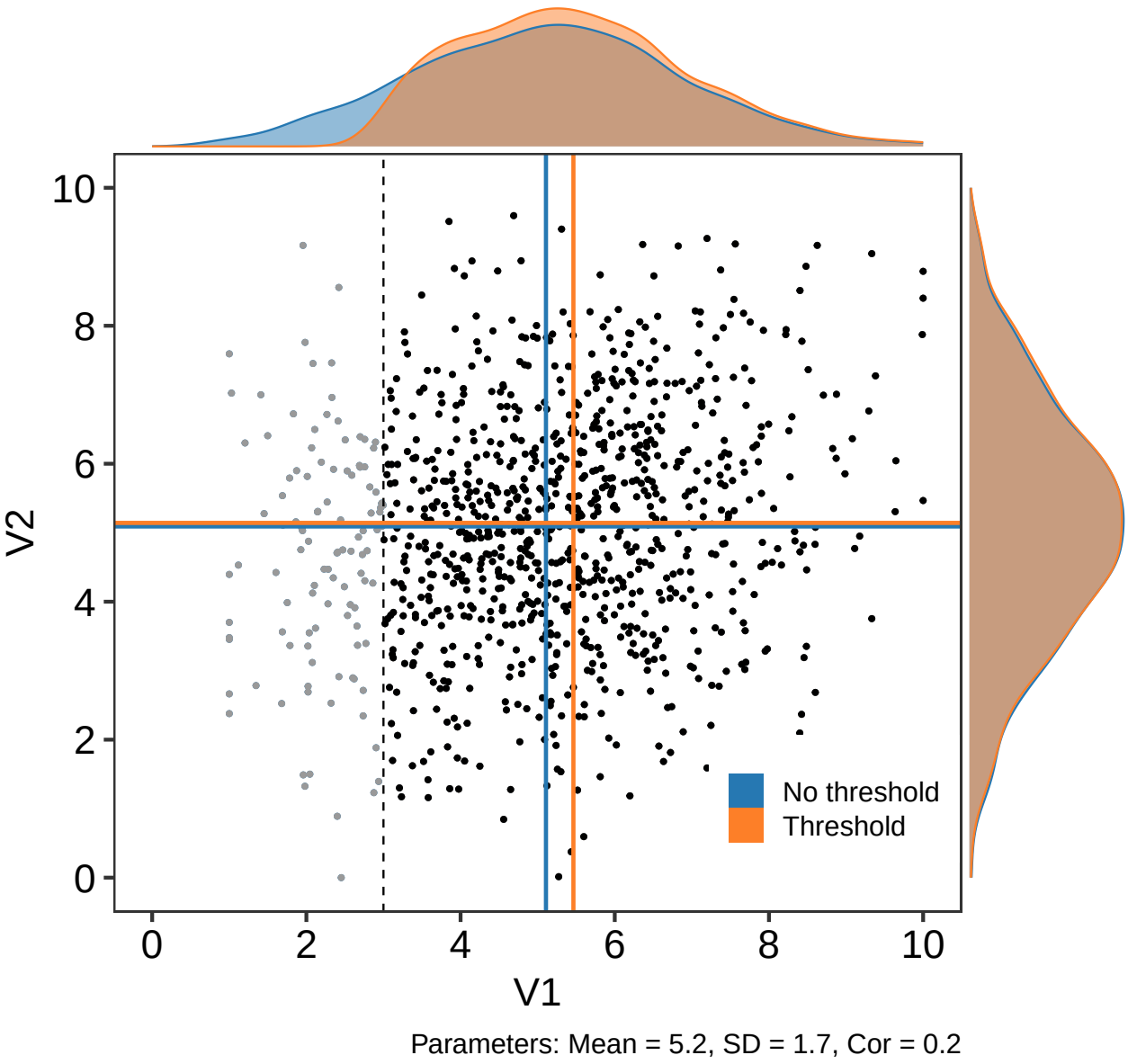
threshold: 3

```
# Process data
placebo_2.3 <- five_2 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
diff_2.3 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_2.3,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_2.3$.id <- 3

kable(diff_2.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	890	0.322	0.95	0.192	0.454

```
# Plot the data
ggMarginal(placebo_2.3[, 1:3] %>%
               bind_rows(five_2) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_2$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_2.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(five_2$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_2.3$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 threshold = 3',
                    caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



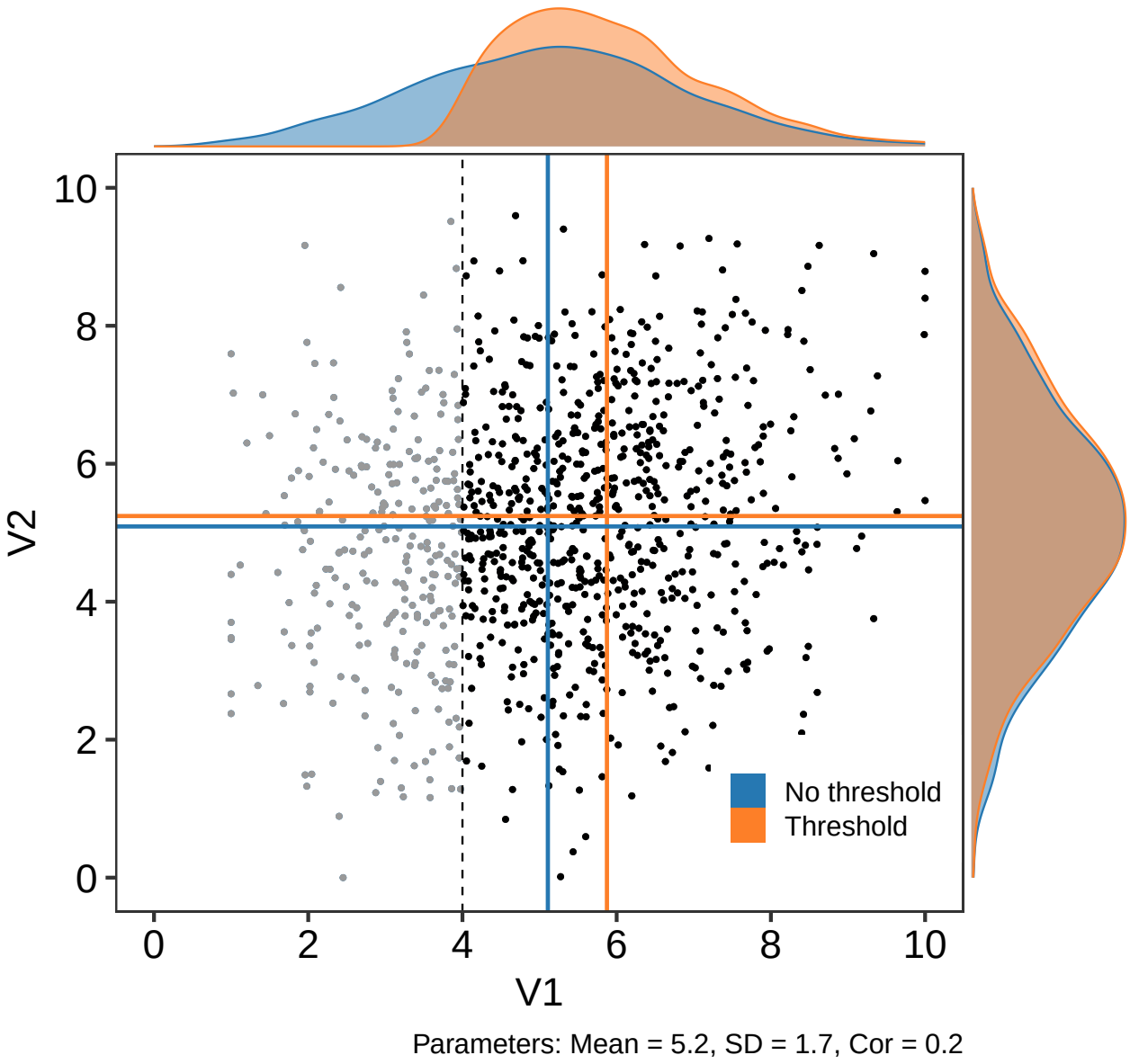
threshold: 4

```
# Process that data
placebo_2.4 <- five_2 %>%
    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_2.4 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_2.4,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_2.4$.id <- 4

kable(diff_2.4)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
4	734	0.632	0.95	0.491	0.776

```
# Plot the data
ggMarginal(placebo_2.4[, 1:3] %>%
               bind_rows(five_2) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_2$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_2.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(five_2$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_2.4$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 threshold = 4',
                    caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



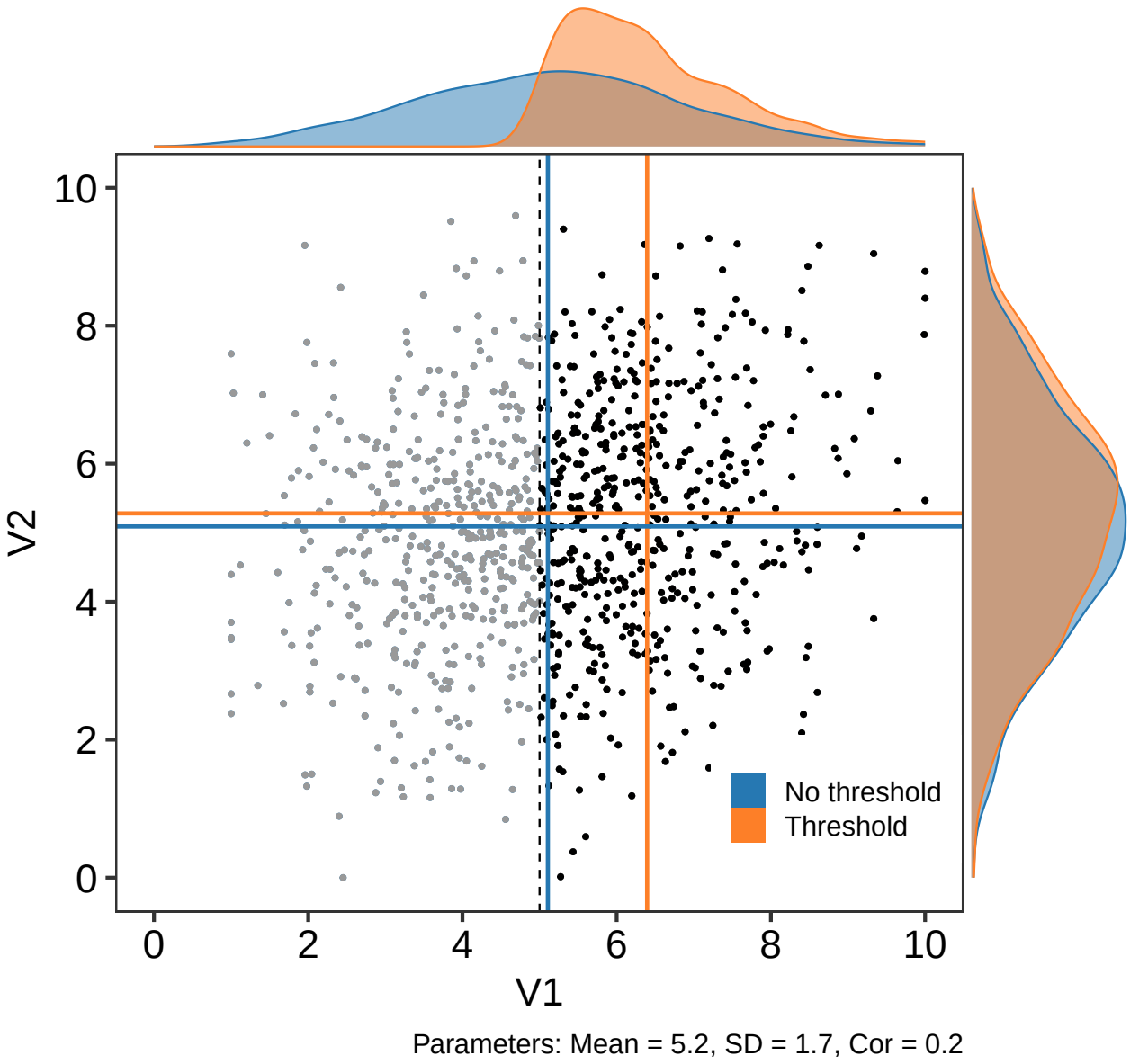
threshold: 5

```
# Process that data
placebo_2.5 <- five_2 %>%
    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
diff_2.5 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_2.5,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_2.5$.id <- 5

kable(diff_2.5)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	530	1.12	0.95	0.958	1.28

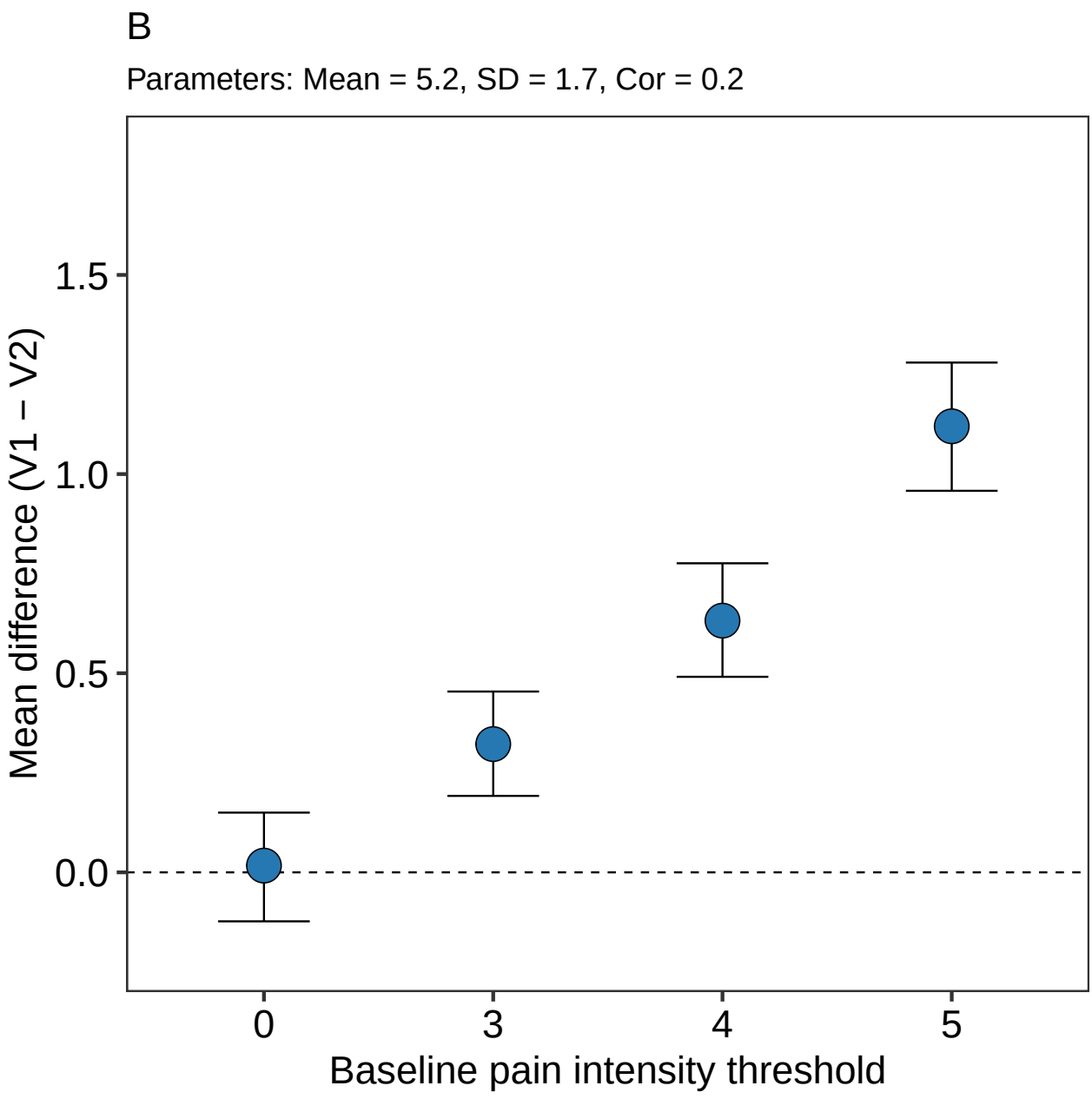
```
# Plot the data
ggMarginal(placebo_2.5[, 1:3] %>%
               bind_rows(five_2) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_2$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_2.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(five_2$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_2.5$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 threshold = 5',
                    caption = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



Difference plot

```
# Bind diff_*.* dataframes
diff_all_2 <- diff_2.0 %>%
  bind_rows(diff_2.3, diff_2.4, diff_2.5)

pp_2 <- diff_all_2 %>%
  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 = 'Parameters: Mean = 5.2, SD = 1.7, Cor = 0.2',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)') +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_2
```

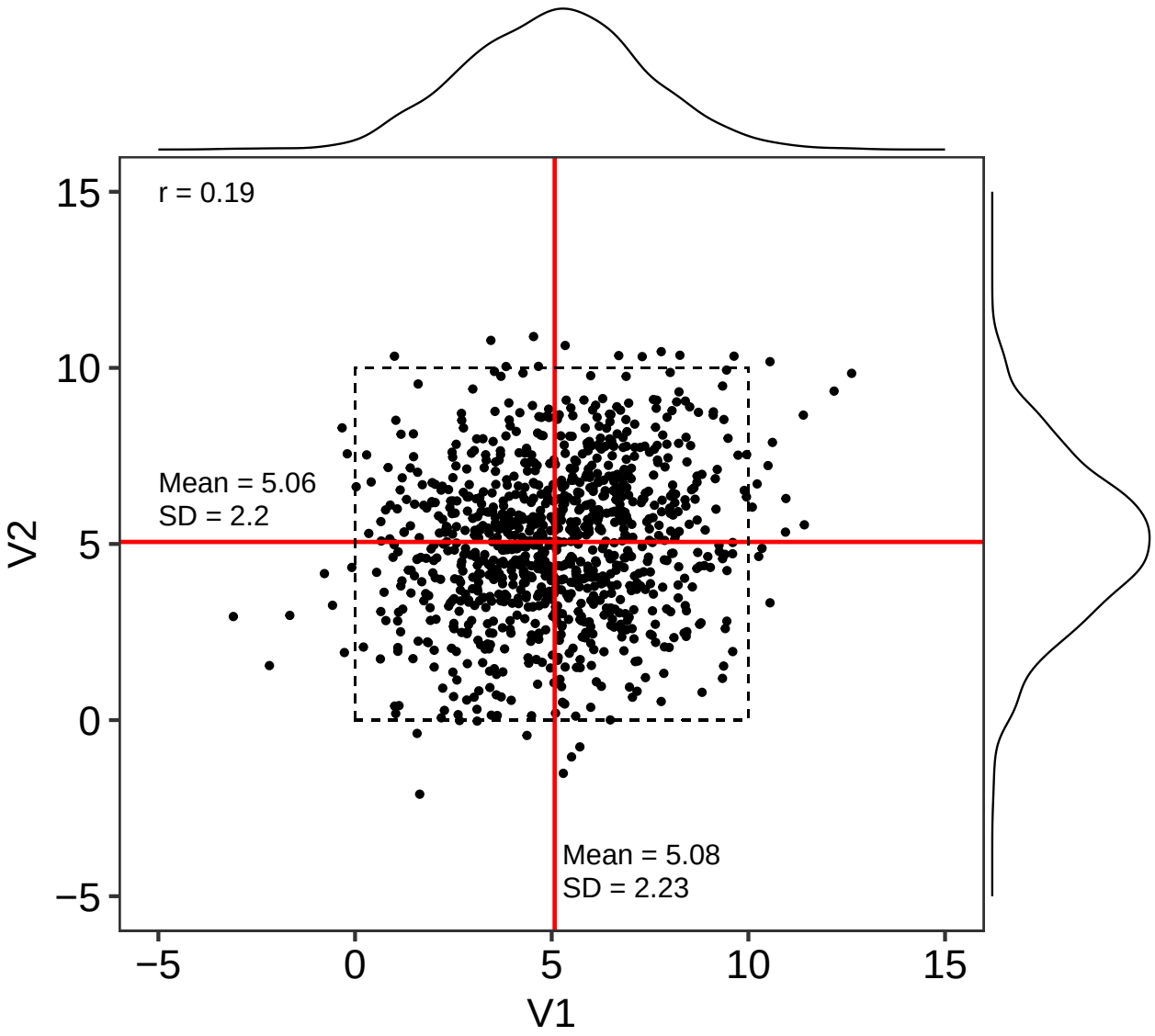


Mean = 5.2, SD = 2.2, Cor = 0.2

Generate and summarise data

Base data

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

# Generate the data
five_3.base <- as.data.frame(mvrnorm(n = 1000, mu = c(5.2, 5.2), Sigma = cov_3))

# Plot base data
ggMarginal(ggplot(data = five_3.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(five_3.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(five_3.base$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(five_3.base$V1, five_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(five_3.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(five_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(five_3.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(five_3.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(five_3.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(five_3.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(five_3.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(five_3.base$V1), 2)}")) +
               labs(title = 'A: Unconstrained',
                    caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(plot.caption = element_text(size = 14)))
```




```
# Linear regression
summary(lm(V2 ~ V1, data = five_3.base))
```

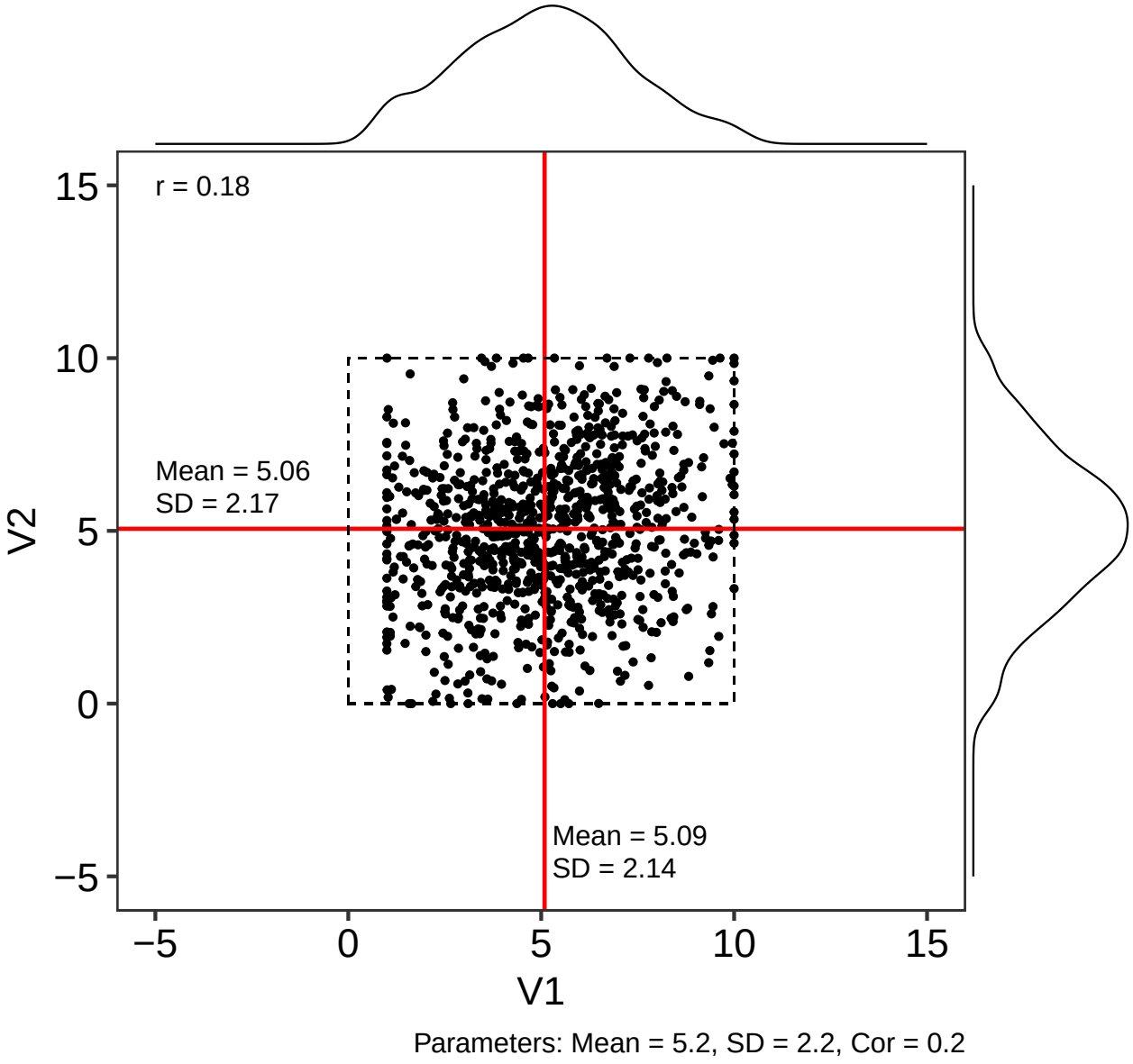
```
##
## Call:
## lm(formula = V2 ~ V1, data = five_3.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6101 -1.3905  0.0722  1.4859  6.0241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.12214    0.16981  24.274  < 2e-16 ***
```

```
## V1           0.18442    0.03063   6.021 2.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.162 on 998 degrees of freedom
## Multiple R-squared:  0.03506,    Adjusted R-squared:  0.03409
## F-statistic: 36.26 on 1 and 998 DF,  p-value: 2.427e-09
```

Constrain values to 0-10 range

```
# Process data
five_3 <- five_3.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)) %>%
    mutate(group = 'No threshold')

# Plot processed data
ggMarginal(ggplot(data = five_3) +
                aes(x = V1, y = V2) +
                geom_point() +
                geom_hline(yintercept = mean(five_3$V2),
                           colour = 'red', size = 1) +
                geom_vline(xintercept = mean(five_3$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(five_3$V1, five_3$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_3$V2) + 1.7,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_3$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = mean(five_3$V2) + 0.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_3$V2),2)}")) +
                annotate(geom = 'text', x = mean(five_3$V1) + 0.2, y = -3.8,
                         hjust = 0, size = 5,
                         label = str_glue("Mean = {round(mean(five_3$V1), 2)}")) +
                annotate(geom = 'text', x = mean(five_3$V1) + 0.2, y = -4.75,
                         hjust = 0, size = 5,
                         label = str_glue("SD = {round(sd(five_3$V1), 2)}")) +
                labs(title = 'B: Constrained (0-10 range)',
                     caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
                scale_y_continuous(limits = c(-5, 15)) +
```

```
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))
```




```
# Linear regression
summary(lm(V2 ~ V1, data = five_3))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = five_3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.314 -1.390  0.068  1.486  5.694
##
## Coefficients:
```

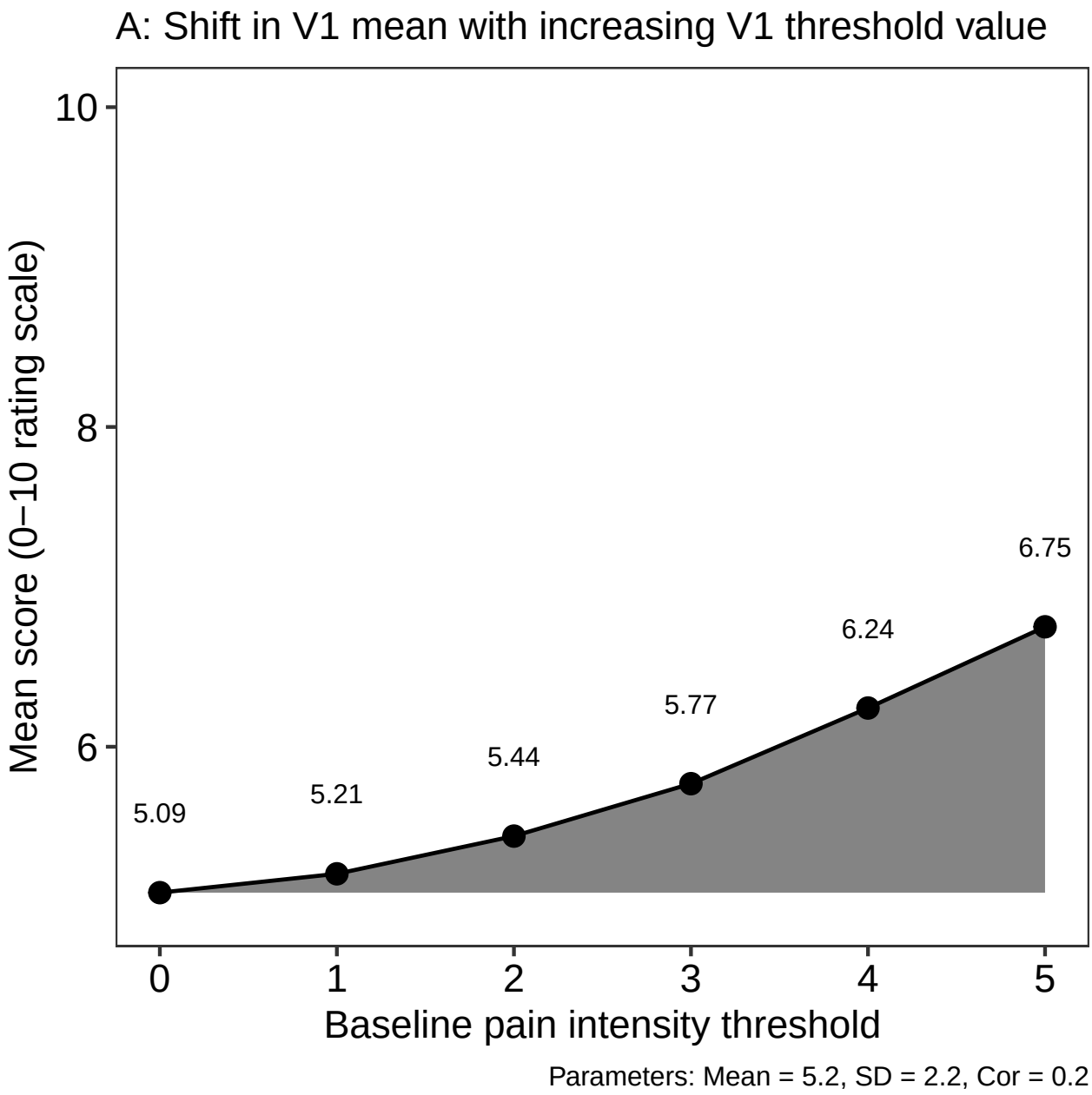
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.12068    0.17414  23.663  < 2e-16 ***
## V1           0.18457    0.03155   5.851 6.64e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.137 on 998 degrees of freedom
## Multiple R-squared:  0.03316,    Adjusted R-squared:  0.03219
## F-statistic: 34.23 on 1 and 998 DF,  p-value: 6.638e-09
```

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

```
# Extract visit 1 data
five_3V1 <- five_3$V1

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

# Generate a vector of V1 means at each V1 threshold
five_3V1.shift <- sapply(cutoff, function(x){mean(five_3V1[five_3V1 > x])})

# Calculate deviation
(five_3V1.df <- data.frame(cutoff = cutoff,
                           mean = five_3V1.shift) %>%
        mutate(deviation = mean - mean(five_3V1)))
```

```
##   cutoff     mean deviation
## 1      0 5.087557 0.0000000
## 2      1 5.205305 0.1177486
## 3      2 5.441239 0.3536825
## 4      3 5.769110 0.6815530
## 5      4 6.242234 1.1546772
## 6      5 6.750182 1.6626254
```

```
# Plot data
ggplot(data = five_3V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_3V1), 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 threshold value',
         caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



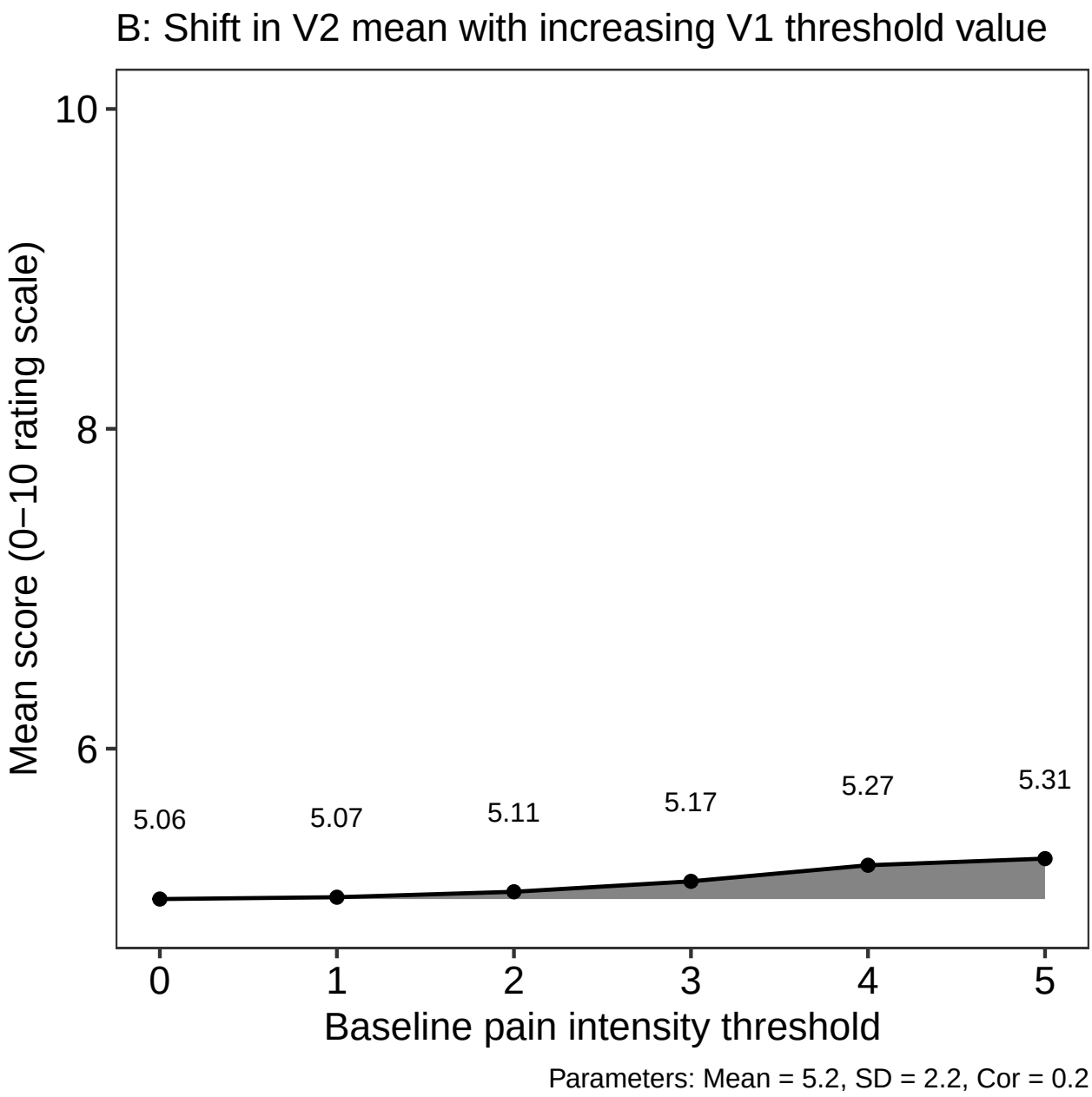
Model mean of V2 with increasing V1 thresholds from 0 to 5

```
# Extract visit 2 data
five_3V2 <- five_3$V2

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

# Generate a vector of V2 means at each V1 threshold
five_3V2.shift <- map_dbl(.x = cutoff,
                          ~ five_3 %>%
                              filter(V1 > .x) %>%
                              .$V2 %>%
```

```
                              mean(.))

# Calculate deviation
(five_3V2.df <- data.frame(cutoff = cutoff,
                           mean = five_3V2.shift) %>%
        mutate(deviation = mean - mean(five_3V2)))
```

```
##   cutoff     mean  deviation
## 1      0 5.059701 0.00000000
## 2      1 5.071221 0.01151964
## 3      2 5.105124 0.04542296
## 4      3 5.170639 0.11093797
## 5      4 5.271064 0.21136221
## 6      5 5.312531 0.25282924
```

```
# Plot data
ggplot(data = five_3V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(five_3V2), ymax = mean) +
    geom_ribbon(alpha = 0.6) +
    geom_point(size = 3) +
    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 = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



Placebo response

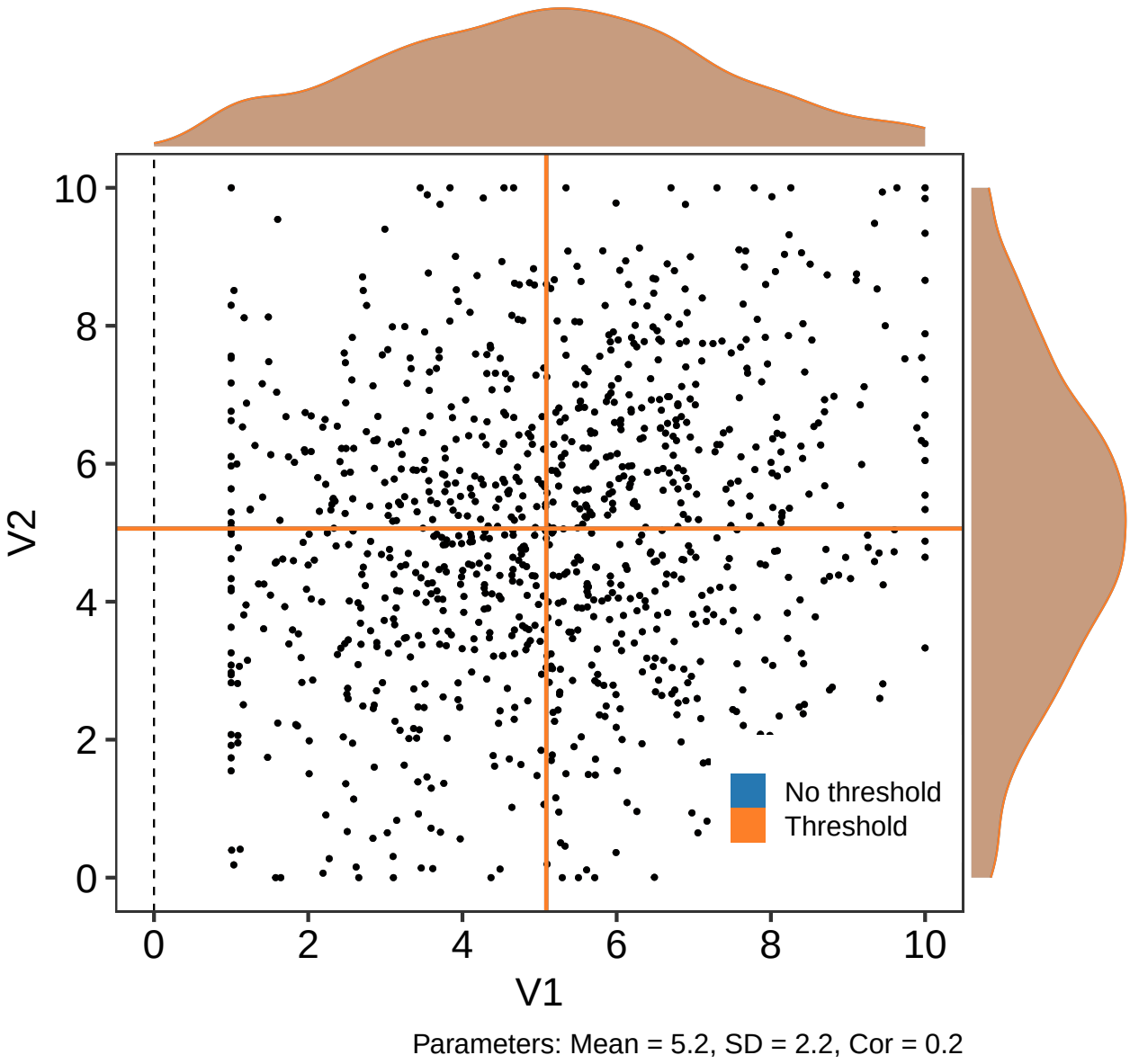
threshold: 0

```
# Process data
placebo_3.0 <- five_3 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Calculate the mean (95%CI) difference between V1 and V2
diff_3.0 <- groupwiseMean(difference ~ 1,
                          data = placebo_3.0,
```

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

diff_3.0$.id <- 0

kable(diff_3.0)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.0279	0.95	-0.15	0.197

```
# Plot the data
ggMarginal(placebo_3.0[, 1:3] %>%
               bind_rows(five_3) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_3$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_3.0$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(five_3$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_3.0$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 threshold = 0',
                    caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



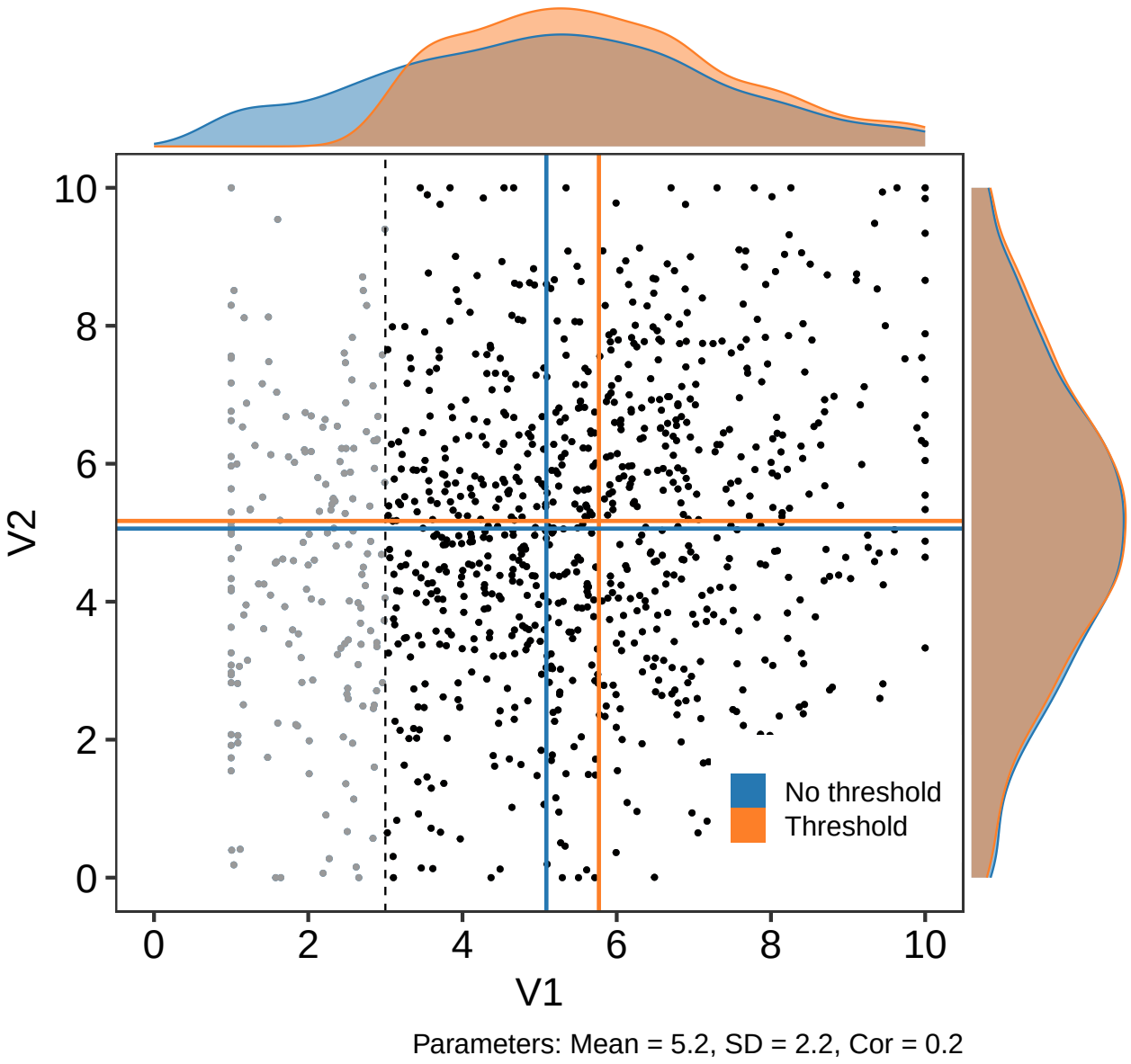
threshold: 3

```
# Process data
placebo_3.3 <- five_3 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Threshold')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
diff_3.3 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_3.3,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_3.3$.id <- 3

kable(diff_3.3)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
3	821	0.598	0.95	0.428	0.774

```
# Plot the data
ggMarginal(placebo_3.3[, 1:3] %>%
               bind_rows(five_3) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_3$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_3.3$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(five_3$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_3.3$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 threshold = 3',
                    caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



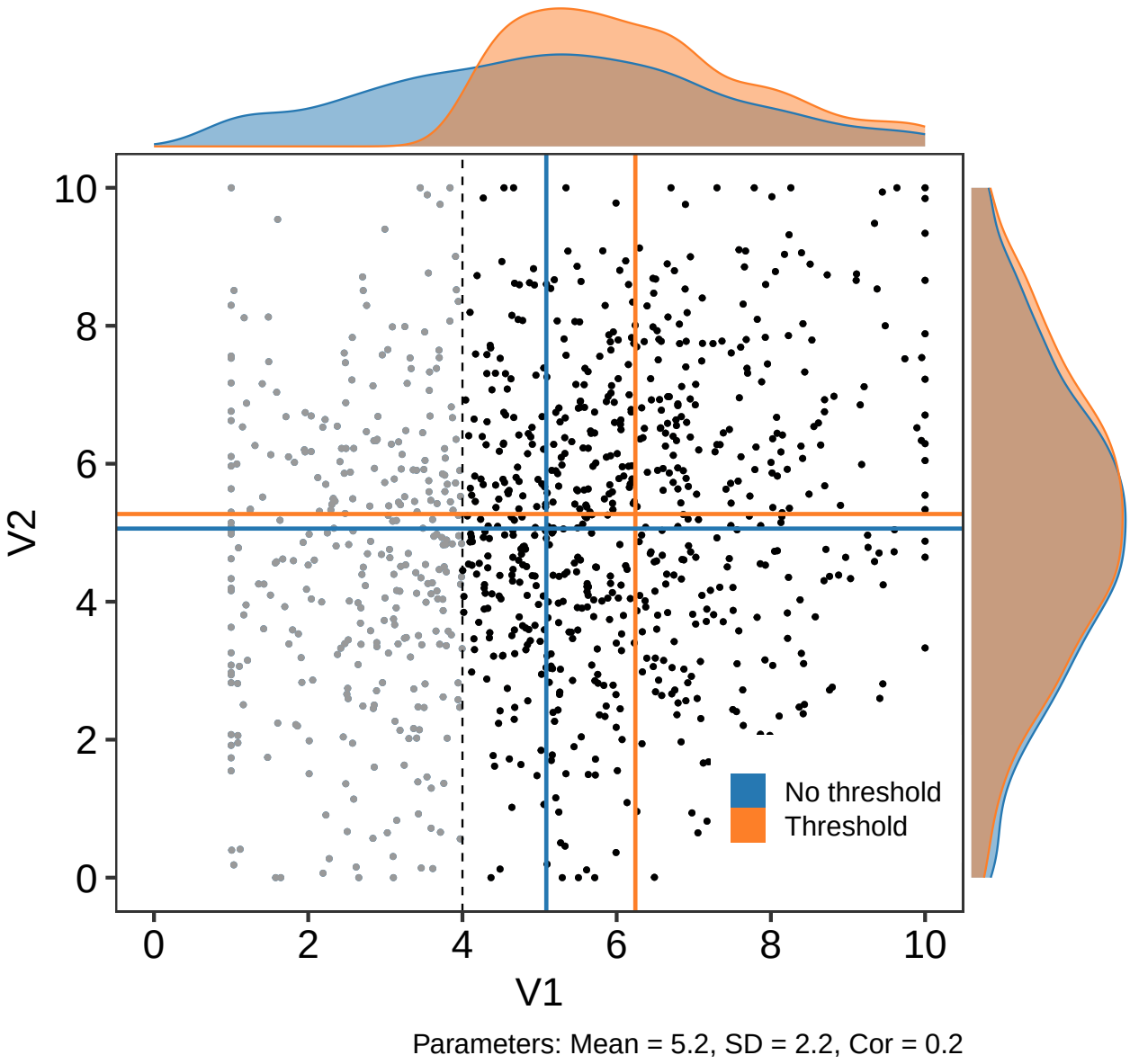
threshold: 4

```
# Process that data
placebo_3.4 <- five_3 %>%
    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_3.4 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_3.4,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_3.4$.id <- 4

kable(diff_3.4)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
4	678	0.971	0.95	0.782	1.17

```
# Plot the data
ggMarginal(placebo_3.4[, 1:3] %>%
               bind_rows(five_3) %>%
               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,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_3$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_3.4$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(five_3$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_3.4$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 threshold = 4',
                    caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



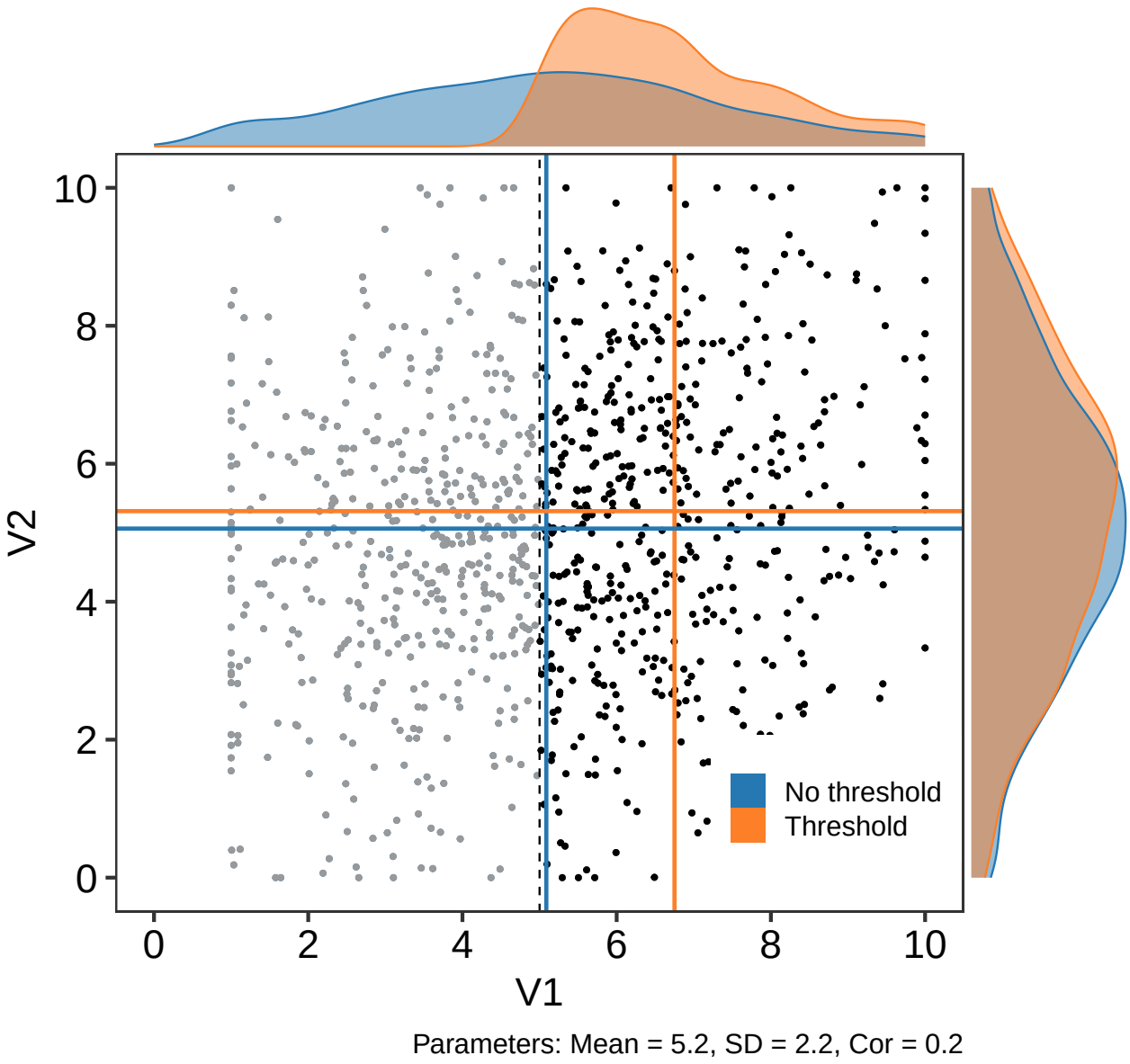
threshold: 5

```
# Process that data
placebo_3.5 <- five_3 %>%
    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
diff_3.5 <- groupwiseMean(difference ~ 1,
```

```
                        data = placebo_3.5,
                        R = 2000,
                        traditional = FALSE,
                        bca = TRUE)

diff_3.5$.id <- 5

kable(diff_3.5)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	523	1.44	0.95	1.24	1.65

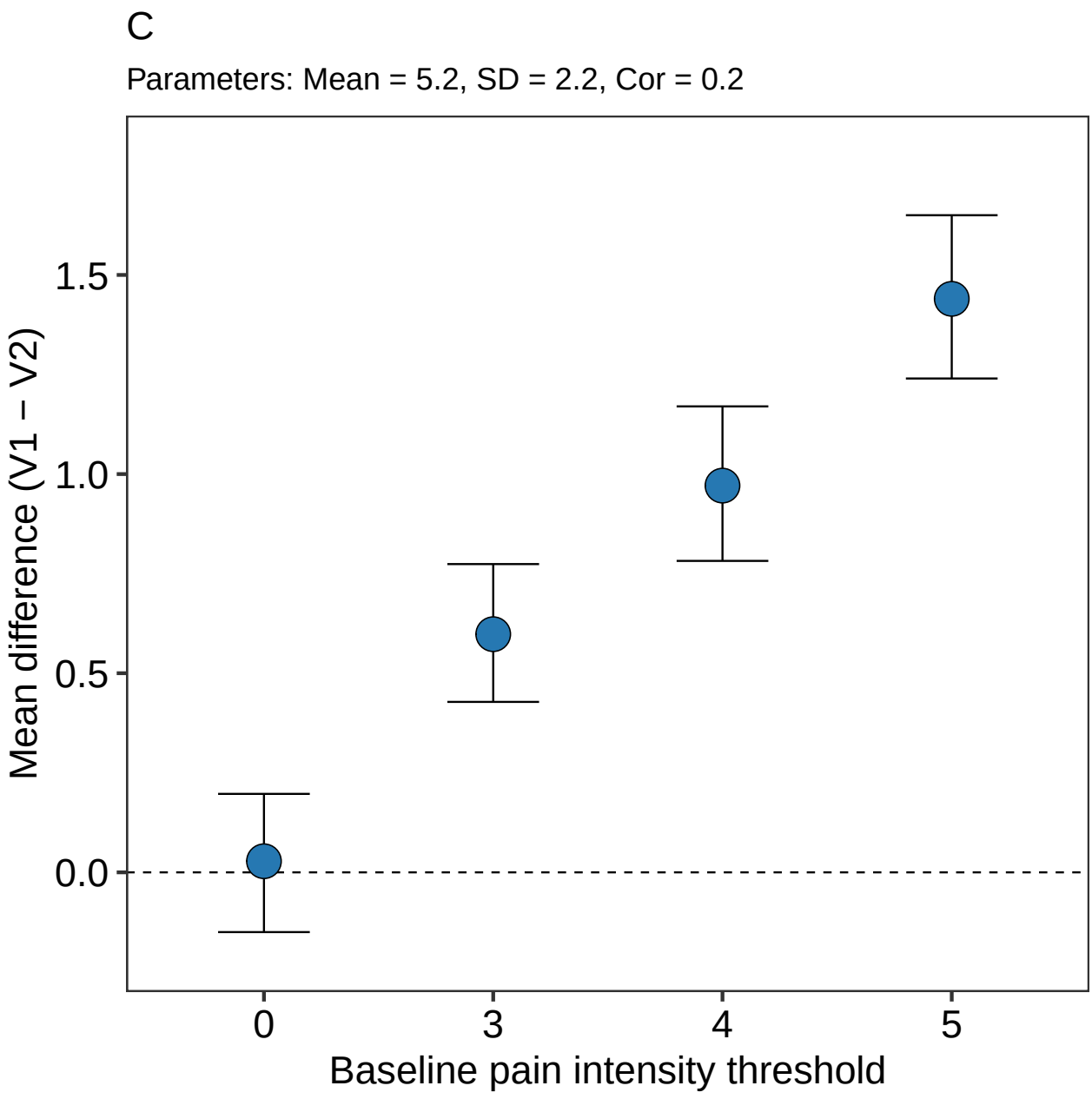
```
# Plot the data
ggMarginal(placebo_3.5[, 1:3] %>%
               bind_rows(five_3) %>%
               mutate(group = factor(group,
                                     levels = c('No threshold', 'Threshold'),
                                     ordered = TRUE)) %>%
               ggplot(data = .) +
               aes(x = V1, y = V2) +
               geom_point(aes(colour = group, fill = group),
                          size = 1,
                          key_glyph = draw_key_rect) +
               geom_point(data = five_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(five_3$V1),
                          colour = pal[1], size = 1) +
               geom_vline(xintercept = mean(placebo_3.5$V1),
                          colour = pal[2], size = 1) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(five_3$V2),
                          colour = pal[1], size = 1) +
               geom_hline(yintercept = mean(placebo_3.5$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 threshold = 5',
                    caption = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



Difference plot

```
# Bind diff_*.* dataframes
diff_all_3 <- diff_3.0 %>%
  bind_rows(diff_3.3, diff_3.4, diff_3.5)

pp_3 <- diff_all_3 %>%
  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 = 'Parameters: Mean = 5.2, SD = 2.2, Cor = 0.2',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)') +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_3
```



Publication plot

```
pp_4 <- pp_1 + pp_2 + pp_3
ggsave('figures/5.2_0.2.png', pp_4, width = 17, height = 7)
```

Session information

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] patchwork_0.0.1  knitr_1.24       MBESS_4.6.0      ggExtra_0.8
##  [5] rcompanion_2.2.2 MASS_7.3-51.4    magrittr_1.5     forcats_0.4.0
##  [9] stringr_1.4.0    dplyr_0.8.3      purrr_0.3.2      readr_1.3.1
## [13] tidyr_0.8.3.9000 tibble_2.1.3     ggplot2_3.2.0    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-141       matrixStats_0.54.0 lubridate_1.7.4
##  [4] httr_1.4.1         tools_3.6.0        backports_1.1.4
##  [7] R6_2.4.0           nortest_1.0-4      lazyeval_0.2.2
## [10] colorspace_1.4-1   withr_2.1.2.9000   tidyselect_0.2.5
## [13] compiler_3.6.0     cli_1.1.0          rvest_0.3.4
## [16] expm_0.999-4       xml2_1.2.2         sandwich_2.5-1
## [19] labeling_0.3       scales_1.0.0       lmtest_0.9-37
## [22] mvtnorm_1.0-11     multcompView_0.1-7 digest_0.6.20
## [25] foreign_0.8-72     rmarkdown_1.14     pkgconfig_2.0.2
## [28] htmltools_0.3.6    manipulate_1.0.1   highr_0.8
## [31] rlang_0.4.0        readxl_1.3.1       rstudioapi_0.10
## [34] shiny_1.3.2        generics_0.0.2     zoo_1.8-6
## [37] jsonlite_1.6       modeltools_0.2-22  Matrix_1.2-17
## [40] Rcpp_1.0.2         DescTools_0.99.28  munsell_0.5.0
## [43] stringi_1.4.3      multcomp_1.4-10    yaml_2.2.0
## [46] plyr_1.8.4         grid_3.6.0         parallel_3.6.0
## [49] promises_1.0.1     crayon_1.3.4       miniUI_0.1.1.1
## [52] lattice_0.20-38    haven_2.1.1        splines_3.6.0
## [55] hms_0.5.0          zeallot_0.1.0      pillar_1.4.2
## [58] EMT_1.1            boot_1.3-23        codetools_0.2-16
## [61] stats4_3.6.0       glue_1.3.1         evaluate_0.14
```

```
## [64] modelr_0.1.5       vctrs_0.2.0        httpuv_1.5.1
## [67] cellranger_1.1.0   gtable_0.3.0       assertthat_0.2.1
## [70] xfun_0.8           mime_0.7           coin_1.3-0
## [73] libcoin_1.0-4      xtable_1.8-4       broom_0.5.2
## [76] later_0.8.0        survival_2.44-1.1  TH.data_1.0-10
```