Regression to the mean modeling

Mean pain rating of 7.2 at 0.5 correlation

Peter Kamerman

Last knitted: 16 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 0.5.

```
cor <- matrix(c(1, 0.5, 0.5, 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,] 1.44 0.72
## [2,] 0.72 1.44
```

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

```
##       [,1]  [,2]
## [1,] 2.890 1.445
## [2,] 1.445 2.890
```

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

```
##      [,1] [,2]
## [1,] 4.84 2.42
## [2,] 2.42 4.84
```

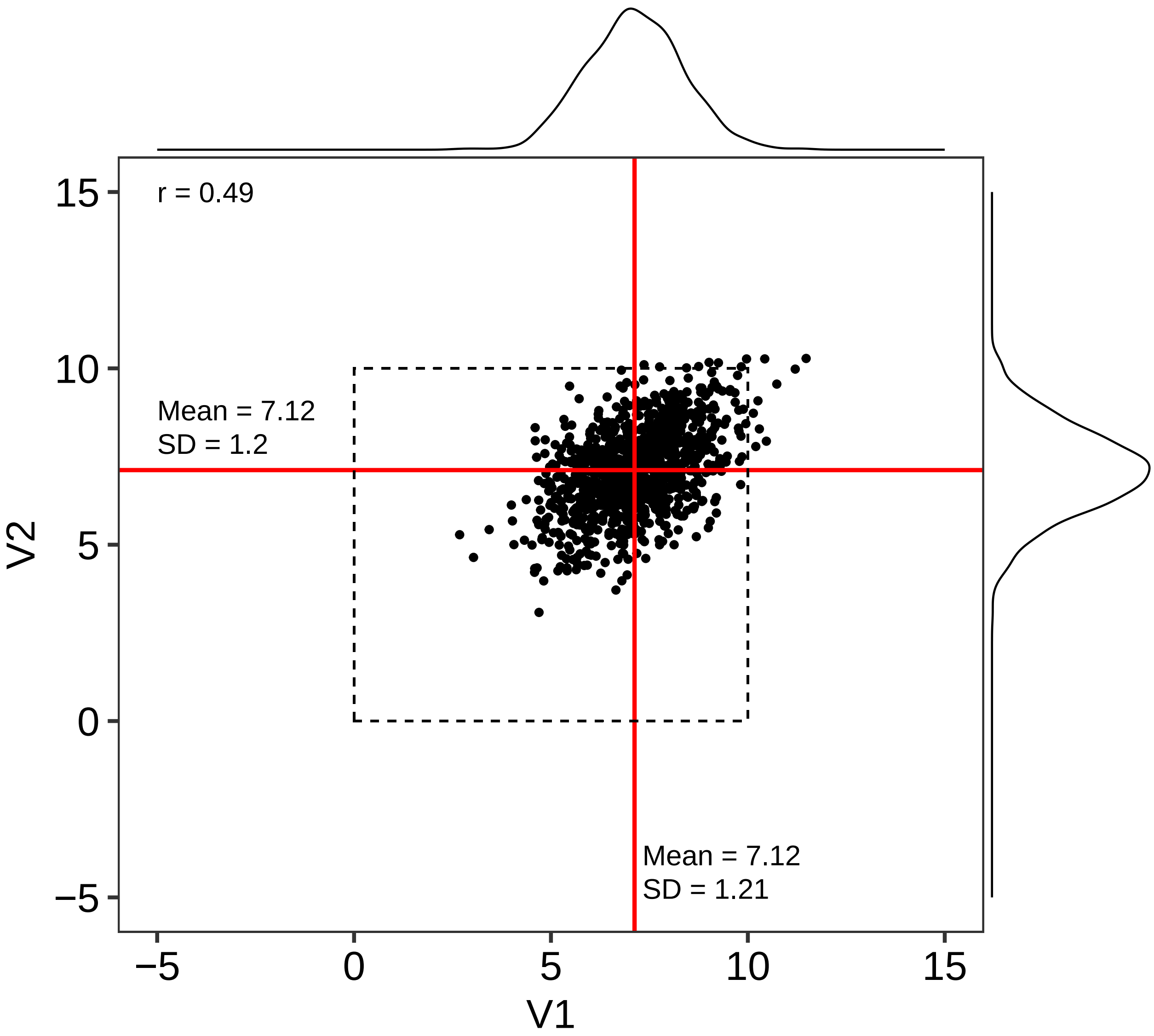
Mean = 7.2, SD = 1.2, Cor = 0.5

Generate and summarise data

Base data

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

# Generate the data
seven_1.base <- as.data.frame(mvrnorm(n = 1000, mu = c(7.2, 7.2), Sigma = cov_1))

# Plot base data
ggMarginal(ggplot(data = seven_1.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_1.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_1.base$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(seven_1.base$V1, seven_1.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_1.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_1.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_1.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_1.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_1.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_1.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_1.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_1.base$V1), 2)}")) +
               labs(title = 'A: Unconstained',
                    caption = 'Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(plot.caption = element_text(size = 14)))
```

A: Unconstained



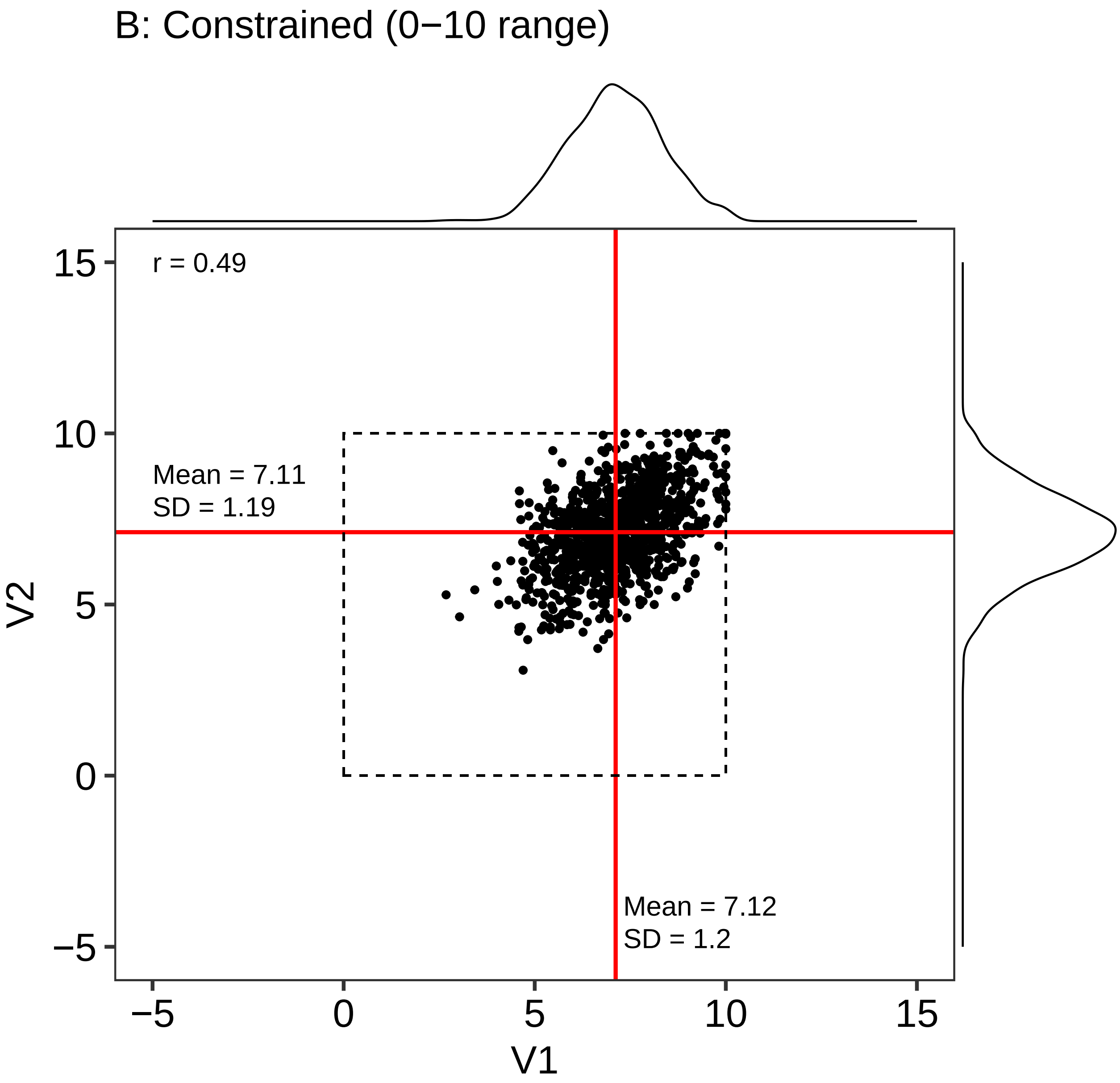
Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5

```
# Linear regression
summary(lm(V2 ~ V1, data = seven_1.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_1.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1716 -0.6956  0.0432  0.6975  3.1783
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.67152    0.19671   18.66   <2e-16 ***
## V1           0.48347    0.02723   17.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.045 on 998 degrees of freedom
## Multiple R-squared:  0.2401, Adjusted R-squared:  0.2394
## F-statistic: 315.4 on 1 and 998 DF,  p-value: < 2.2e-16
```

Constrain values to 0-10 range

```
# Process data
seven_1 <- seven_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 = seven_1) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_1$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_1$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(seven_1$V1, seven_1$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_1$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_1$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_1$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_1$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_1$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_1$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_1$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_1$V1), 2)}")) +
               labs(title = 'B: Constrained (0-10 range)',
                    caption = 'Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
    theme(plot.caption = element_text(size = 14)))
```



Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5

```
# Linear regression
summary(lm(V2 ~ V1, data = seven_1))
```

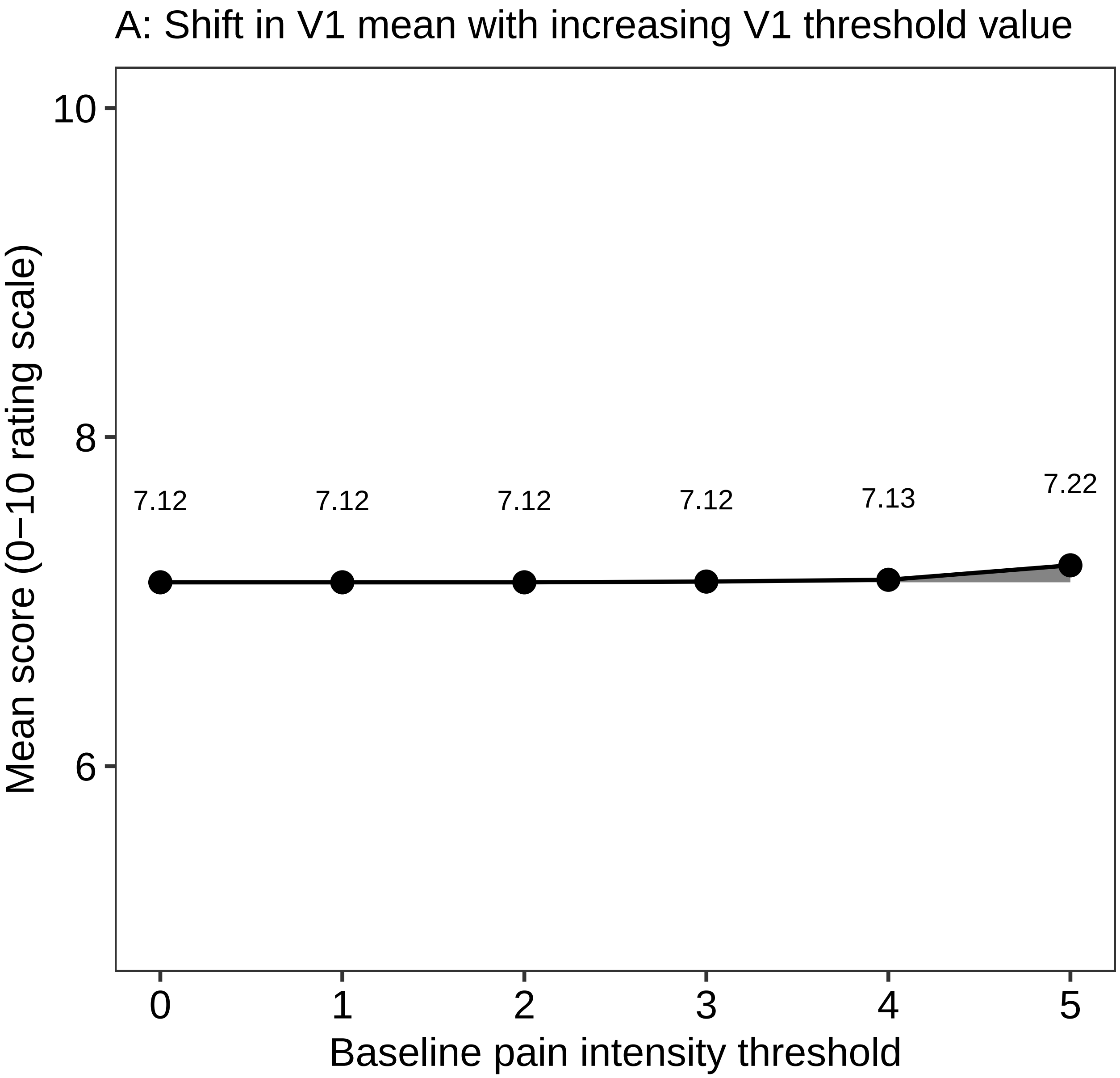
```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1725 -0.6904  0.0424  0.6992  3.1780
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.66932    0.19878   18.46   <2e-16 ***
## V1           0.48394    0.02754   17.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.045 on 998 degrees of freedom
## Multiple R-squared:  0.2363, Adjusted R-squared:  0.2355
## F-statistic: 308.8 on 1 and 998 DF,  p-value: < 2.2e-16
```

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

```
# Extract visit 1 data
seven_1V1 <- seven_1$V1

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

# Generate a vector of V1 means at each V1 threshold
seven_1V1.shift <- sapply(cutoff, function(x){mean(seven_1V1[seven_1V1 > x])})

# Calculate deviation
(seven_1V1.df <- data.frame(cutoff = cutoff,
                            mean = seven_1V1.shift) %>%
        mutate(deviation = mean - mean(seven_1V1)))
```

```
##   cutoff     mean   deviation
## 1      0 7.117263 0.000000000
## 2      1 7.117263 0.000000000
## 3      2 7.117263 0.000000000
## 4      3 7.121704 0.004440731
## 5      4 7.132653 0.015389924
## 6      5 7.220575 0.103311258
```

```
# Plot data
ggplot(data = seven_1V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 1.2, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



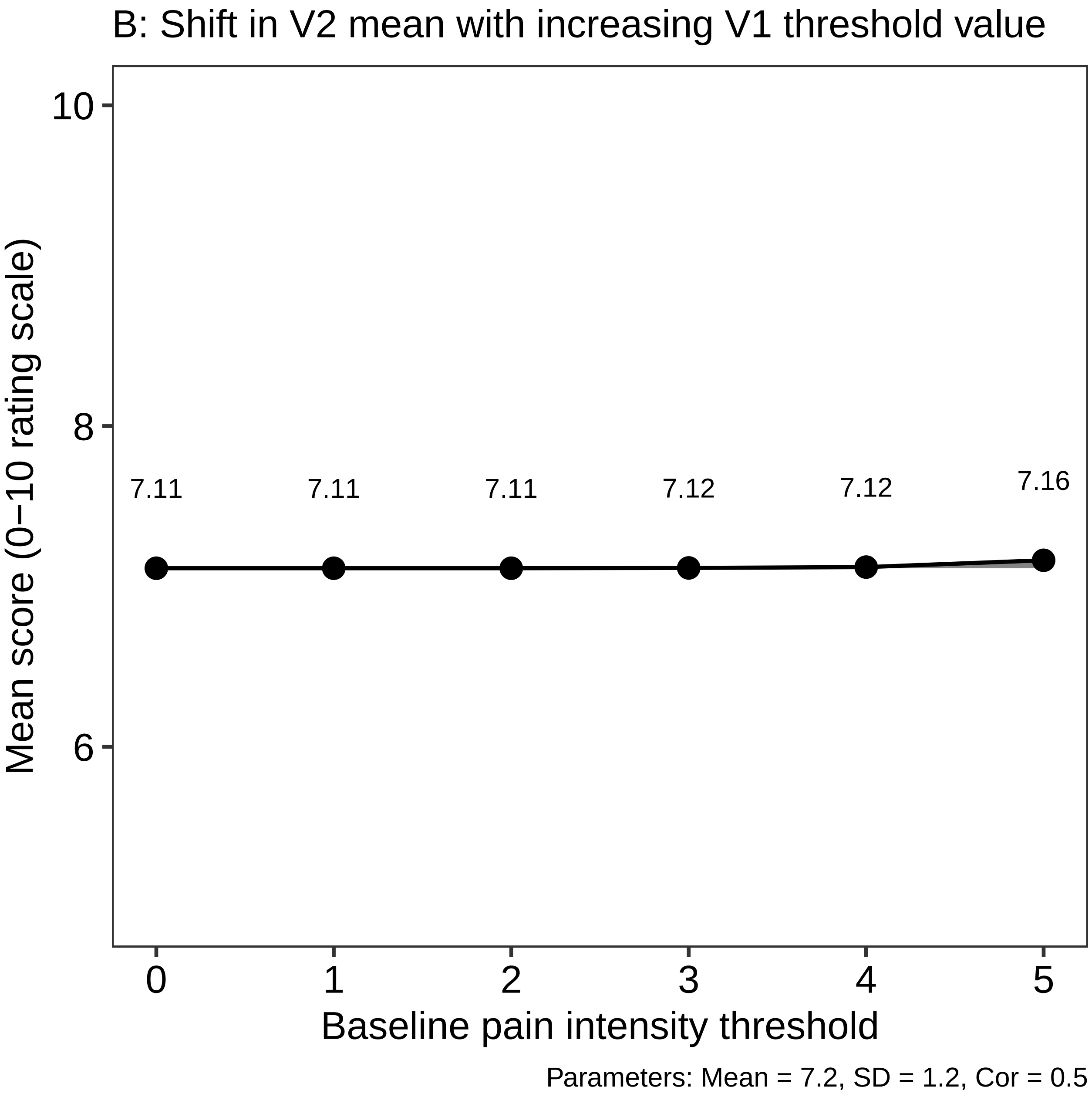
Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5

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

```
# Extract visit 2 data
seven_1V2 <- seven_1$V2

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

# Generate a vector of V2 means at each V1 threshold
seven_1V2.shift <- map_dbl(.x = cutoff,
                          ~ seven_1 %>%
                              filter(V1 > .x) %>%
                              .$V2 %>%
                              mean(.))

# Calculate deviation
(seven_1V2.df <- data.frame(cutoff = cutoff,
                           mean = seven_1V2.shift) %>%
        mutate(deviation = mean - mean(seven_1V2)))
```

```
##   cutoff     mean   deviation
## 1      0 7.113629 0.000000000
## 2      1 7.113629 0.000000000
## 3      2 7.113629 0.000000000
## 4      3 7.115461 0.001832421
## 5      4 7.120636 0.007007209
## 6      5 7.163116 0.049486903
```

```
# Plot data
ggplot(data = seven_1V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 1.2, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



Placebo response

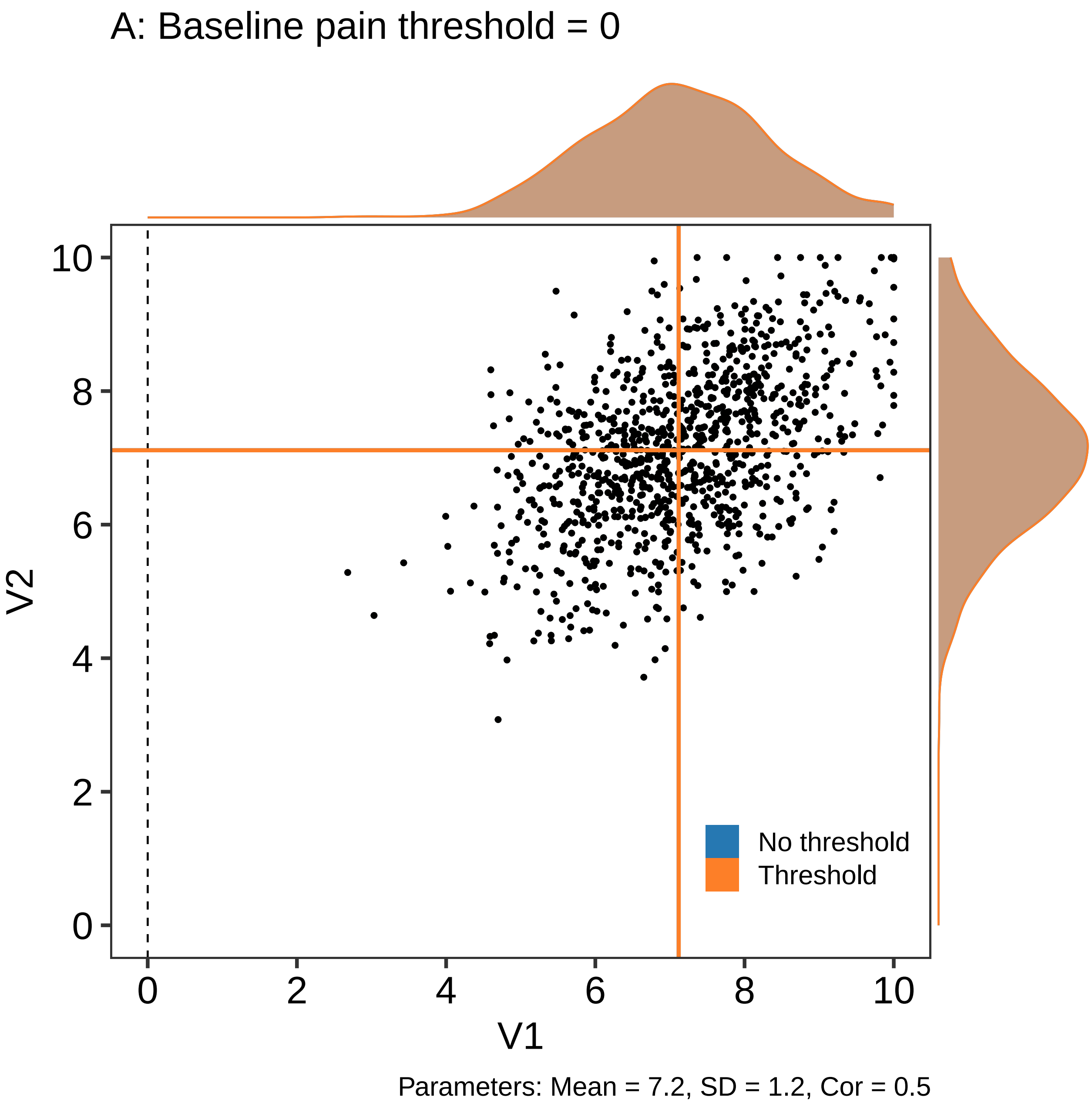
threshold: 0

```
# Process data
placebo_1.0 <- seven_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
                          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	0.00363	0.95	-0.0746	0.078

```r
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(seven_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 = seven_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

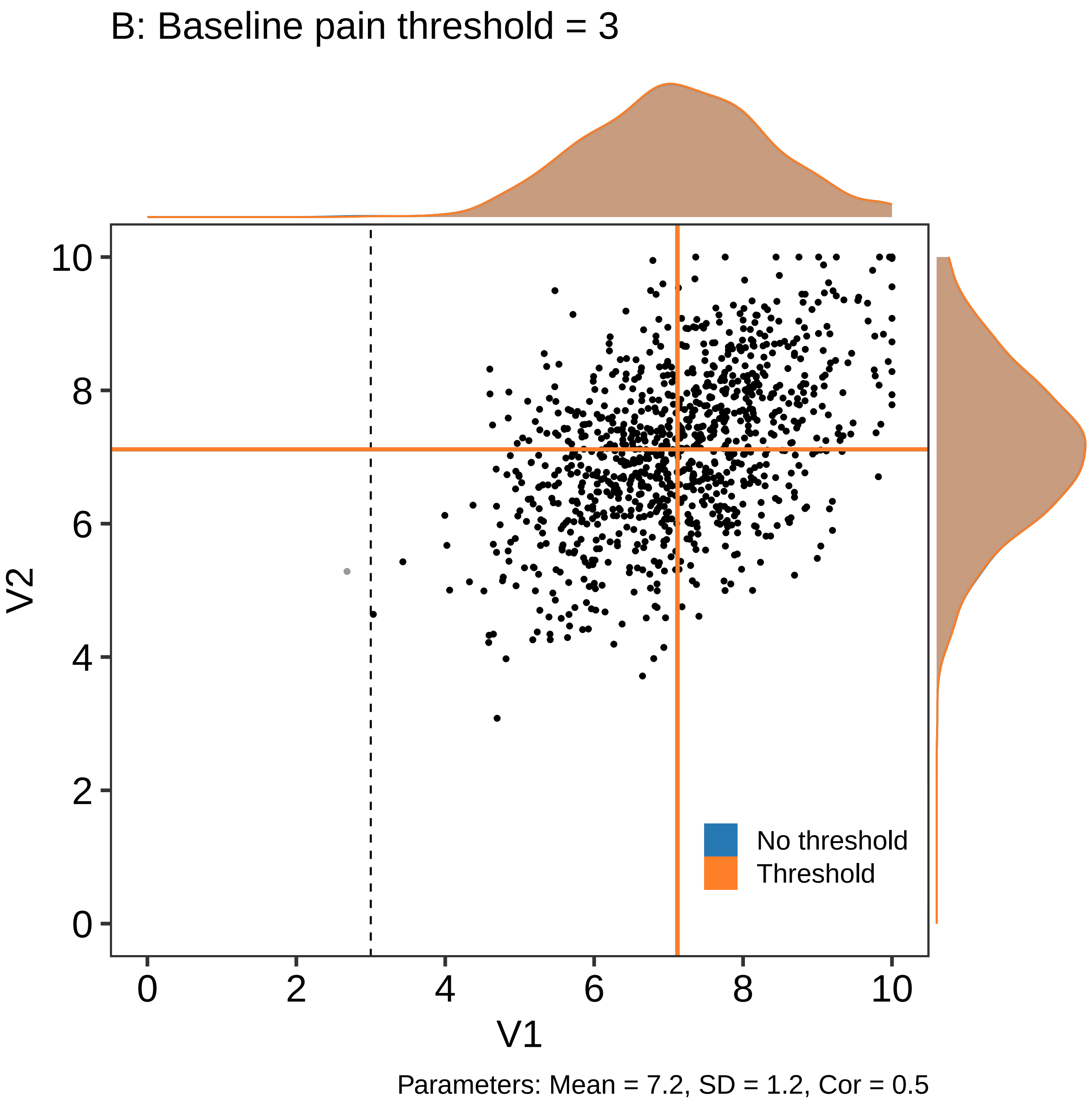


threshold: 3

```
# Process data
placebo_1.3 <- seven_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	999	0.00624	0.95	-0.0664	0.0759

```
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(seven_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 = seven_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



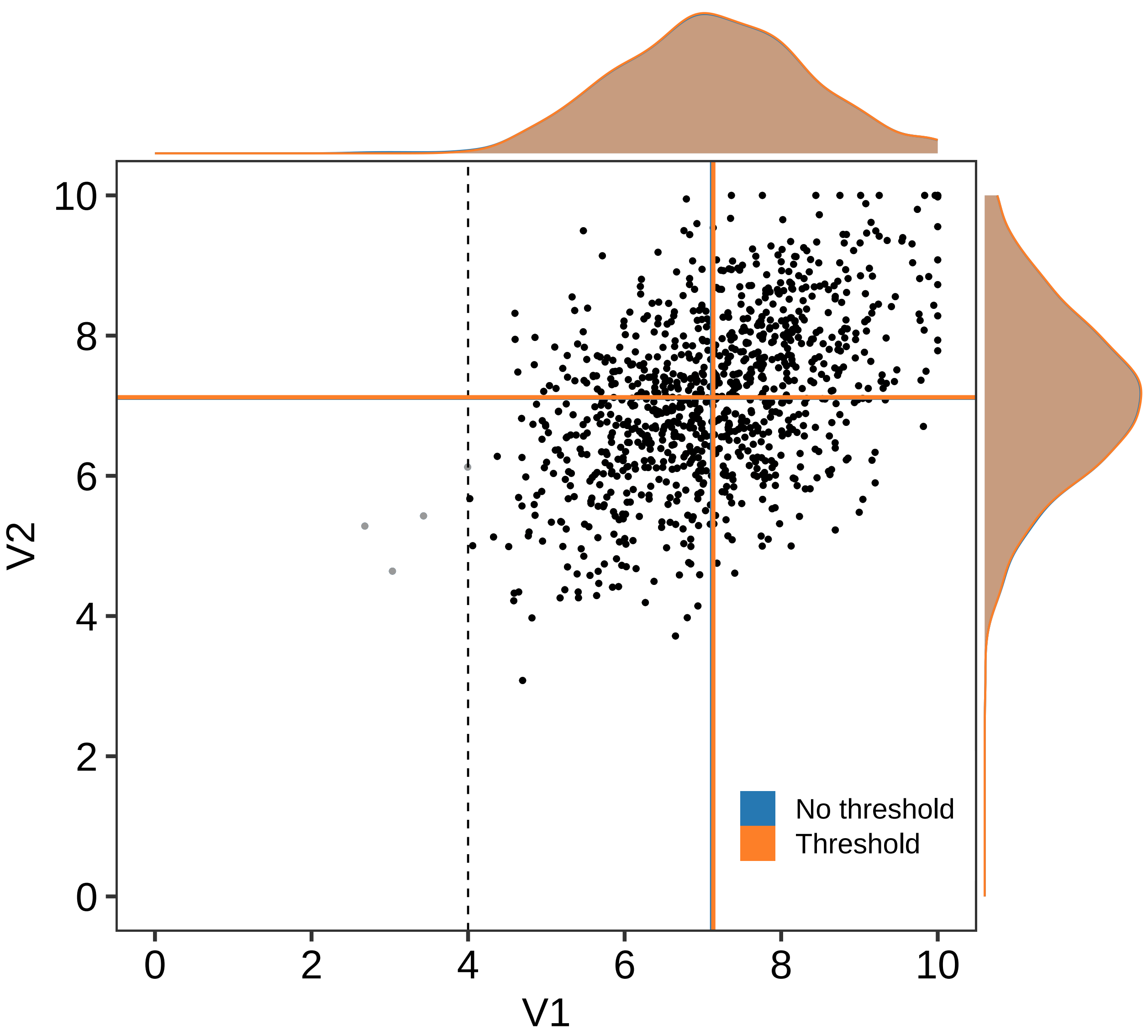
threshold: 4

```
# Process that data
placebo_1.4 <- seven_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	996	0.012	0.95	-0.0622	0.0889

```
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
               bind_rows(seven_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 = seven_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

C: Baseline pain threshold = 4



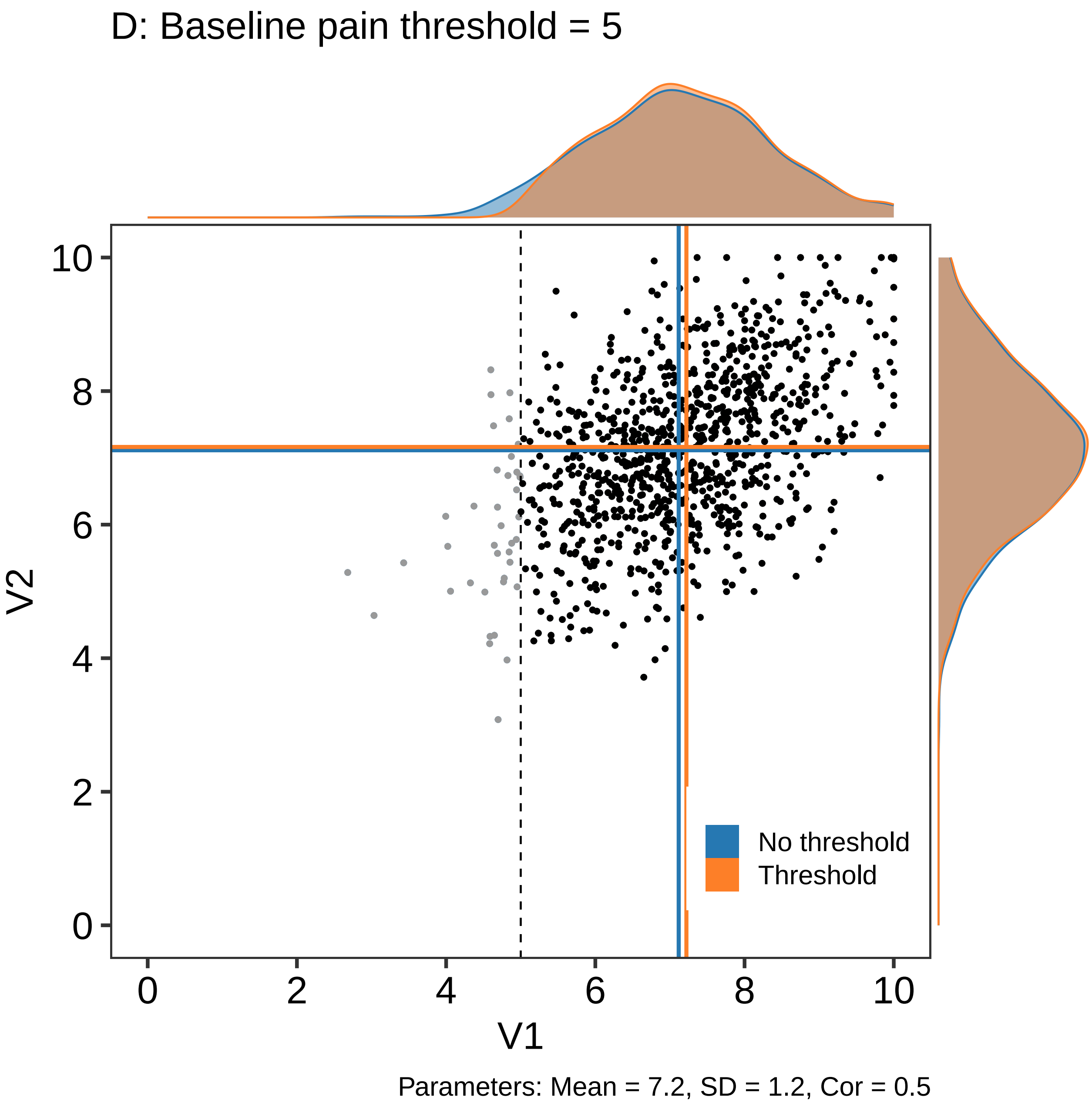
Parameters: Mean = 7.2, SD = 1.2, Cor = 0.5

threshold: 5

```
# Process that data
placebo_1.5 <- seven_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)
```

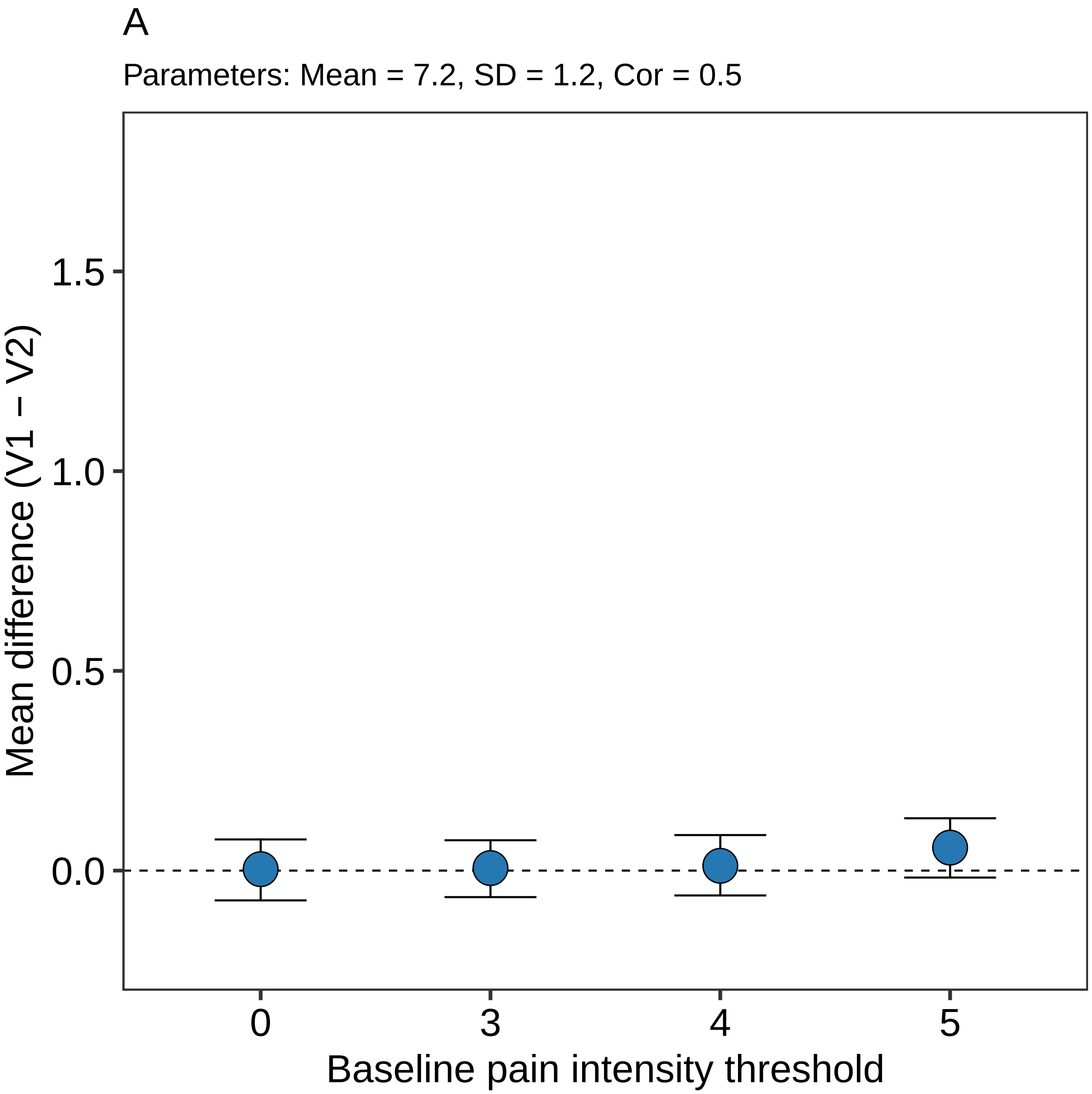
.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	961	0.0575	0.95	-0.0174	0.131

```
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(seven_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 = seven_1,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_1.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.2, Cor = 0.5') +
               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 = 7.2, SD = 1.2, Cor = 0.5',
       x = 'Baseline pain intensity threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_1
```

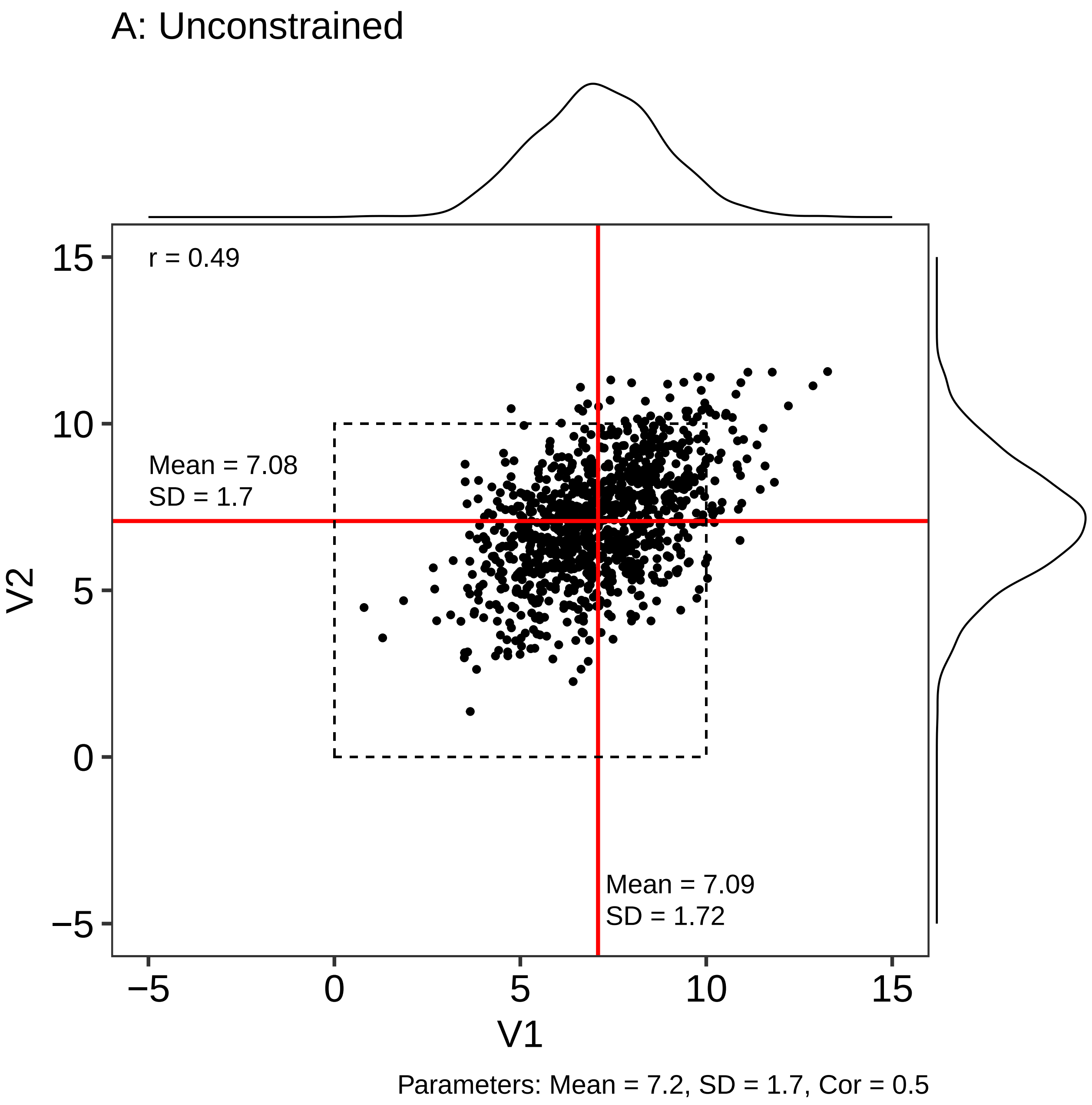


Mean = 7.2, SD = 1.7, Cor = 0.5

Generate and summarise data

Base data

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

# Generate the data
seven_2.base <- as.data.frame(mvrnorm(n = 1000, mu = c(7.2, 7.2), Sigma = cov_2))

# Plot base data
ggMarginal(ggplot(data = seven_2.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_2.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_2.base$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(seven_2.base$V1, seven_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_2.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_2.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_2.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_2.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_2.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_2.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_2.base$V1), 2)}")) +
               labs(title = 'A: Unconstrained',
                    caption = 'Parameters: Mean = 7.2, SD = 1.7, Cor = 0.5') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(plot.caption = element_text(size = 14)))
```

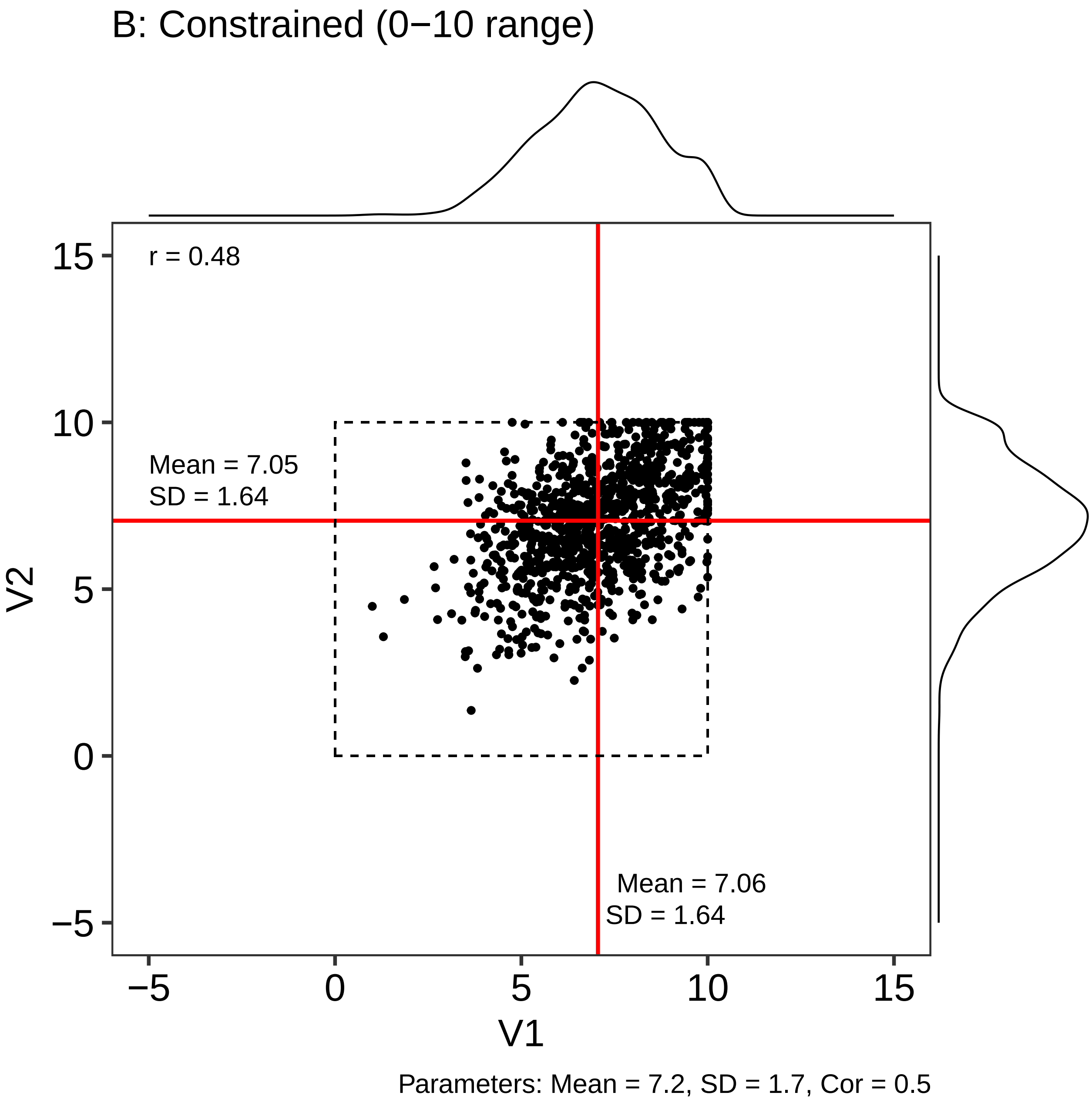


```
# Linear regression
summary(lm(V2 ~ V1, data = seven_2.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_2.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4932 -0.9854  0.0613  0.9882  4.5026
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.65173    0.19863   18.39   <2e-16 ***
## V1           0.48347    0.02723   17.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.481 on 998 degrees of freedom
## Multiple R-squared:  0.2401, Adjusted R-squared:  0.2394
## F-statistic: 315.4 on 1 and 998 DF,  p-value: < 2.2e-16
```

Constrain values to 0-10 range

```
# Process data
seven_2 <- seven_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 = seven_2) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_2$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_2$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(seven_2$V1, seven_2$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_2$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_2$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_2$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_2$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_2$V1) + 0.5, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_2$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_2$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_2$V1), 2)}")) +
               labs(title ='B: Constrained (0-10 range)',
                    caption = 'Parameters: Mean = 7.2, SD = 1.7, Cor = 0.5') +
               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 = seven_2))
```

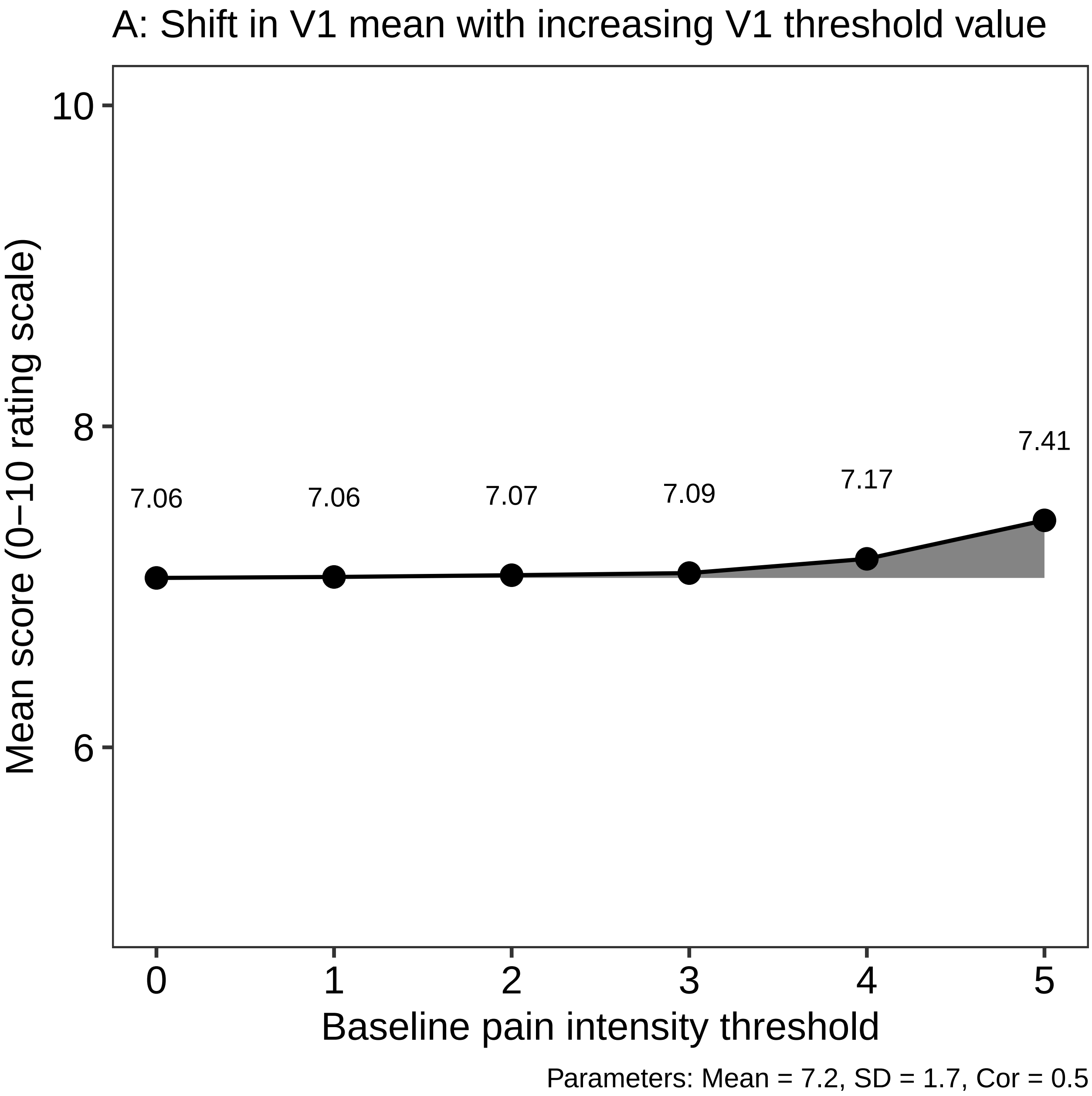
```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4870 -0.9326  0.0837  1.0138  4.0433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.69519    0.20117   18.37   <2e-16 ***
## V1           0.47574    0.02777   17.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.443 on 998 degrees of freedom
## Multiple R-squared:  0.2273, Adjusted R-squared:  0.2265
## F-statistic: 293.5 on 1 and 998 DF,  p-value: < 2.2e-16
```

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

```
# Extract visit 1 data
seven_2V1 <- seven_2$V1

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

# Generate a vector of V1 means at each V1 threshold
seven_2V1.shift <- sapply(cutoff, function(x){mean(seven_2V1[seven_2V1 > x])})

# Calculate deviation
(seven_2V1.df <- data.frame(cutoff = cutoff,
                           mean = seven_2V1.shift) %>%
        mutate(deviation = mean - mean(seven_2V1)))
```

```
##   cutoff     mean   deviation
## 1      0 7.055536 0.000000000
## 2      1 7.061597 0.006061597
## 3      2 7.072595 0.017059225
## 4      3 7.085784 0.030248378
## 5      4 7.174413 0.118877473
## 6      5 7.414388 0.358852371
```

```
# Plot data
ggplot(data = seven_2V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 1.7, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```

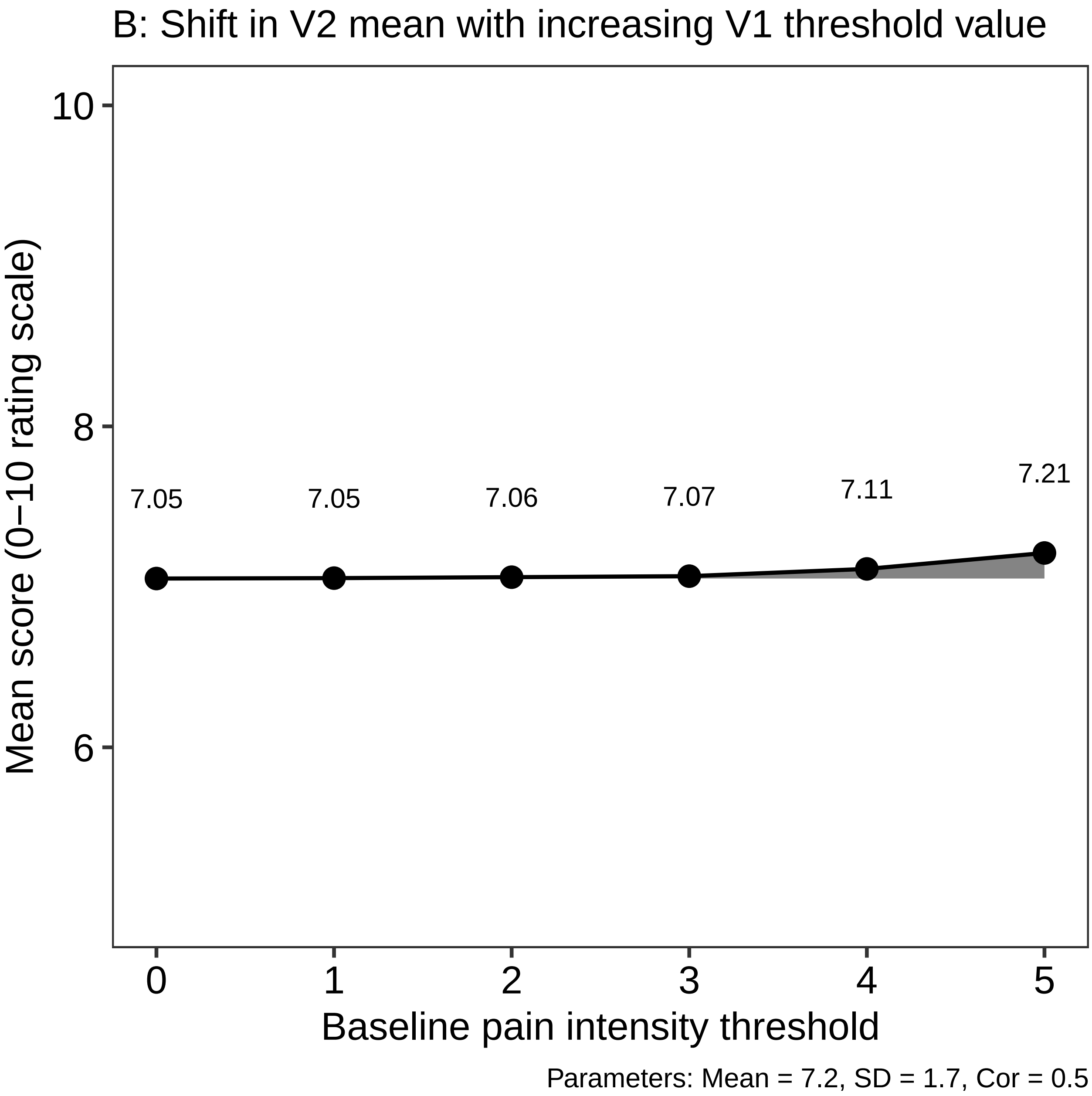


Model mean of V2 with increasing V1 thresholds from 0 to 5

```
# Extract visit 2 data
seven_2V2 <- seven_2$V2

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

# Generate a vector of V2 means at each V1 threshold
seven_2V2.shift <- map_dbl(.x = cutoff,
                           ~ seven_2 %>%
                               filter(V1 > .x) %>%
                               .$V2 %>%
                              mean(.))

# Calculate deviation
(seven_2V2.df <- data.frame(cutoff = cutoff,
                           mean = seven_2V2.shift) %>%
        mutate(deviation = mean - mean(seven_2V2)))
```

```
##   cutoff     mean   deviation
## 1      0 7.051805 0.000000000
## 2      1 7.054375 0.002570068
## 3      2 7.060238 0.008433189
## 4      3 7.066656 0.014850775
## 5      4 7.111739 0.059933681
## 6      5 7.210972 0.159166519
```

```
# Plot data
ggplot(data = seven_2V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 1.7, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



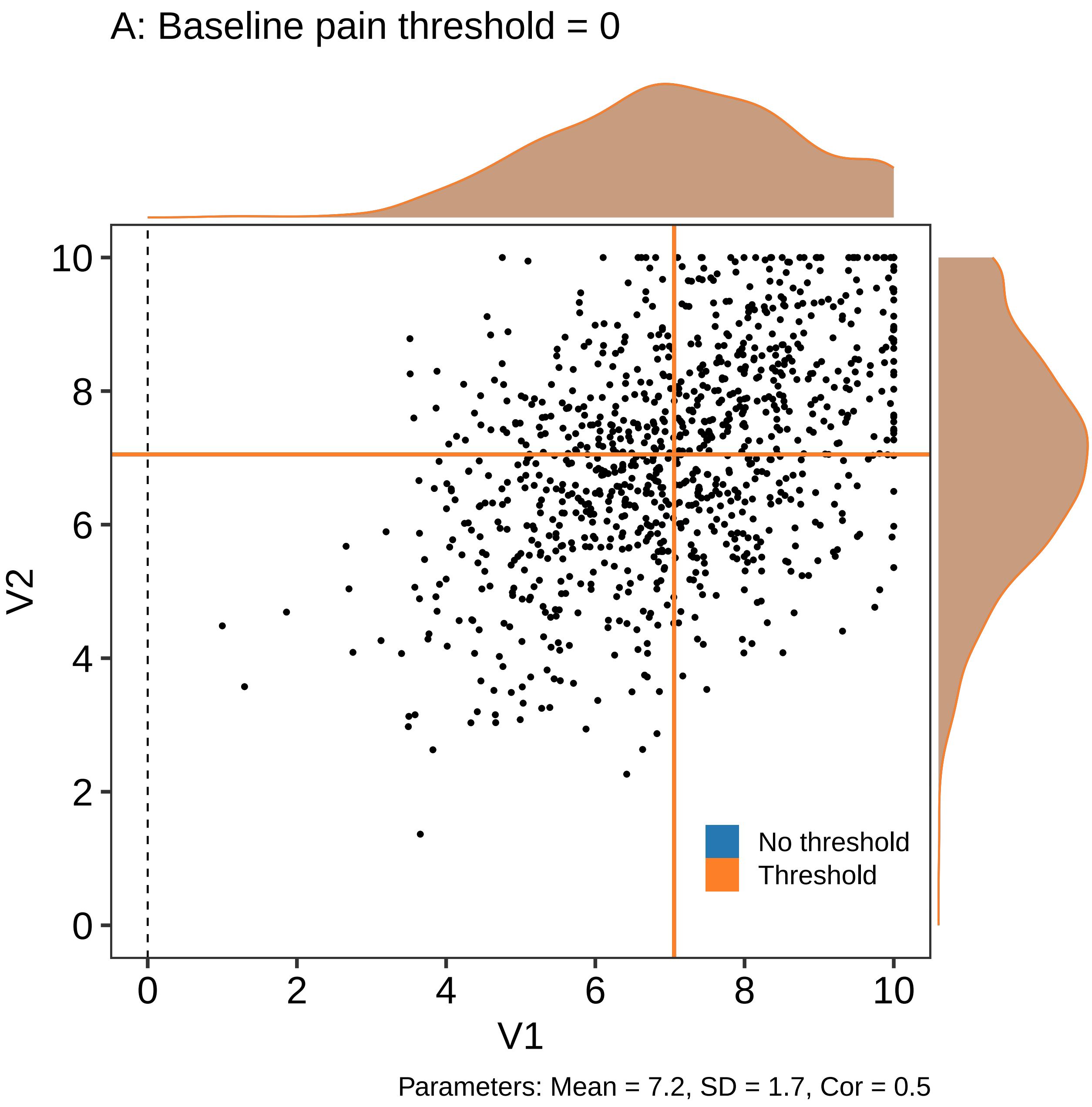
Placebo response

threshold: 0

```
# Process data
placebo_2.0 <- seven_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	0.00373	0.95	-0.104	0.108

```
# Plot the data
ggMarginal(placebo_2.0[, 1:3] %>%
               bind_rows(seven_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 = seven_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.7, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



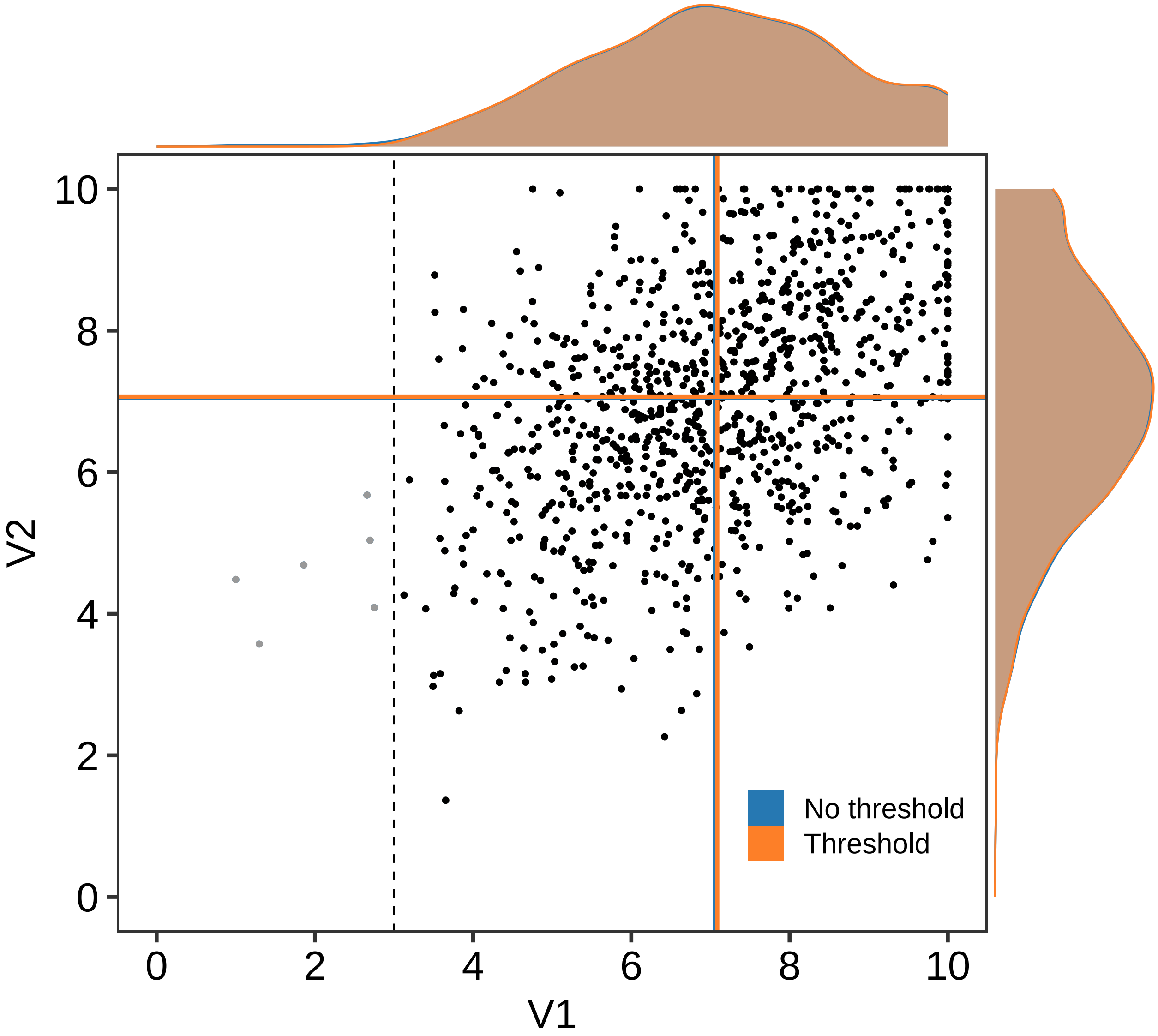
threshold: 3

```
# Process data
placebo_2.3 <- seven_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	994	0.0191	0.95	-0.0931	0.117

```
# Plot the data
ggMarginal(placebo_2.3[, 1:3] %>%
               bind_rows(seven_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 = seven_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.7, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

B: Baseline pain threshold = 3



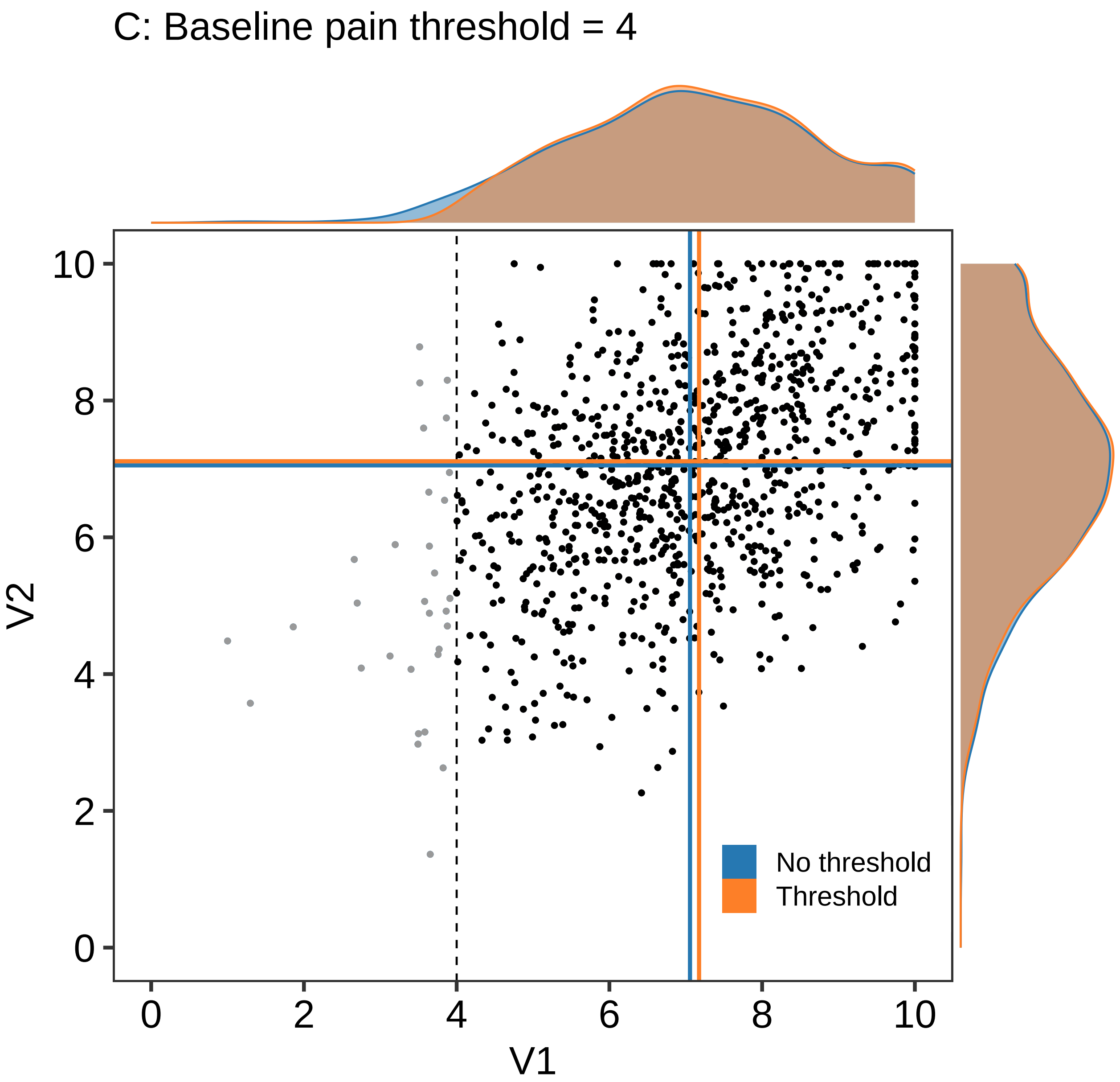
Parameters: Mean = 7.2, SD = 1.7, Cor = 0.5

threshold: 4

```
# Process that data
placebo_2.4 <- seven_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	969	0.0627	0.95	-0.0353	0.165

```
# Plot the data
ggMarginal(placebo_2.4[, 1:3] %>%
               bind_rows(seven_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 = seven_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.7, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```



C: Baseline pain threshold = 4

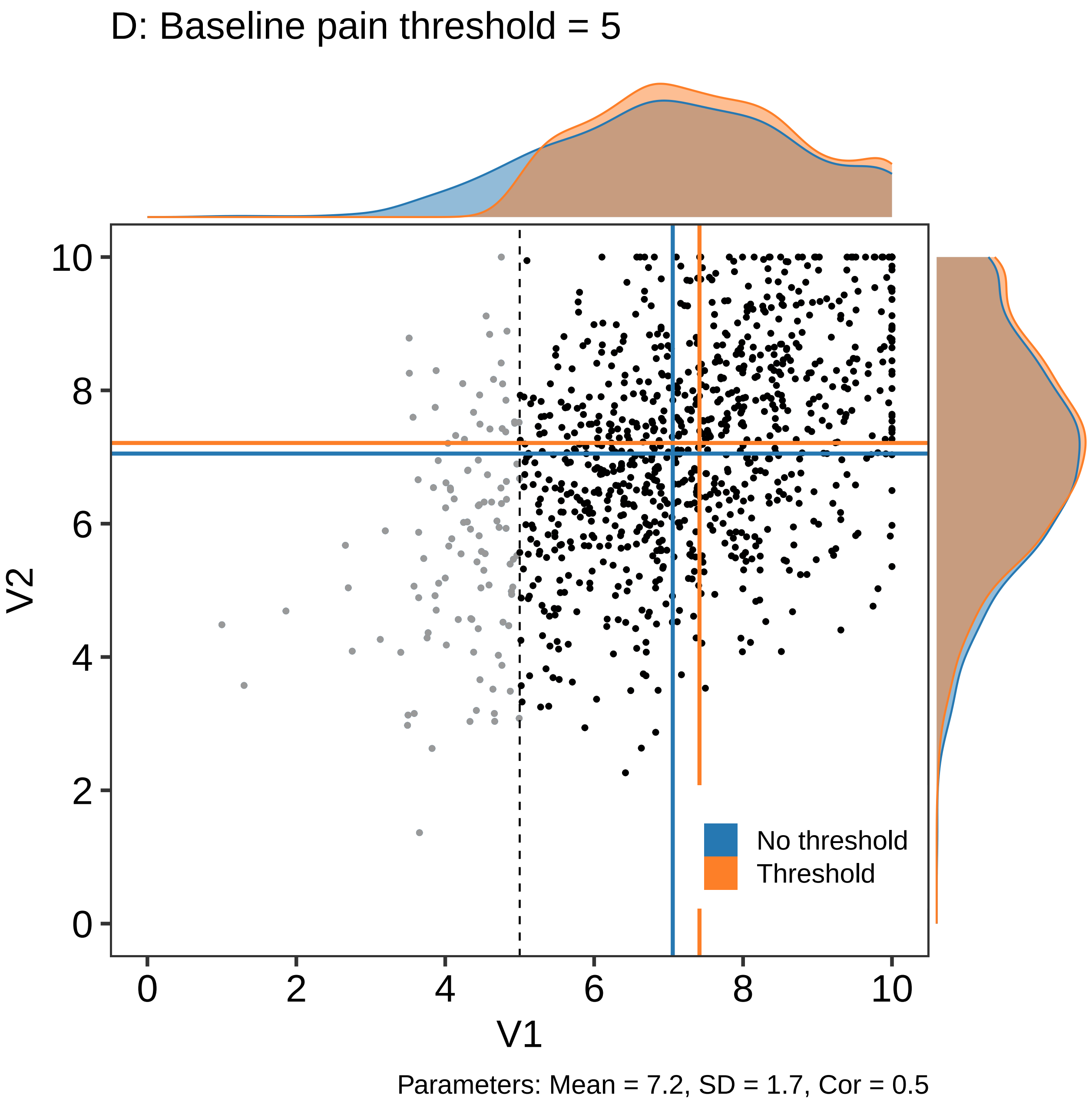
Parameters: Mean = 7.2, SD = 1.7, Cor = 0.5

threshold: 5

```
# Process that data
placebo_2.5 <- seven_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)
```

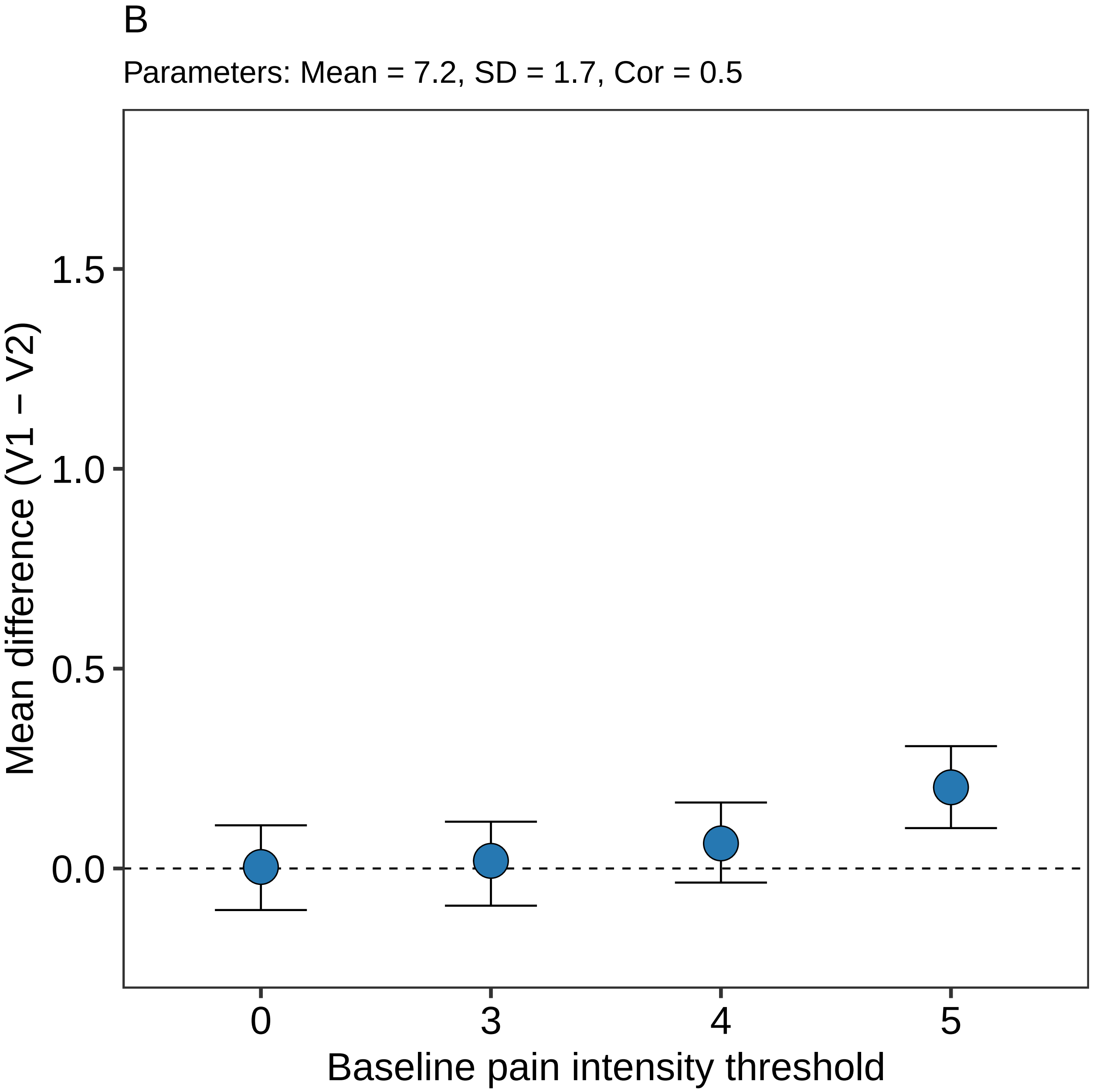
.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	888	0.203	0.95	0.101	0.306

```
# Plot the data
ggMarginal(placebo_2.5[, 1:3] %>%
               bind_rows(seven_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 = seven_2,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_2.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 1.7, Cor = 0.5') +
               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 = 7.2, SD = 1.7, Cor = 0.5',
       x = 'Baseline pain intensity threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_2
```



Mean = 7.2, SD = 2.2, Cor = 0.5

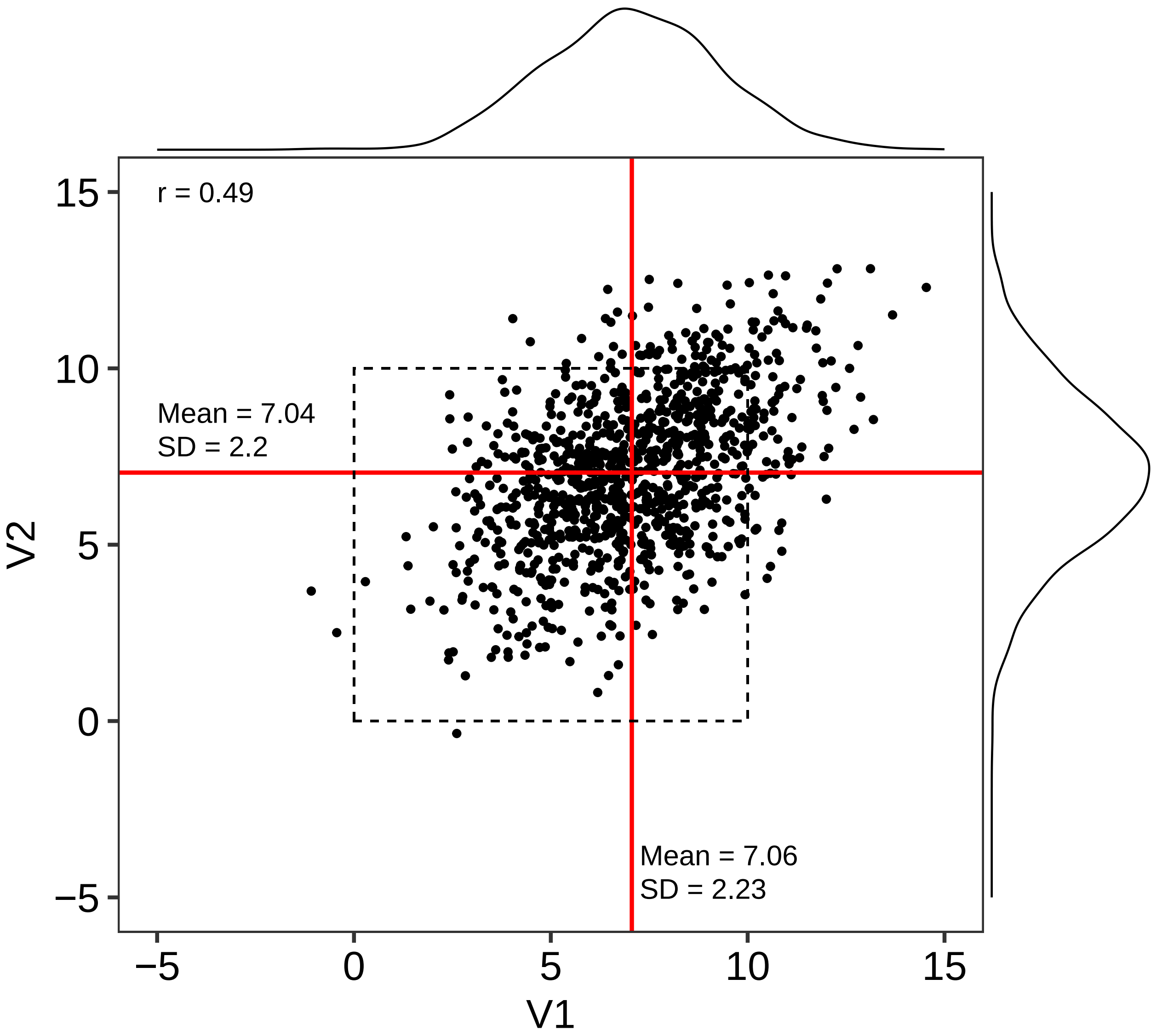
Generate and summarise data

Base data

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

# Generate the data
seven_3.base <- as.data.frame(mvrnorm(n = 1000, mu = c(7.2, 7.2), Sigma = cov_3))

# Plot base data
ggMarginal(ggplot(data = seven_3.base) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_3.base$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_3.base$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(seven_3.base$V1, seven_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_3.base$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_3.base$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_3.base$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_3.base$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_3.base$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_3.base$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_3.base$V1), 2)}")) +
               labs(title = 'A: Unconstrained',
                    caption = 'Parameters: Mean = 7.2, SD = 2.2, Cor = 0.5') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(plot.caption = element_text(size = 14)))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
```

A: Unconstrained



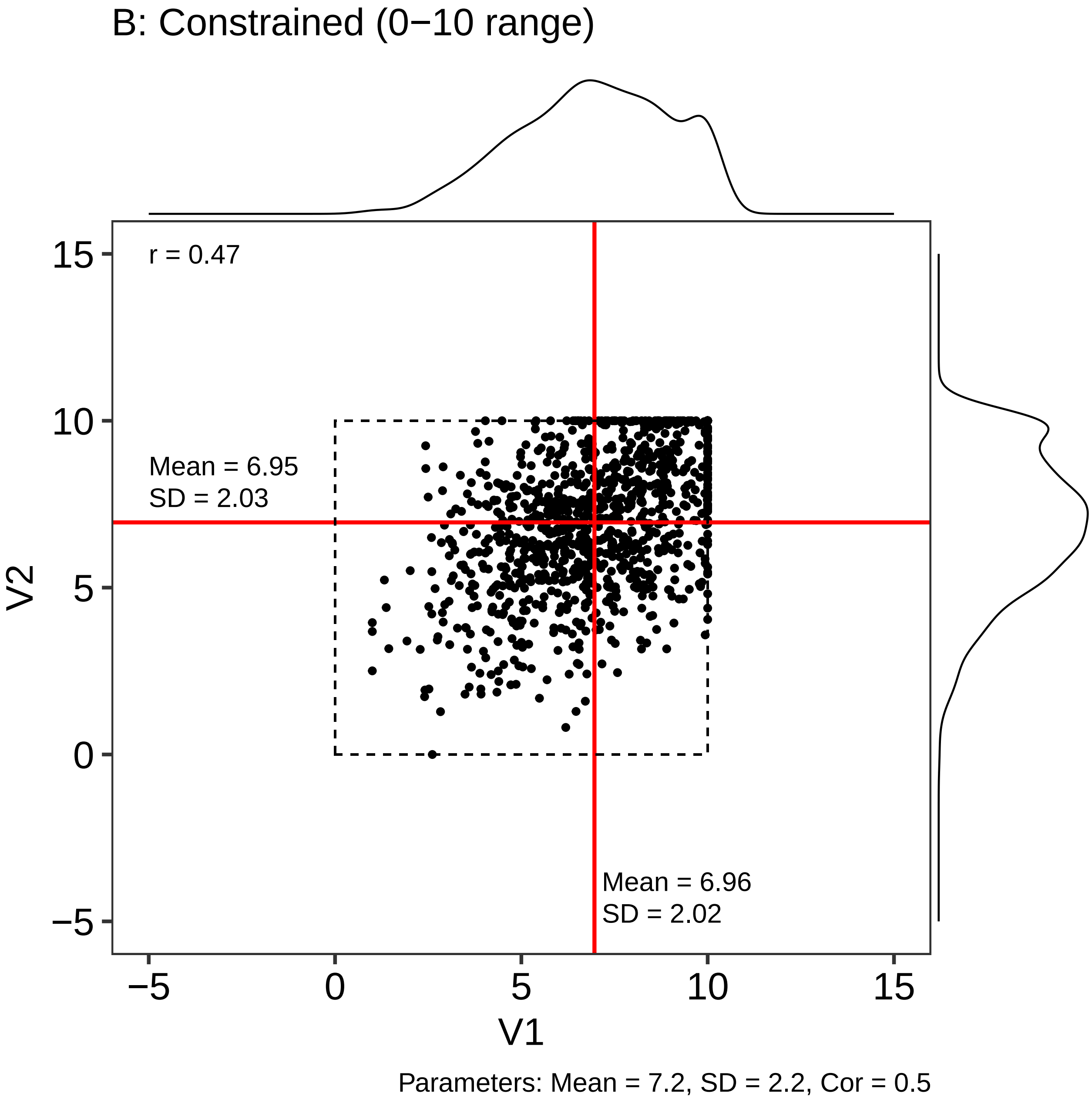
Parameters: Mean = 7.2, SD = 2.2, Cor = 0.5

```
# Linear regression
summary(lm(V2 ~ V1, data = seven_3.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_3.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8147 -1.2753  0.0793  1.2788  5.8269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.63194    0.20148   18.03   <2e-16 ***
## V1           0.48347    0.02723   17.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.916 on 998 degrees of freedom
## Multiple R-squared:  0.2401, Adjusted R-squared:  0.2394
## F-statistic: 315.4 on 1 and 998 DF,  p-value: < 2.2e-16
```

Constrain values to 0-10 range

```
# Process data
seven_3 <- seven_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 = seven_3) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(seven_3$V2),
                          colour = 'red', size = 1) +
               geom_vline(xintercept = mean(seven_3$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(seven_3$V1, seven_3$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_3$V2) + 1.7,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_3$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = mean(seven_3$V2) + 0.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_3$V2),2)}")) +
               annotate(geom = 'text', x = mean(seven_3$V1) + 0.2, y = -3.8,
                        hjust = 0, size = 5,
                        label = str_glue("Mean = {round(mean(seven_3$V1), 2)}")) +
               annotate(geom = 'text', x = mean(seven_3$V1) + 0.2, y = -4.75,
                        hjust = 0, size = 5,
                        label = str_glue("SD = {round(sd(seven_3$V1), 2)}")) +
               labs(title = 'B: Constrained (0-10 range)',
                    caption = 'Parameters: Mean = 7.2, SD = 2.2, Cor = 0.5') +
               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 = seven_3))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = seven_3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7815 -1.1542  0.1592  1.3512  4.4269
##
## Coefficients:
```

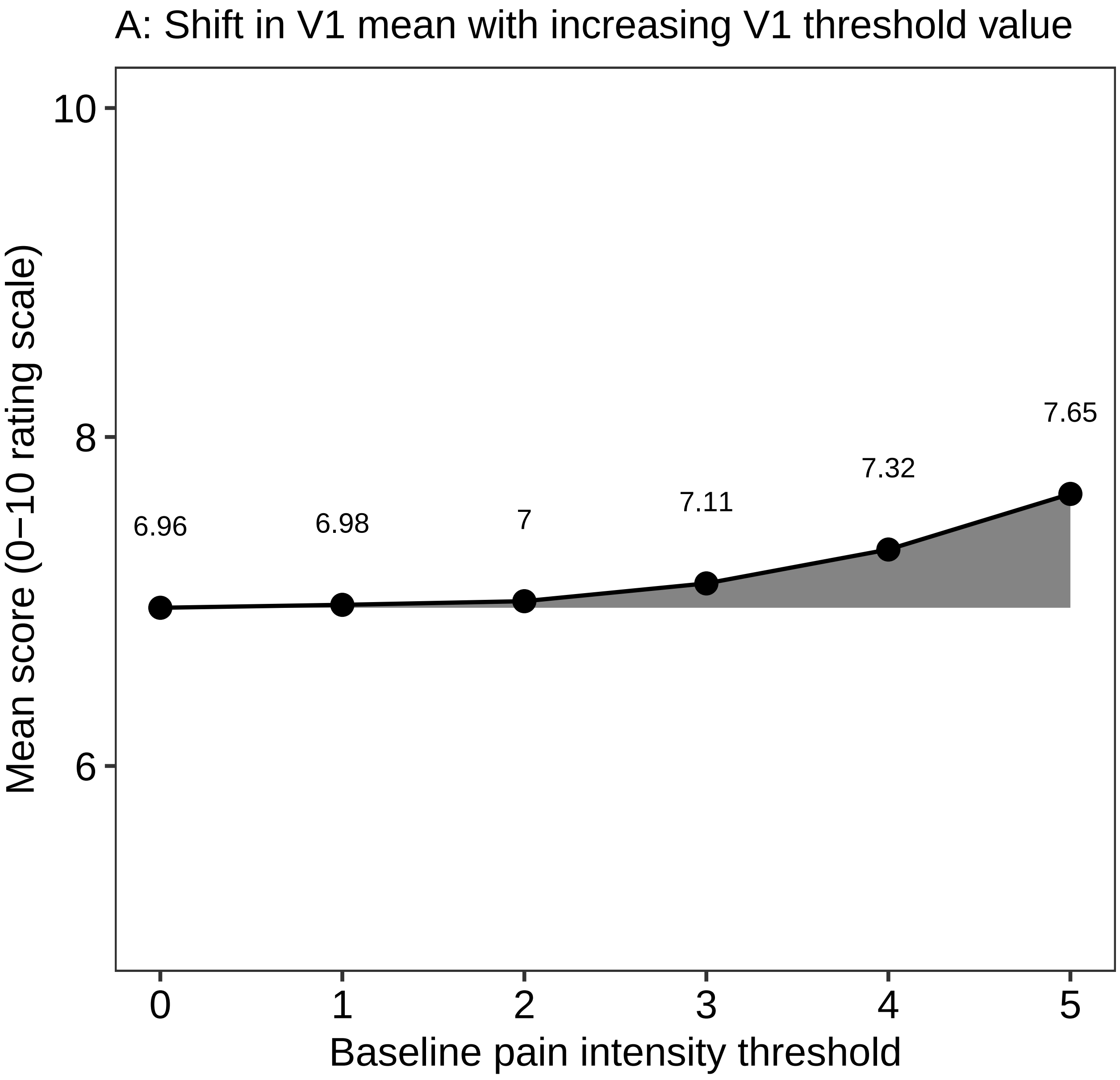
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6802     0.2037   18.07   <2e-16 ***
## V1            0.4703     0.0281   16.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.798 on 998 degrees of freedom
## Multiple R-squared:  0.2192, Adjusted R-squared:  0.2184
## F-statistic: 280.2 on 1 and 998 DF,  p-value: < 2.2e-16
```

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

```
# Extract visit 1 data
seven_3V1 <- seven_3$V1

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

# Generate a vector of V1 means at each V1 threshold
seven_3V1.shift <- sapply(cutoff, function(x){mean(seven_3V1[seven_3V1 > x])})

# Calculate deviation
(seven_3V1.df <- data.frame(cutoff = cutoff,
                           mean = seven_3V1.shift) %>%
        mutate(deviation = mean - mean(seven_3V1)))
```

```
##   cutoff     mean  deviation
## 1      0 6.961739 0.00000000
## 2      1 6.979678 0.01793903
## 3      2 7.001682 0.03994261
## 4      3 7.109841 0.14810171
## 5      4 7.315893 0.35415366
## 6      5 7.653894 0.69215480
```

```
# Plot data
ggplot(data = seven_3V1.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 2.2, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```


A: Shift in V1 mean with increasing V1 threshold value

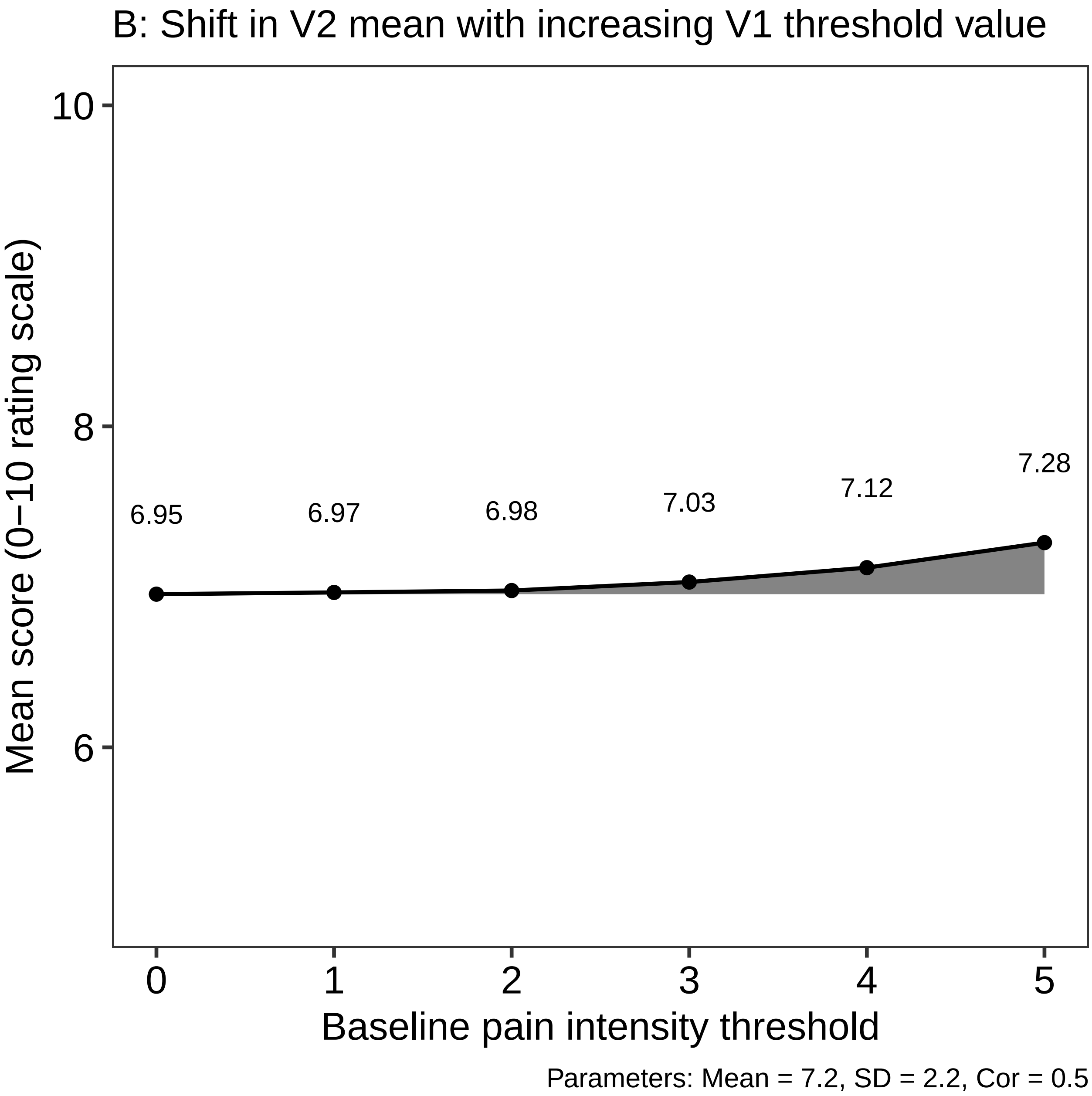
Parameters: Mean = 7.2, SD = 2.2, Cor = 0.5

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

```
# Extract visit 2 data
seven_3V2 <- seven_3$V2

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

# Generate a vector of V2 means at each V1 threshold
seven_3V2.shift <- map_dbl(.x = cutoff,
                          ~ seven_3 %>%
                              filter(V1 > .x) %>%
                              .$V2 %>%
                              mean(.))

# Calculate deviation
(seven_3V2.df <- data.frame(cutoff = cutoff,
                           mean = seven_3V2.shift) %>%
        mutate(deviation = mean - mean(seven_3V2)))
```

```
##   cutoff     mean  deviation
## 1      0 6.954406 0.00000000
## 2      1 6.965158 0.01075162
## 3      2 6.976898 0.02249223
## 4      3 7.029882 0.07547615
## 5      4 7.119502 0.16509600
## 6      5 7.275633 0.32122716
```

```
# Plot data
ggplot(data = seven_3V2.df) +
    aes(x = cutoff, y = mean, ymin = mean(seven_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 = 7.2, SD = 2.2, Cor = 0.5',
         x = 'Baseline pain intensity threshold',
         y = 'Mean score (0-10 rating scale)') +
    theme(plot.caption = element_text(size = 14))
```



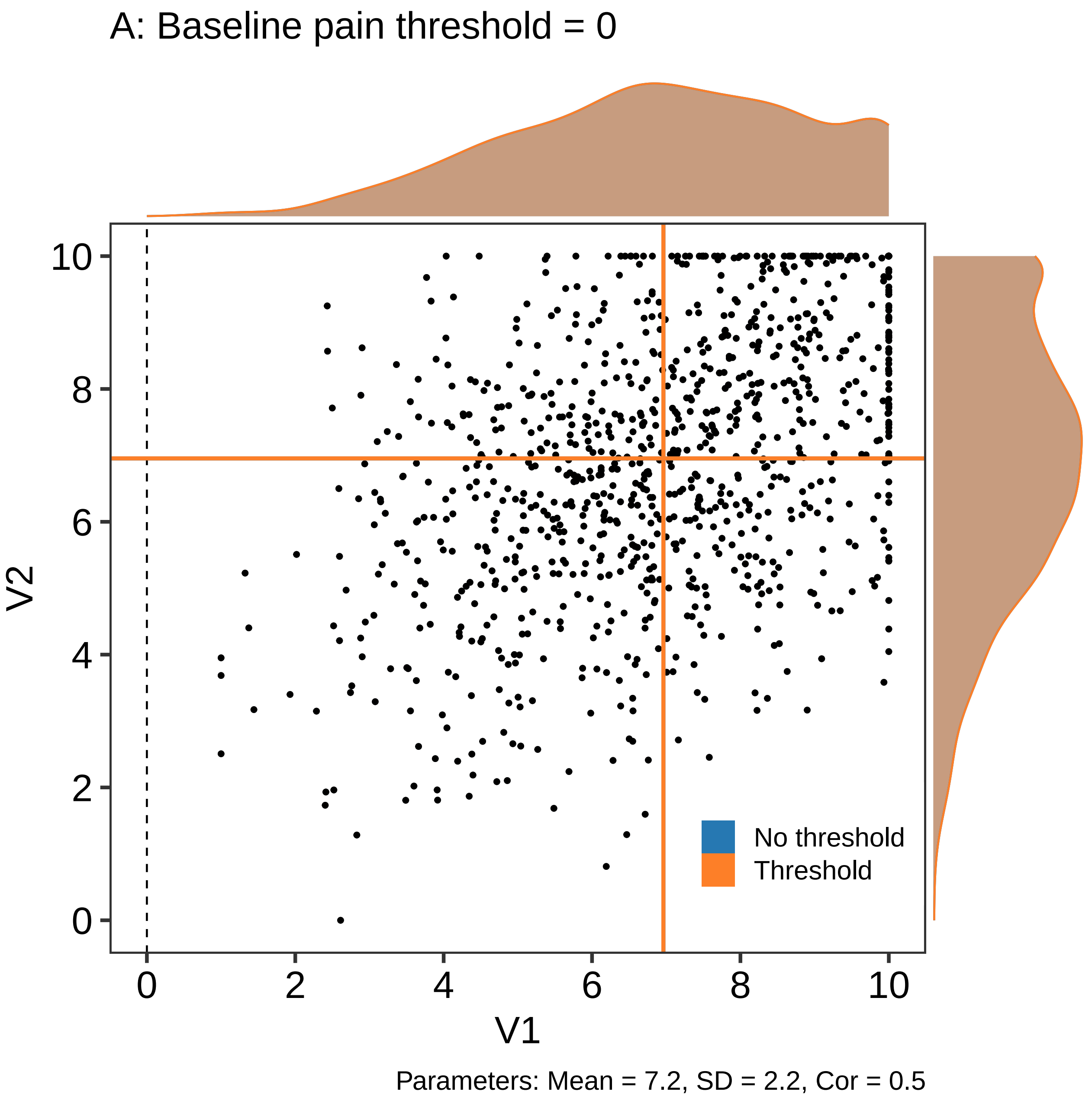
Placebo response

threshold: 0

```
# Process data
placebo_3.0 <- seven_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	0.00733	0.95	-0.124	0.134

```
# Plot the data
ggMarginal(placebo_3.0[, 1:3] %>%
               bind_rows(seven_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 = seven_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.0,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 2.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

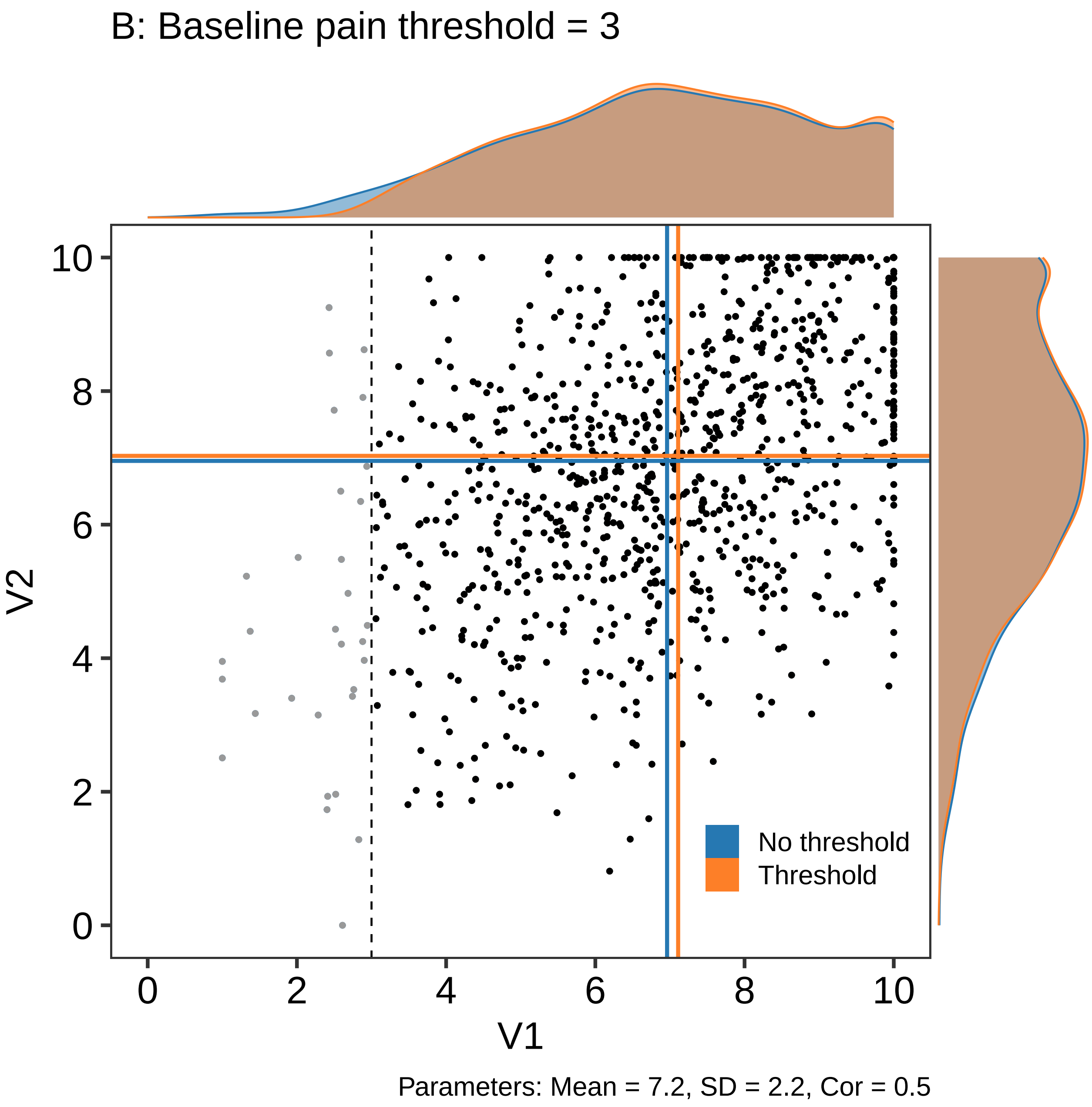


threshold: 3

```
# Process data
placebo_3.3 <- seven_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	969	0.08	0.95	-0.0429	0.21

```
# Plot the data
ggMarginal(placebo_3.3[, 1:3] %>%
               bind_rows(seven_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 = seven_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.3,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 2.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

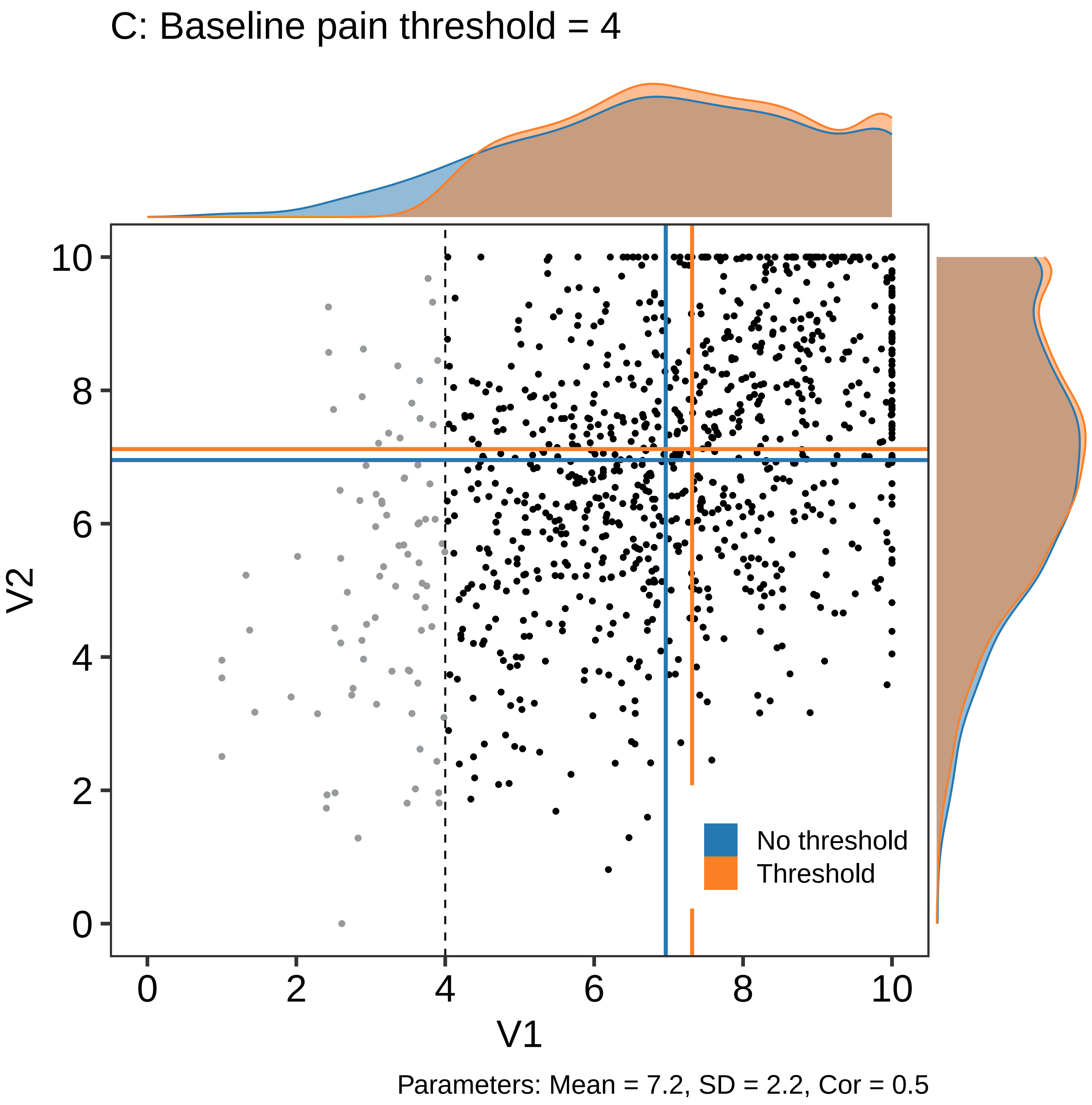


threshold: 4

```
# Process that data
placebo_3.4 <- seven_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	916	0.196	0.95	0.0625	0.329

```
# Plot the data
ggMarginal(placebo_3.4[, 1:3] %>%
               bind_rows(seven_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 = seven_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.4,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 2.2, Cor = 0.5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     plot.caption = element_text(size = 14)),
           groupColour = TRUE,
           groupFill = TRUE)
```

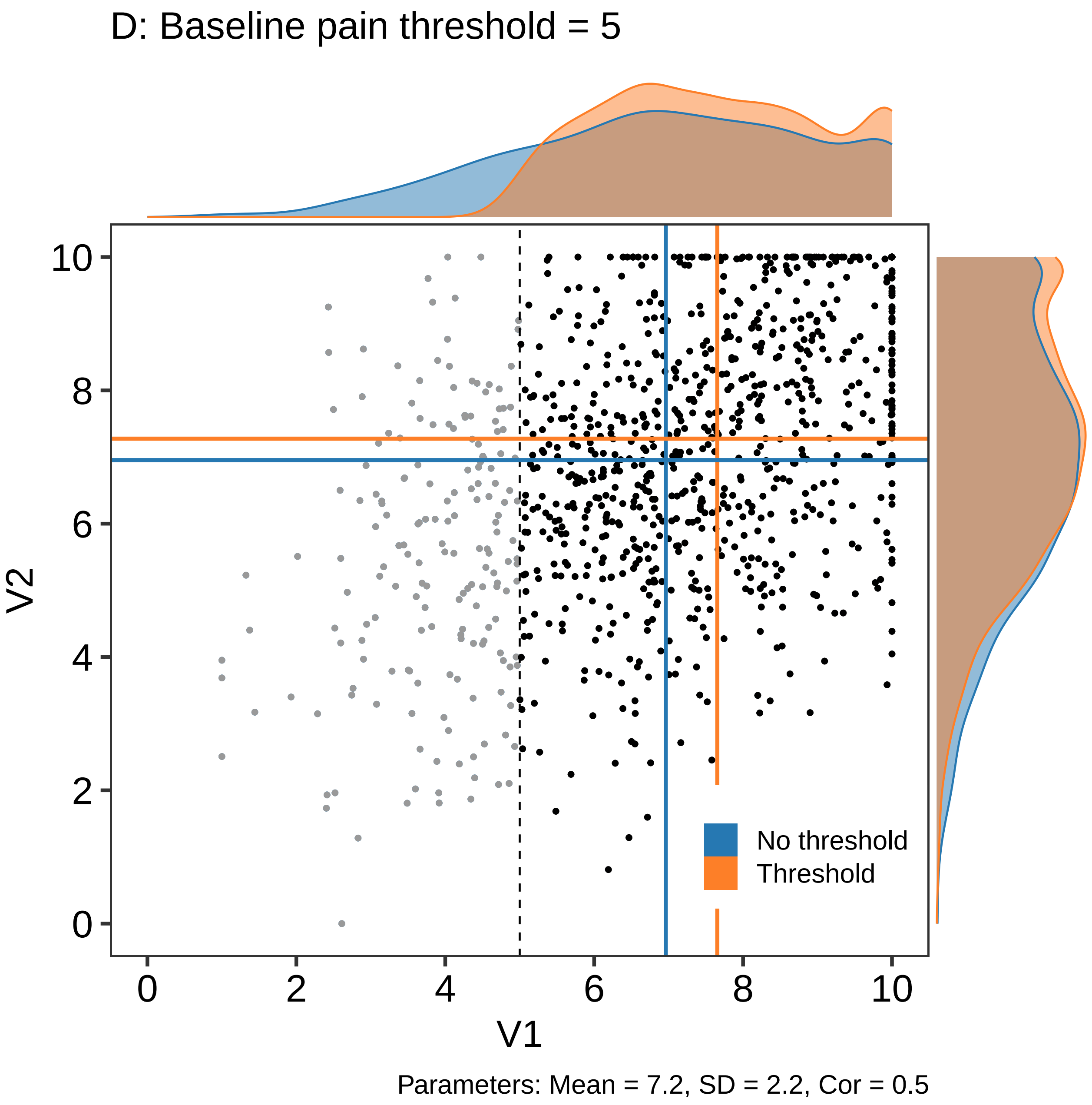


threshold: 5

```
# Process that data
placebo_3.5 <- seven_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)
```

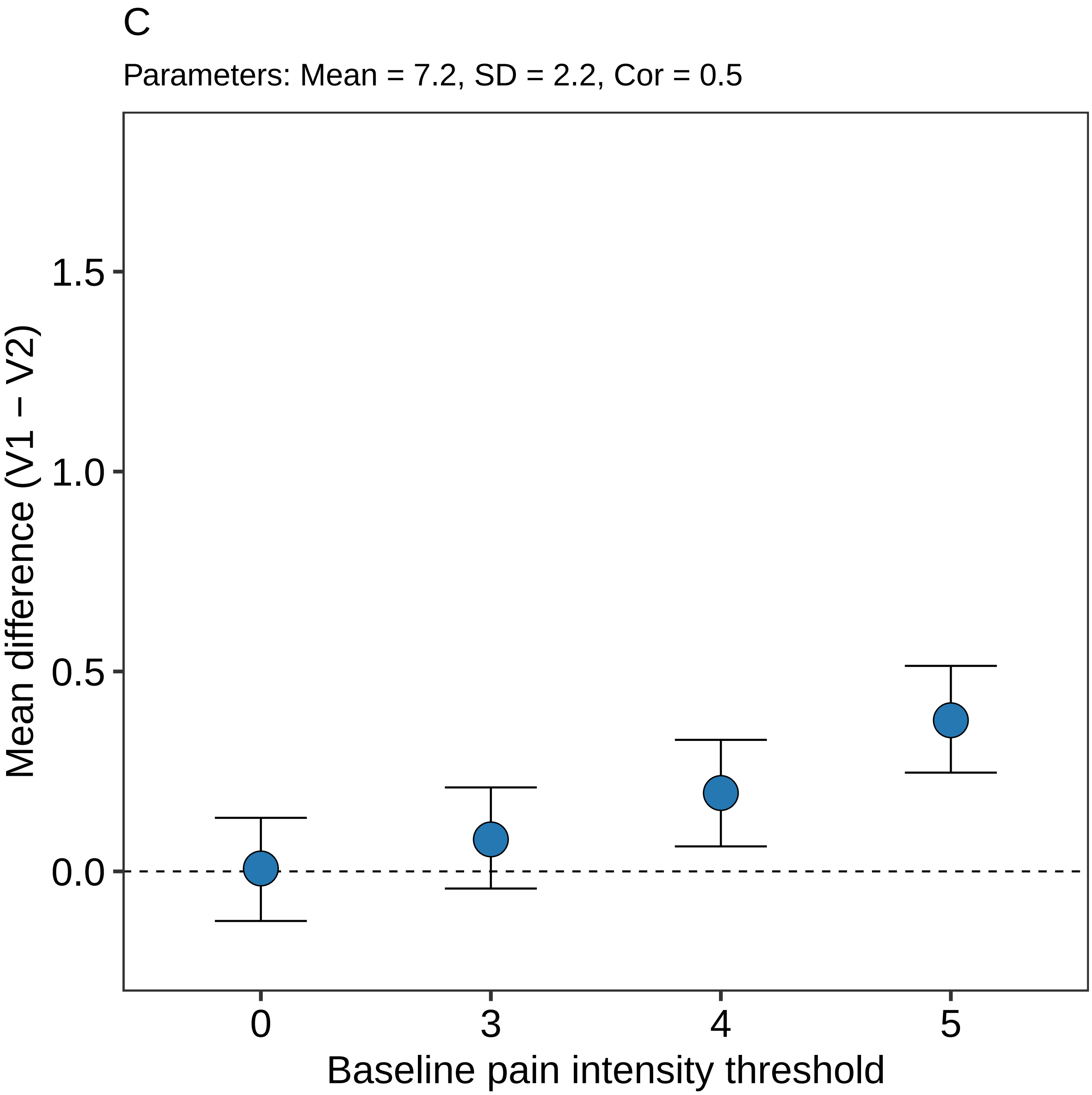
.id	n	Mean	Conf.level	Bca.lower	Bca.upper
5	817	0.378	0.95	0.247	0.514

```
# Plot the data
ggMarginal(placebo_3.5[, 1:3] %>%
               bind_rows(seven_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 = seven_3,
                          colour = '#999999',
                          size = 1) +
               geom_point(data = placebo_3.5,
                          size = 1,
                          colour = '#000000') +
               geom_vline(xintercept = mean(seven_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(seven_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 = 7.2, SD = 2.2, Cor = 0.5') +
               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 = 7.2, SD = 2.2, Cor = 0.5',
       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/7.2_0.5.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
```