

Regression to the mean modeling

Mean pain rating of 5.2 at 0.2 correlation

Peter Kamerman

Last knitted: 10 October 2019

Contents

Generate 2x2 covariance matrix	2
Mean = 5.2, SD = 1.2, r = 0.2	2
Generate and summarise data	2
Base data	2
Constrain values to 0-10 range	4
Model mean of V1 with increasing V1 thresholds from 0 to 5	6
Model mean of V2 with increasing V1 thresholds from 0 to 5	8
Placebo response	10
threshold: 0	10
threshold: 3	12
threshold: 4	14
threshold: 5	16
Publication plots	18
Mean = 5.2, SD = 1.7, r = 0.2	21
Generate and summarise data	21
Base data	21
Constrain values to 0-10 range	23
Model mean of V1 with increasing V1 thresholds from 0 to 5	25
Model mean of V2 with increasing V1 thresholds from 0 to 5	27
Placebo response	29
threshold: 0	29
threshold: 3	31
threshold: 4	33
threshold: 5	35
Publication plots	37
Mean = 5.2, SD = 2.2, r = 0.2	40
Generate and summarise data	40
Base data	40
Constrain values to 0-10 range	42
Model mean of V1 with increasing V1 thresholds from 0 to 5	44
Model mean of V2 with increasing V1 thresholds from 0 to 5	46
Placebo response	48
threshold: 0	48
threshold: 3	50
threshold: 4	52
threshold: 5	54
Publication plots	56
Publication composite plots	59
Session information	60

Generate 2x2 covariance matrix

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

```
# Correlation matrices
cor <- matrix(c(1, 0.2, 0.2, 1), ncol = 2)
```

```
# SDs
std_1 <- c(1.2, 1.2)
std_2 <- c(1.7, 1.7)
std_3 <- c(2.2, 2.2)
```

```
# Covariance matrices
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, r = 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))
```

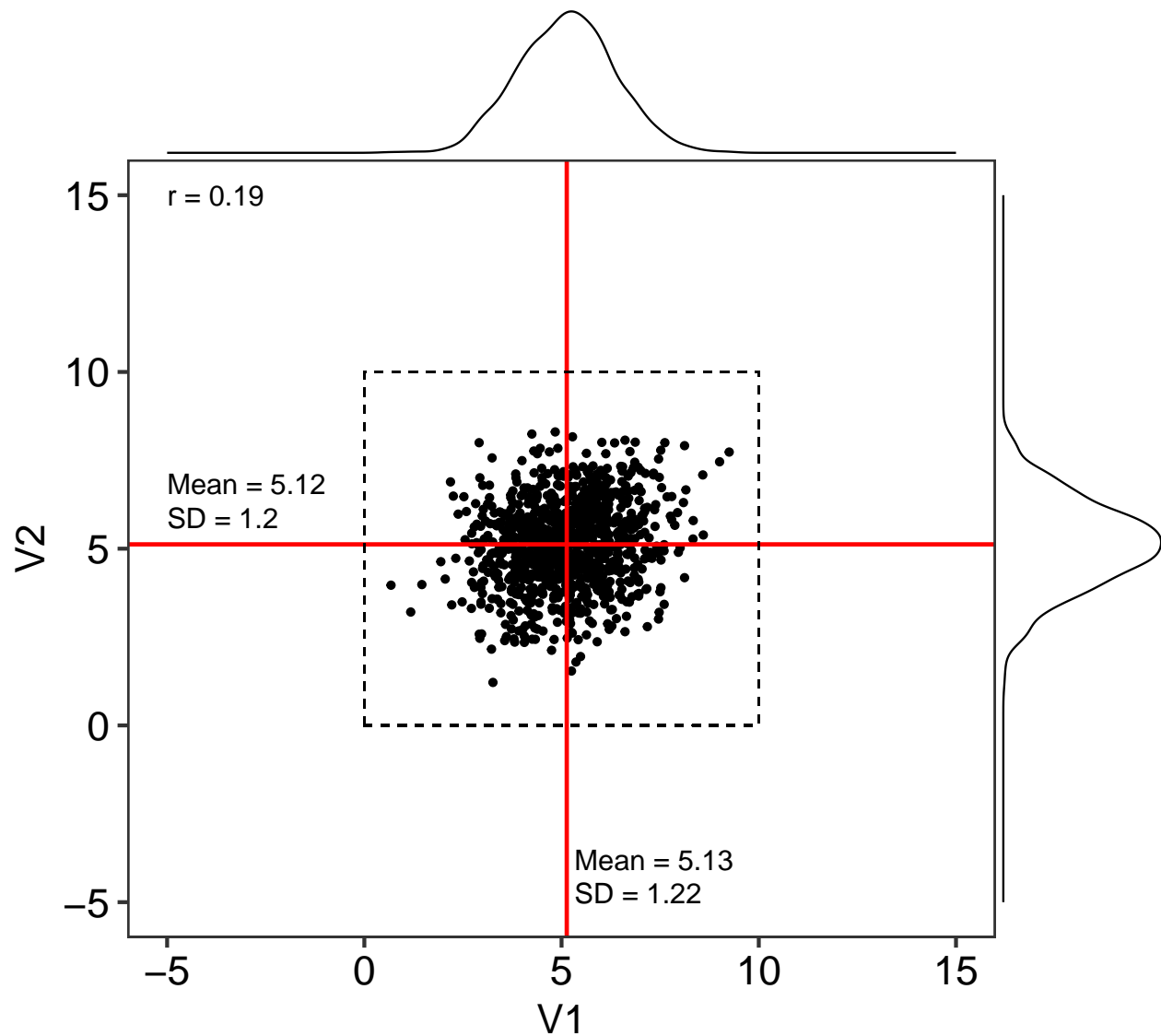
```
# Plot base data
ggMarginal(ggplot(data = five_1.base) +
            aes(x = V1, y = V2) +
```

```

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, r = 0.2') +
scale_y_continuous(limits = c(-5, 15)) +
scale_x_continuous(limits = c(-5, 15)) +
theme(plot.caption = element_text(size = 14))

```

A: Unconstrained



Parameters: Mean = 5.2, SD = 1.2, $r = 0.2$

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)) %>%
```

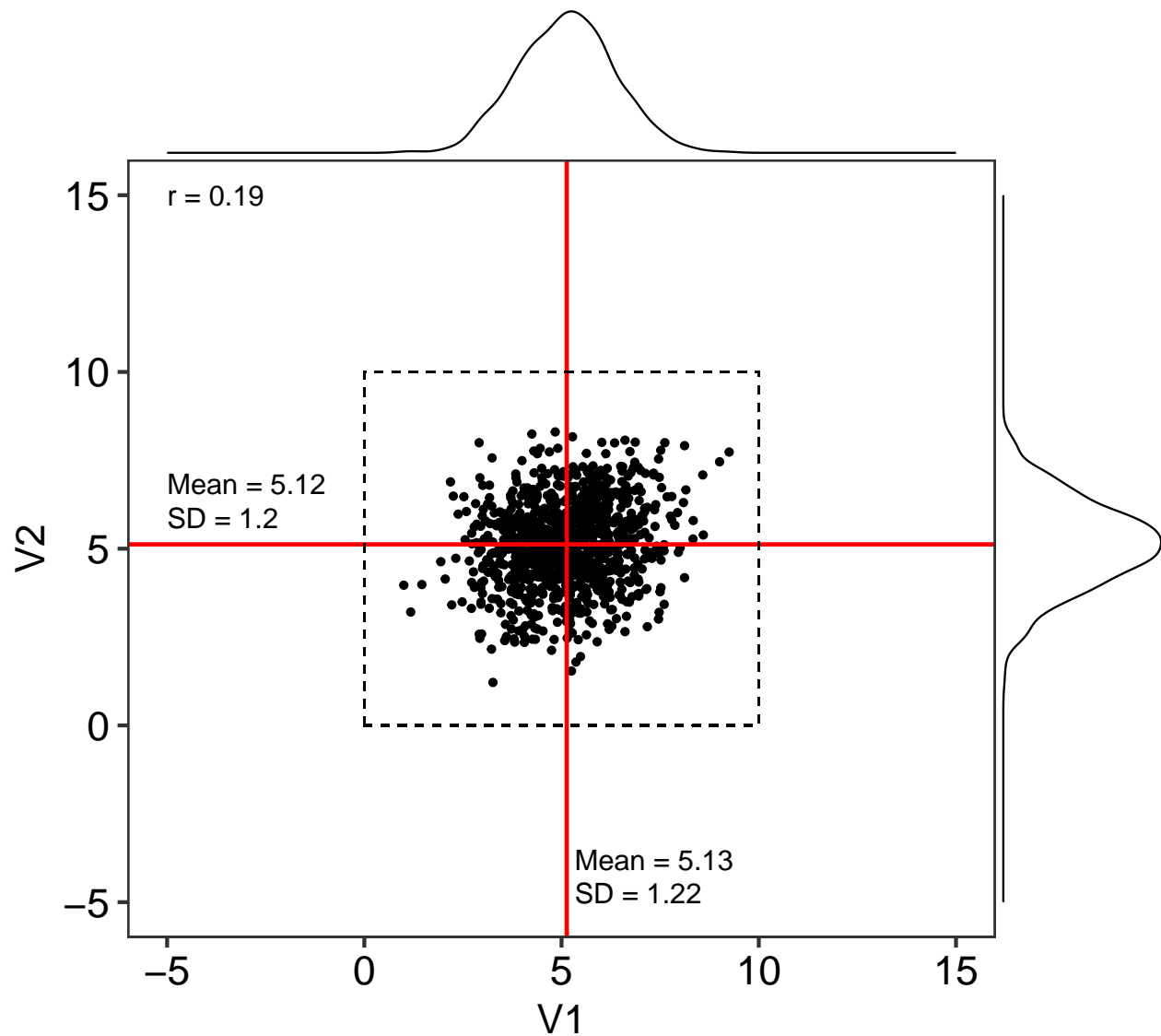
```

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, r = 0.2') +
  scale_y_continuous(limits = c(-5, 15)) +
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))

```

B: Constrained (0–10 range)



Parameters: Mean = 5.2, SD = 1.2, $r = 0.2$

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])})

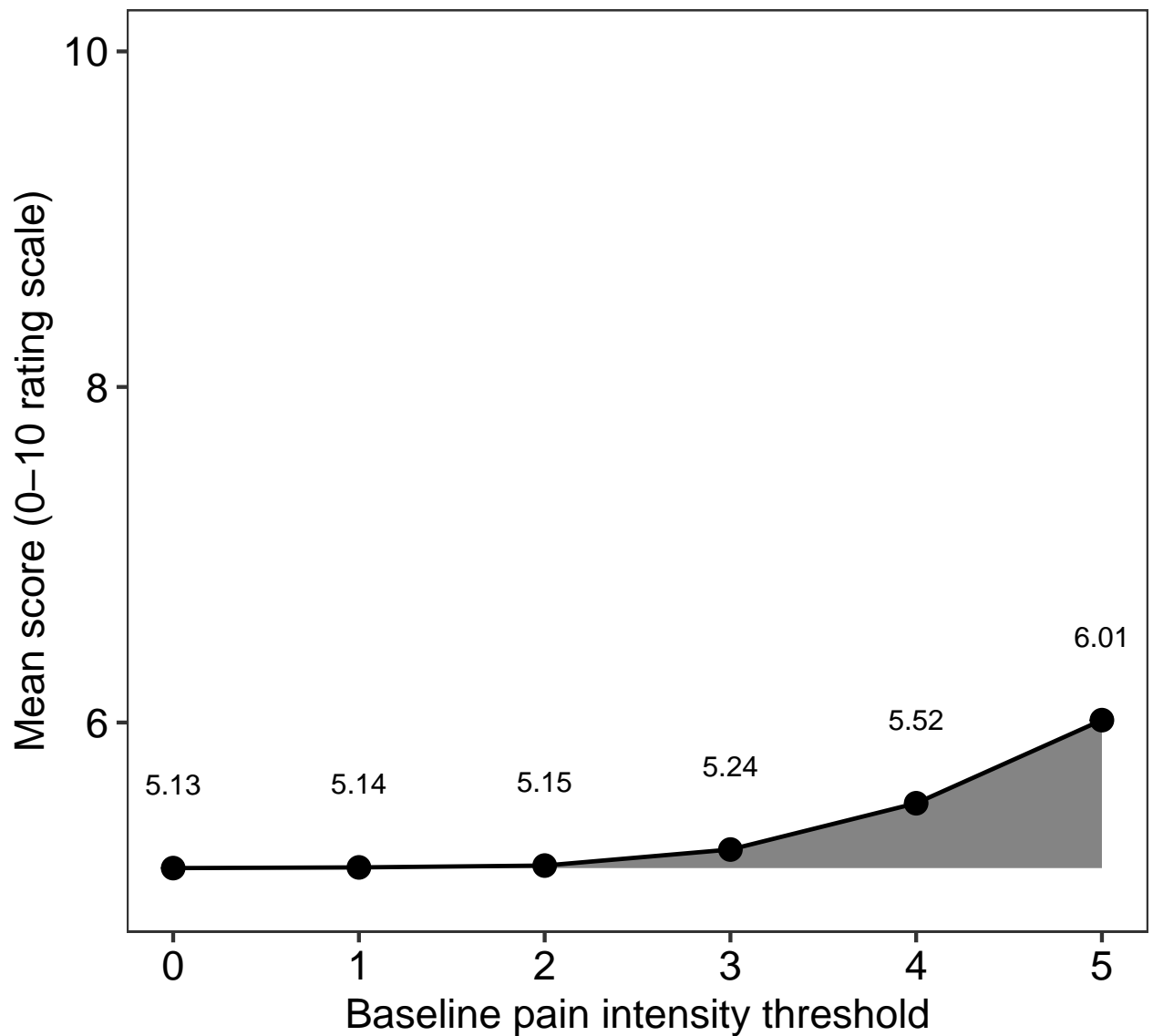
# Calculate deviation
```

```
(five_1V1.df <- data.frame(time = 'V1',
                           cutoff = cutoff,
                           cutoff2 = cutoff - 0.15,
                           mean = five_1V1.shift) %>%
  mutate(deviation = mean - mean(five_1V1),
         time = as.character(time)))
```

```
##   time cutoff cutoff2    mean  deviation
## 1  V1      0   -0.15 5.132359 0.000000000
## 2  V1      1    0.85 5.136496 0.004136496
## 3  V1      2    1.85 5.147375 0.015015587
## 4  V1      3    2.85 5.242943 0.110584082
## 5  V1      4    3.85 5.518841 0.386482235
## 6  V1      5    4.85 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, r = 0.2',
       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 = 5.2, SD = 1.2, $r = 0.2$

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(.))

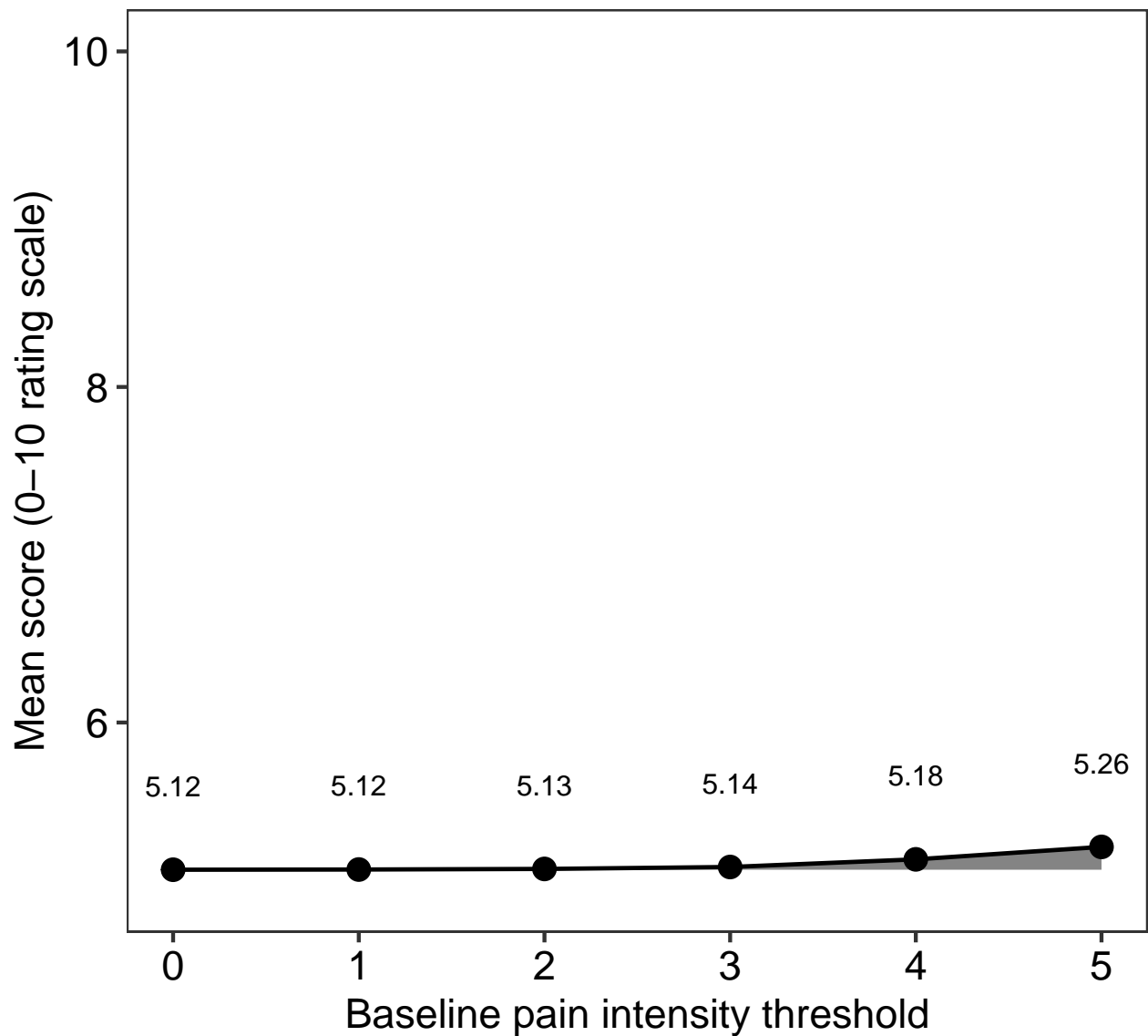
(five_1V2.df <- data.frame(time = 'V2',
                          cutoff = cutoff,
                          cutoff2 = cutoff + 0.15,
                          mean = five_1V2.shift) %>%
  mutate(deviation = mean - mean(five_1V1),
         time = as.character(time)))

##   time cutoff cutoff2    mean  deviation
## 1   V2      0    0.15 5.122621 -0.009737793
## 2   V2      1    1.15 5.123778 -0.008581446
## 3   V2      2    2.15 5.127338 -0.005020987
## 4   V2      3    3.15 5.138913  0.006553417
## 5   V2      4    4.15 5.184385  0.052025336
## 6   V2      5    5.15 5.258636  0.126276286

# 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, r = 0.2',
       x = 'Baseline pain intensity threshold',
       y = 'Mean score (0-10 rating scale)') +
  theme(plot.caption = element_text(size = 14))

```

B: Shift in V2 mean with increasing V1 threshold value



Parameters: Mean = 5.2, SD = 1.2, $r = 0.2$

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

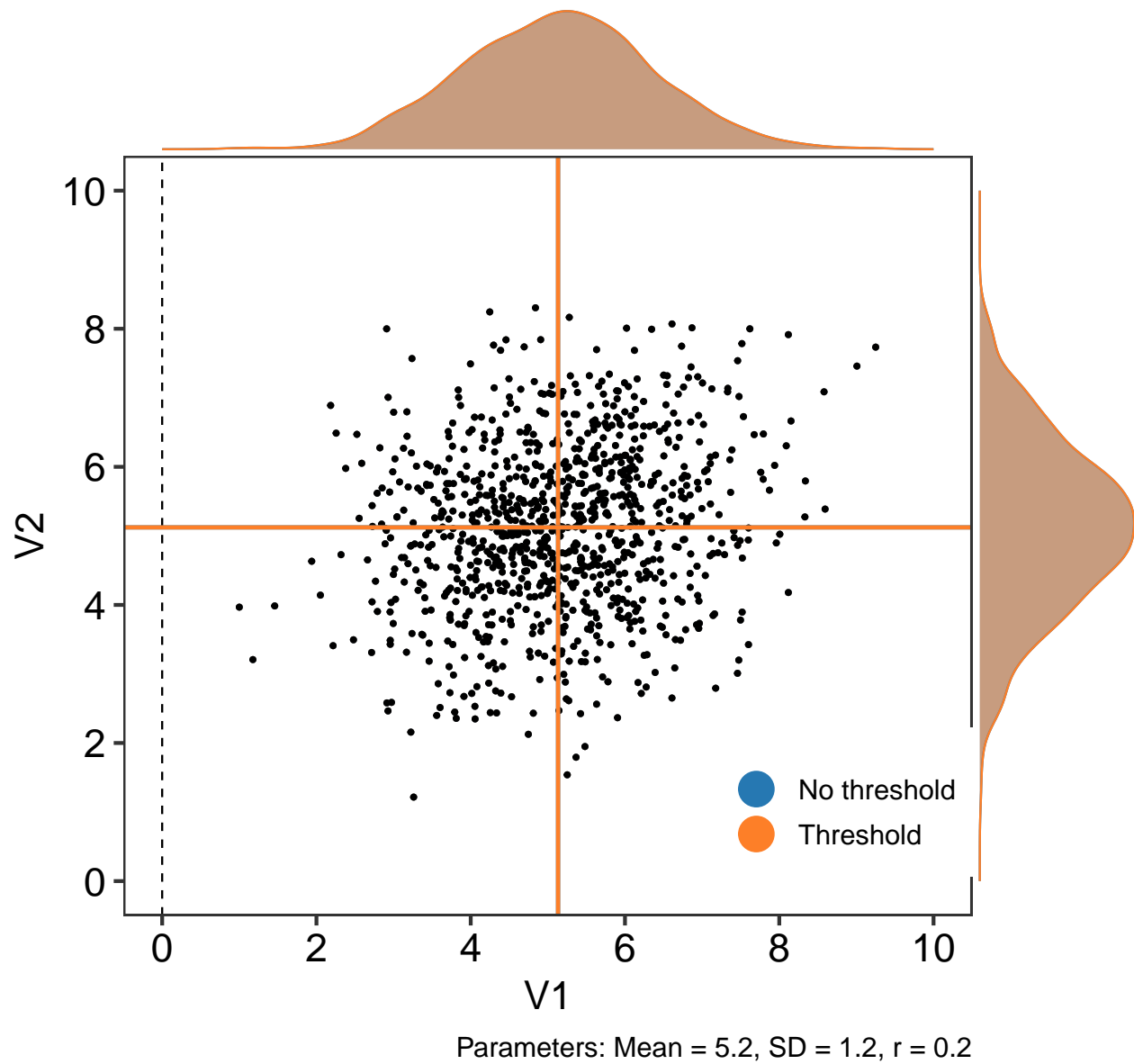
```

# 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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
groupColour = TRUE,
groupFill = TRUE)

```

A: Baseline pain threshold = 0



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

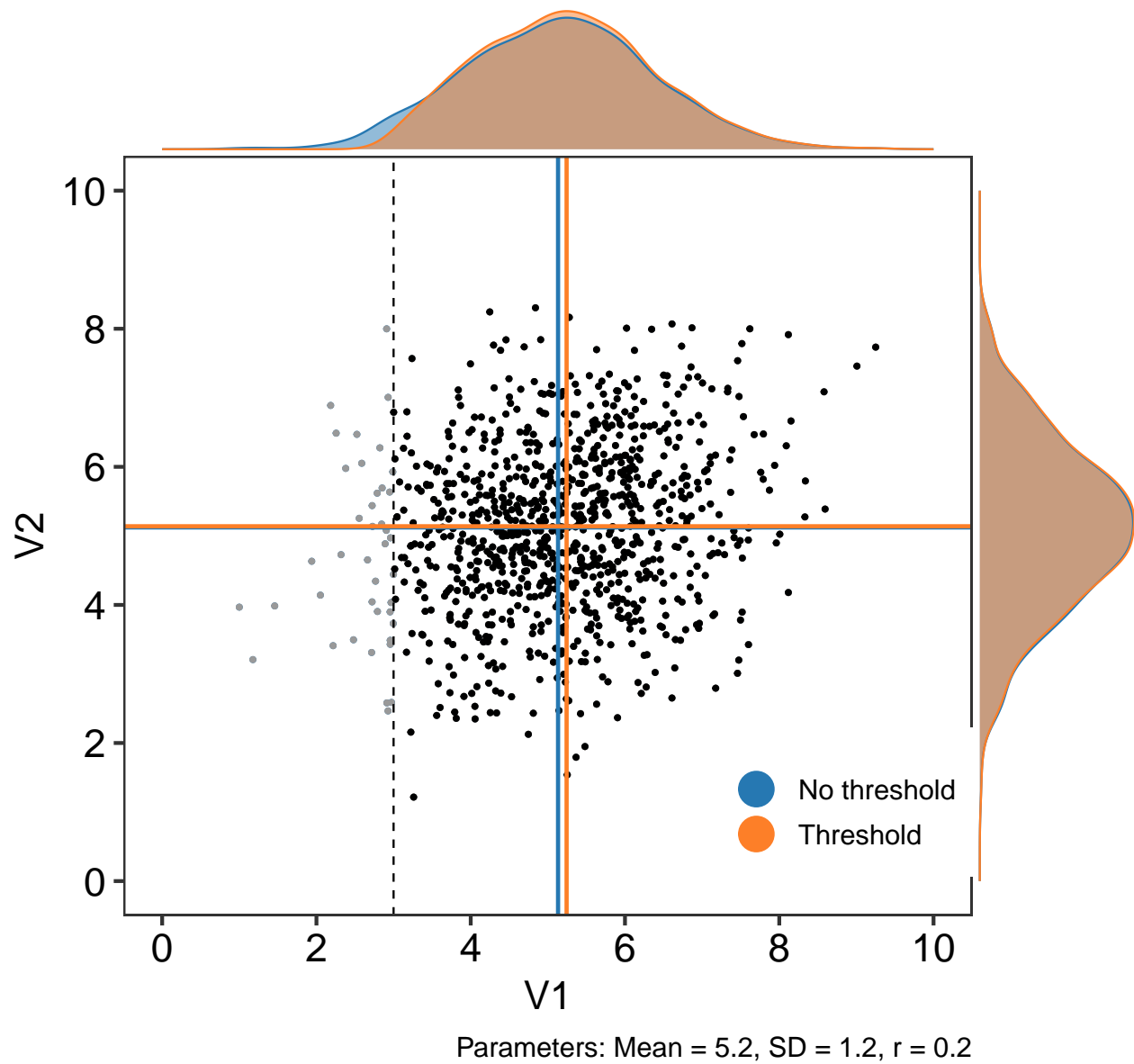
```

# 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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

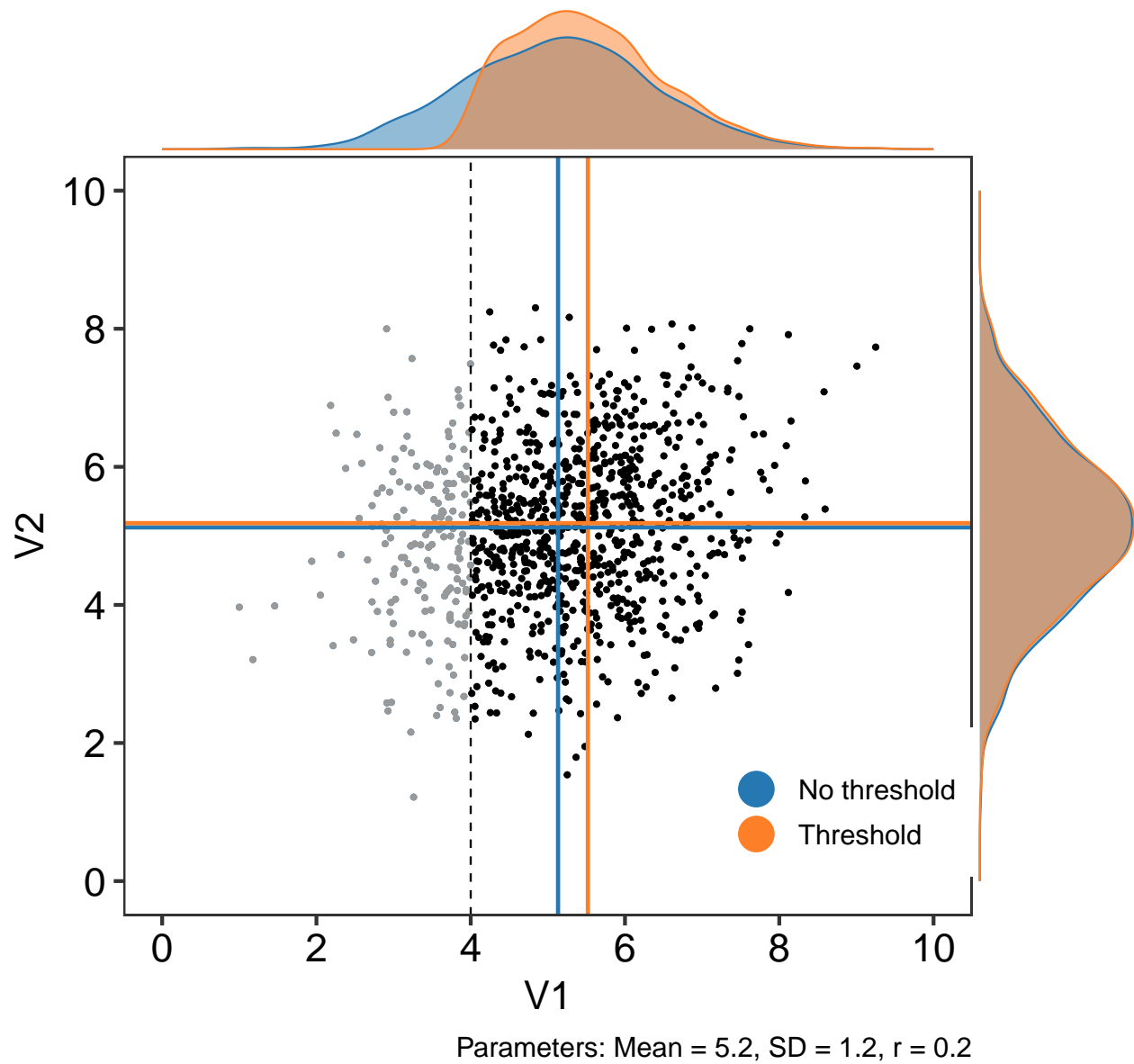
kable(diff_1.4)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

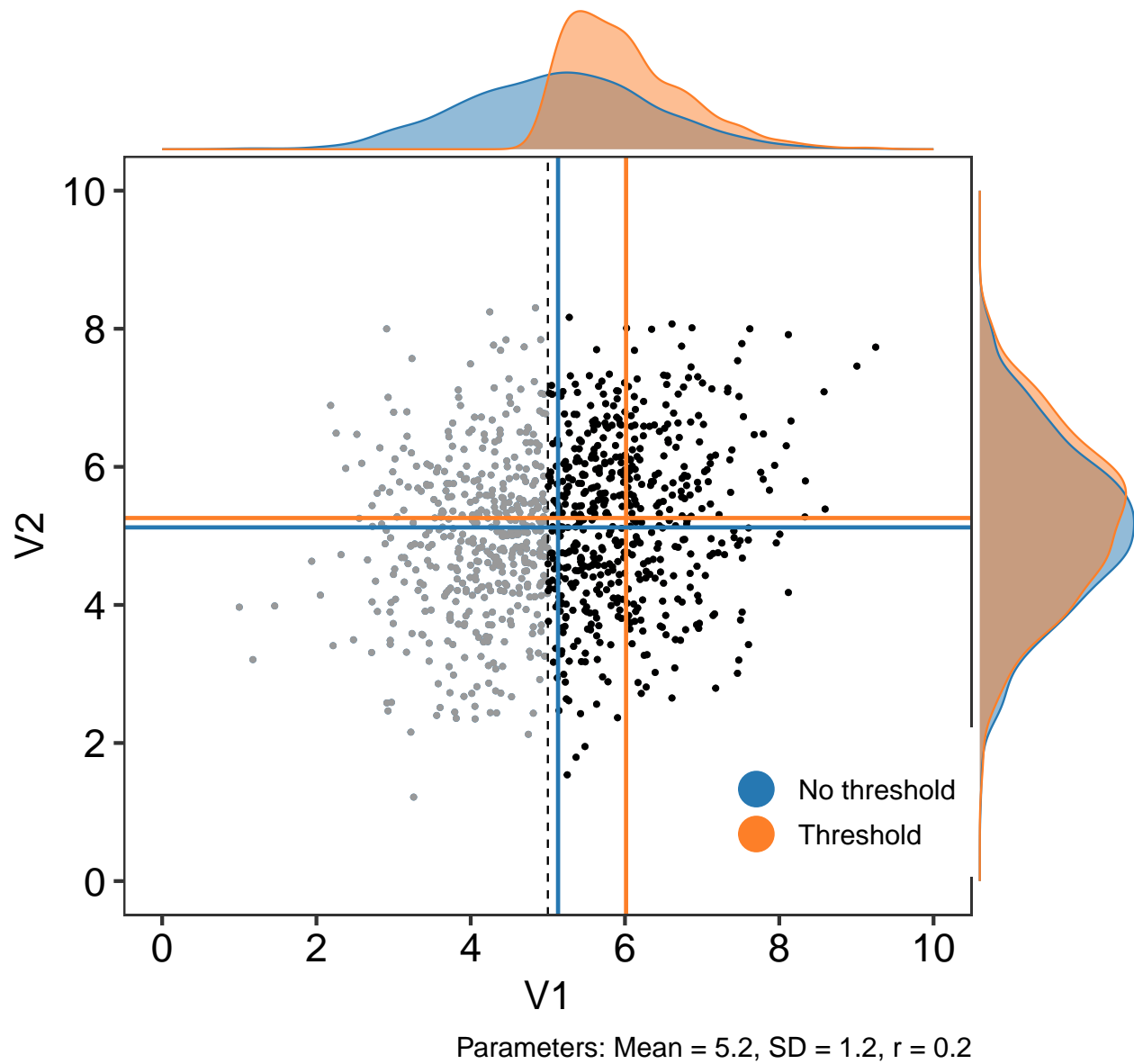
kable(diff_1.5)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
  groupColour = TRUE,
  groupFill = TRUE)
```

D: Baseline pain threshold = 5



Publication plots

```
# Plot 1
shift_1 <- five_1V1.df %>%
  bind_rows(five_1V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
       x = cutoff2,
       fill = time) +
  geom_hline(yintercept = 5.2,
             linetype = 2) +
  geom_point(shape = 21,
```

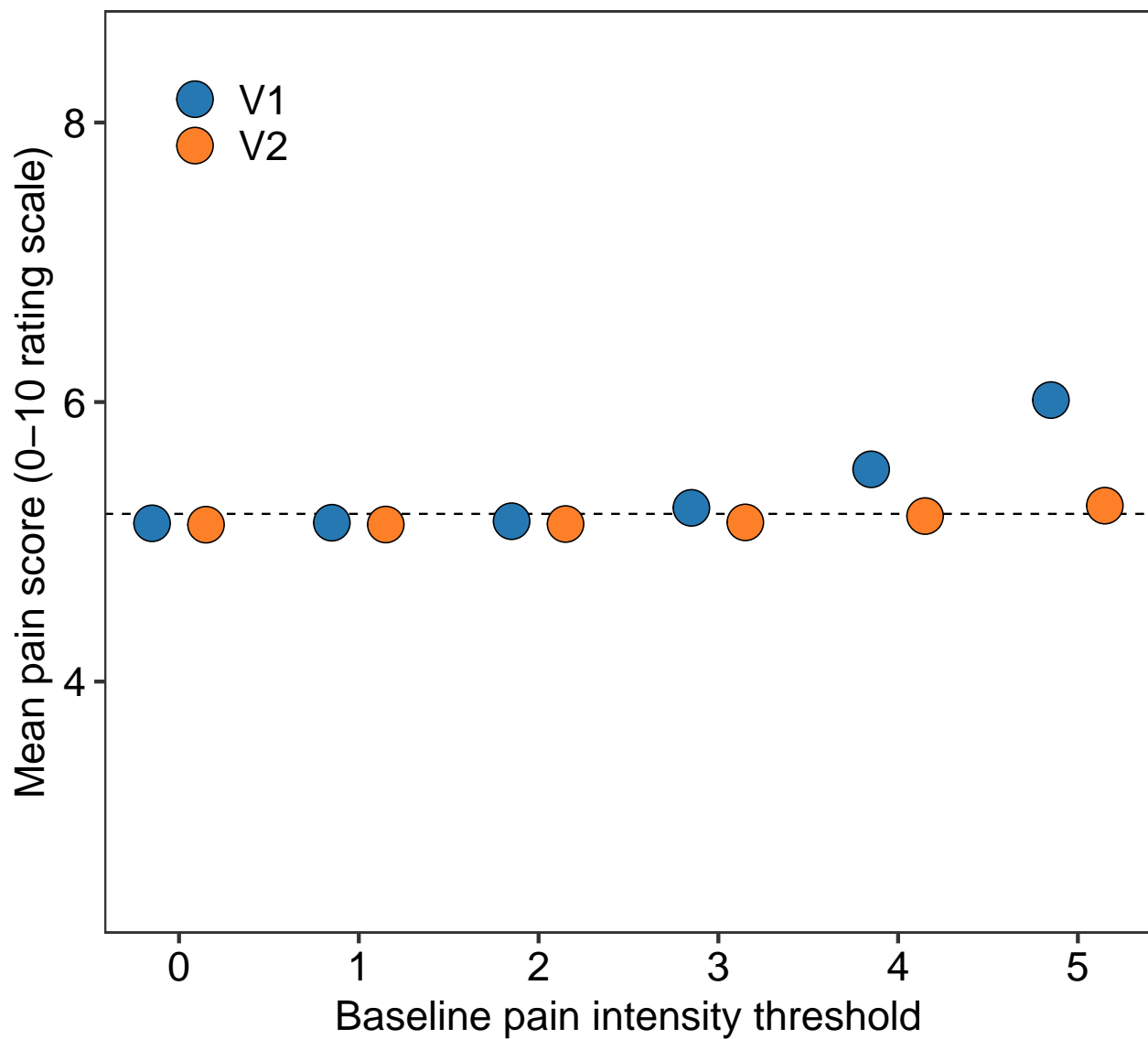
```

      size = 8) +
labs(title = 'A',
      subtitle = 'Parameters: Mean = 5.2, SD = 1.2, r = 0.2',
      x = 'Baseline pain intensity threshold',
      y = 'Mean pain score (0-10 rating scale)') +
scale_x_continuous(breaks = 0:5) +
scale_y_continuous(limits = c(2.5, 8.5)) +
scale_fill_manual(values = pal) +
theme(legend.title = element_blank(),
      legend.position = c(0.12, 0.89),
      legend.text = element_text(size = 20)); shift_1

```

A

Parameters: Mean = 5.2, SD = 1.2, $r = 0.2$



```

# Plot 2
# Bind diff_*. dataframes

```

```

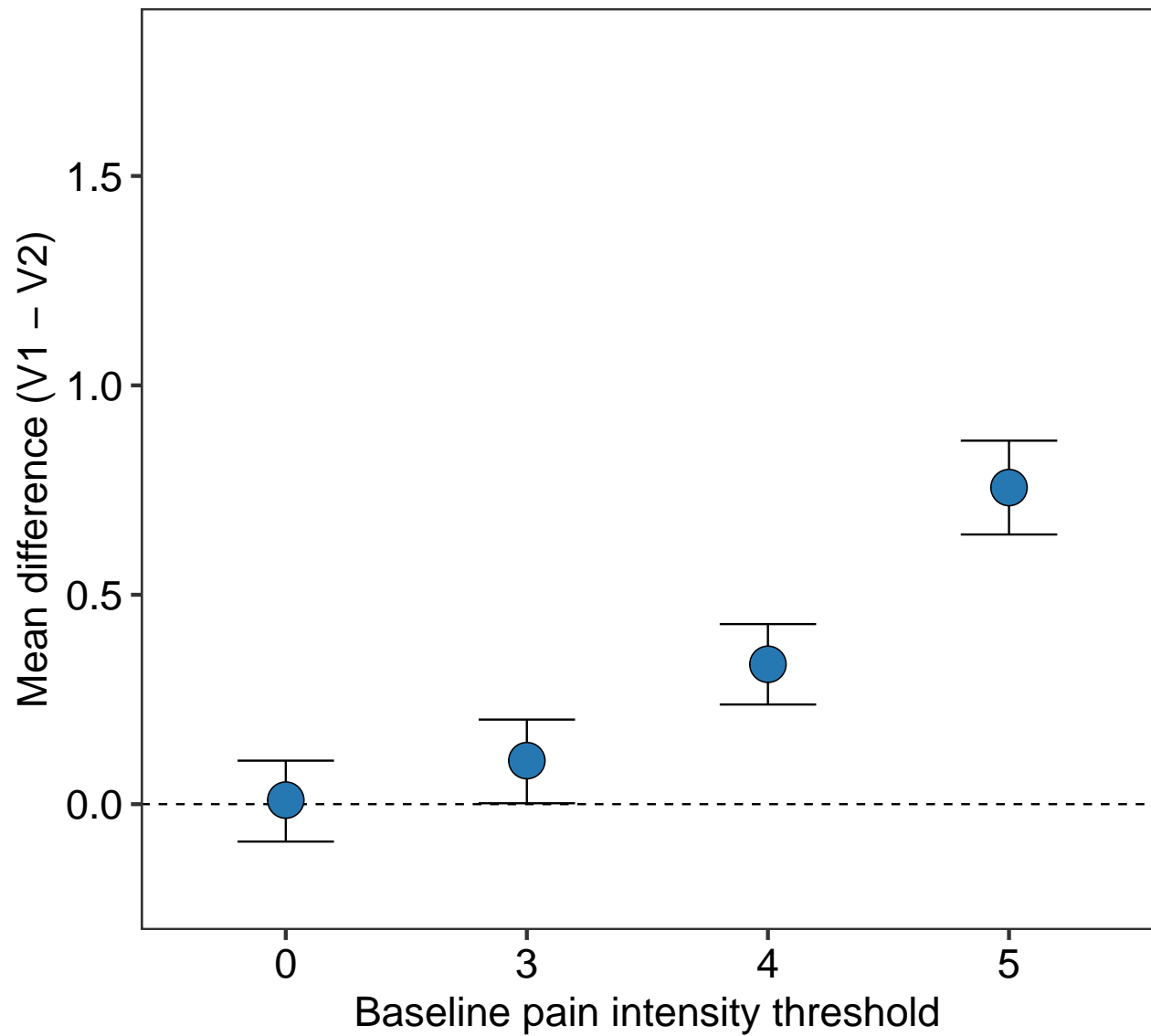
diff_all_1 <- diff_1.0 %>%
  bind_rows(diff_1.3, diff_1.4, diff_1.5)

pp_1 <- diff_all_1 %>%
  mutate(Threshold = factor(.id)) %>%
  ggplot(data = .) +
  aes(x = Threshold,
      y = Mean,
      ymin = Bca.lower,
      ymax = Bca.upper) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_errorbar(width = 0.4) +
  geom_point(shape = 21,
             fill = pal[[1]],
             size = 8) +
  labs(title = 'A',
       subtitle = 'Parameters: Mean = 5.2, SD = 1.2, r = 0.2',
       x = 'Baseline pain intensity threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_1

```

A

Parameters: Mean = 5.2, SD = 1.2, $r = 0.2$



Mean = 5.2, SD = 1.7, $r = 0.2$

Generate and summarise data

Base data

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

# Generate the data
```

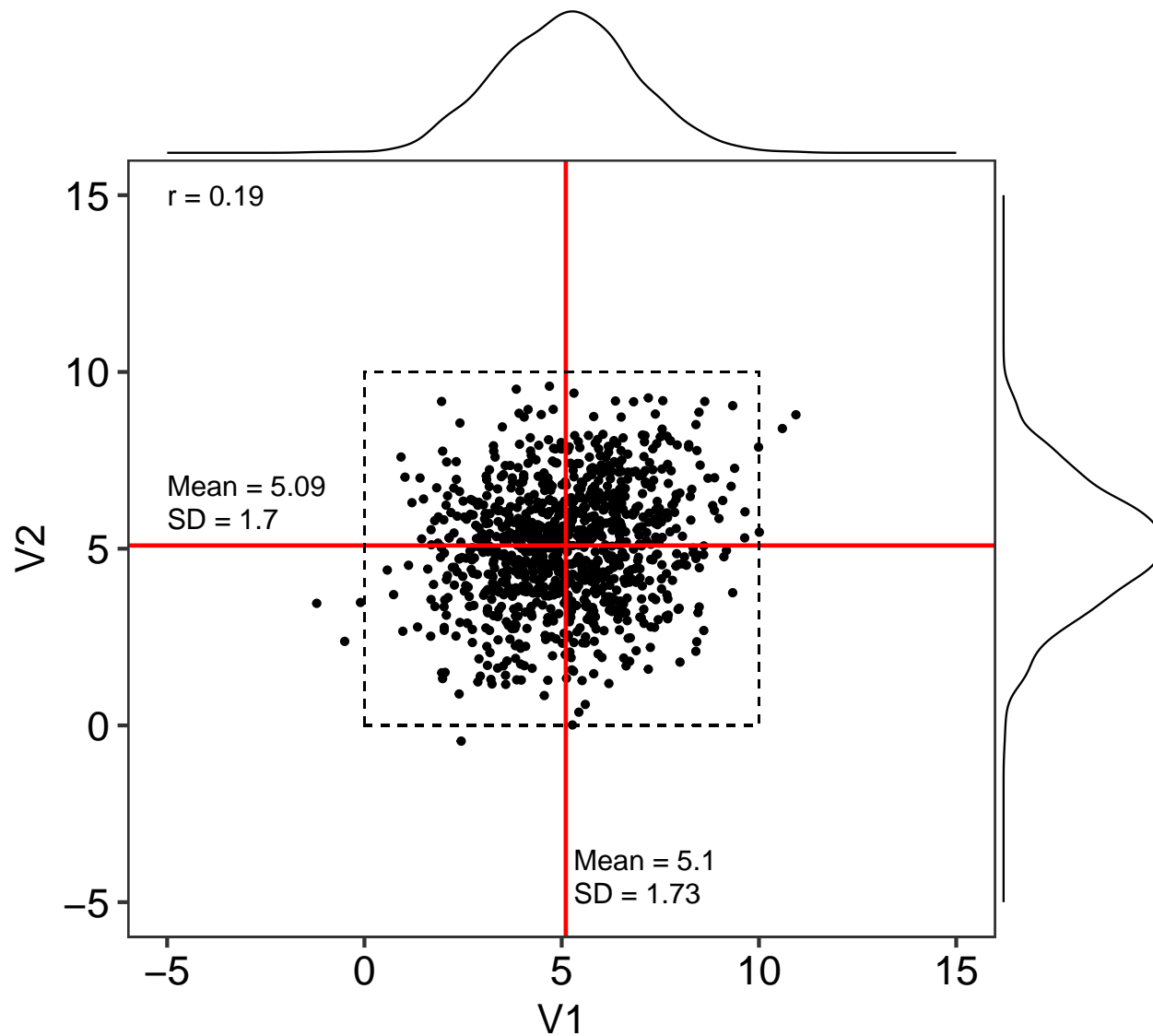
```

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, r = 0.2') +
  scale_y_continuous(limits = c(-5, 15)) +
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))

```

A: Unconstrained



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)) %>%
```

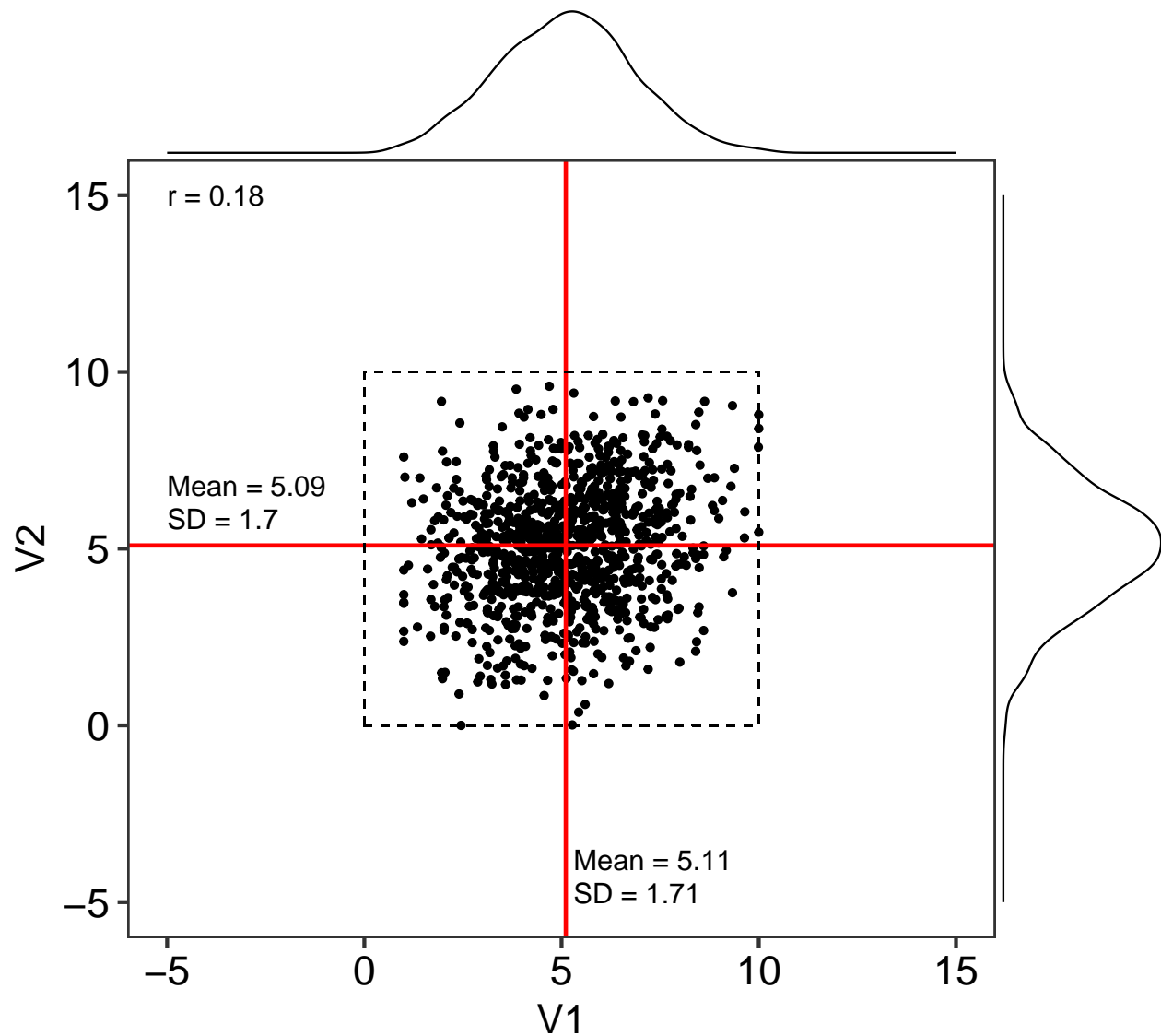
```

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, r = 0.2') +
  scale_y_continuous(limits = c(-5, 15)) +
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))

```


B: Constrained (0–10 range)



Parameters: Mean = 5.2, SD = 1.7, $r = 0.2$

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
```

```

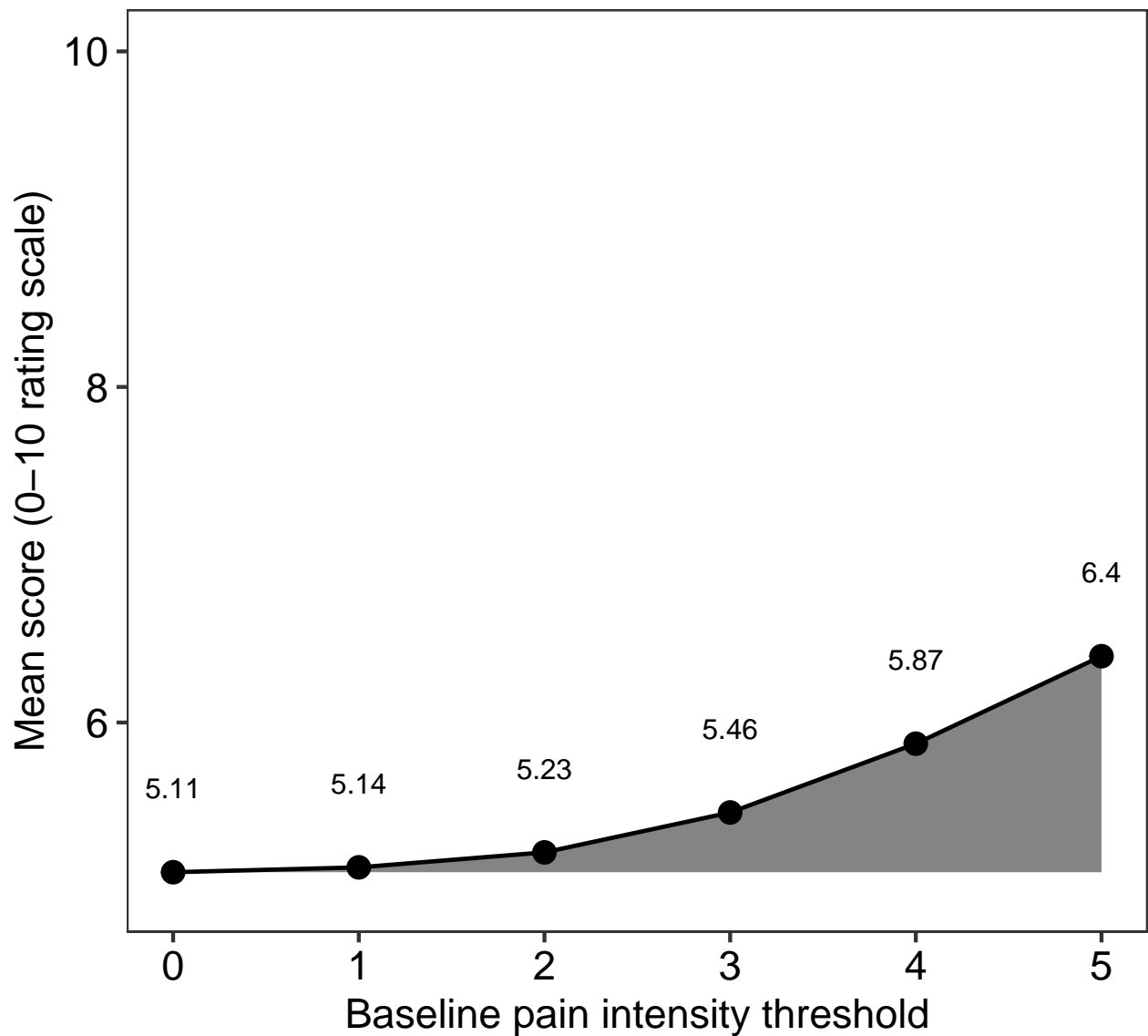
(five_2V1.df <- data.frame(time = 'V1',
                           cutoff = cutoff,
                           cutoff2 = cutoff - 0.15,
                           mean = five_2V1.shift) %>%
  mutate(deviation = mean - mean(five_2V1),
         time = as.character(time)))

##   time cutoff cutoff2    mean deviation
## 1  V1      0   -0.15 5.107766 0.00000000
## 2  V1      1    0.85 5.136723 0.02895706
## 3  V1      2    1.85 5.225986 0.11822051
## 4  V1      3    2.85 5.463707 0.35594105
## 5  V1      4    3.85 5.873839 0.76607343
## 6  V1      5    4.85 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, r = 0.2',
       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 = 5.2, SD = 1.7, $r = 0.2$

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(.))

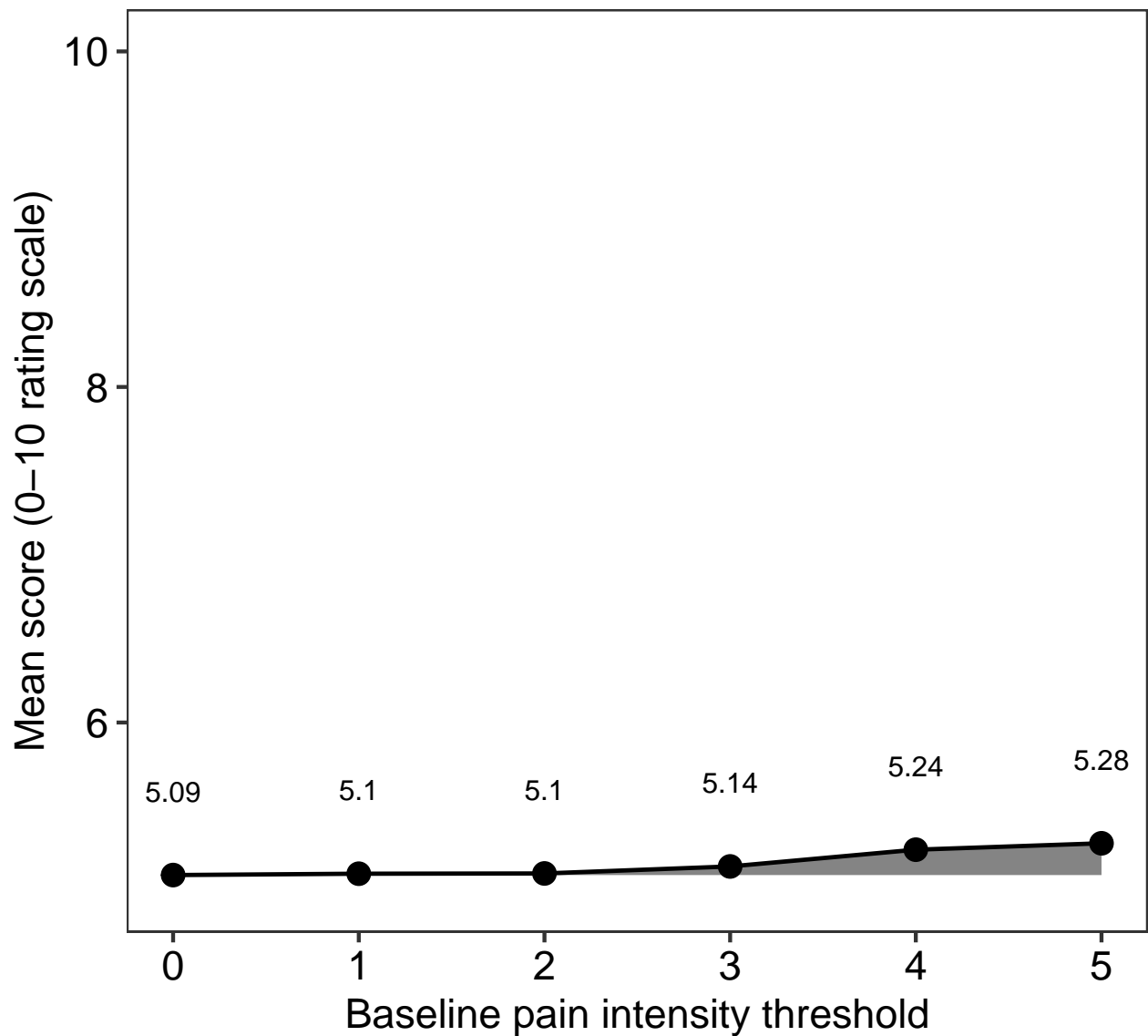
(five_2V2.df <- data.frame(time = 'V2',
    cutoff = cutoff,
    cutoff2 = cutoff + 0.15,
    mean = five_2V2.shift) %>%
    mutate(deviation = mean - mean(five_2V1),
    time = as.character(time)))

##   time cutoff cutoff2    mean  deviation
## 1   V2      0    0.15 5.090824 -0.016941534
## 2   V2      1    1.15 5.098853 -0.008912712
## 3   V2      2    2.15 5.100386 -0.007379513
## 4   V2      3    3.15 5.142141  0.034375676
## 5   V2      4    4.15 5.241978  0.134212195
## 6   V2      5    5.15 5.280114  0.172348592

# 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, r = 0.2',
    x = 'Baseline pain intensity threshold',
    y = 'Mean score (0-10 rating scale)') +
  theme(plot.caption = element_text(size = 14))

```

B: Shift in V2 mean with increasing V1 threshold value



Parameters: Mean = 5.2, SD = 1.7, $r = 0.2$

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

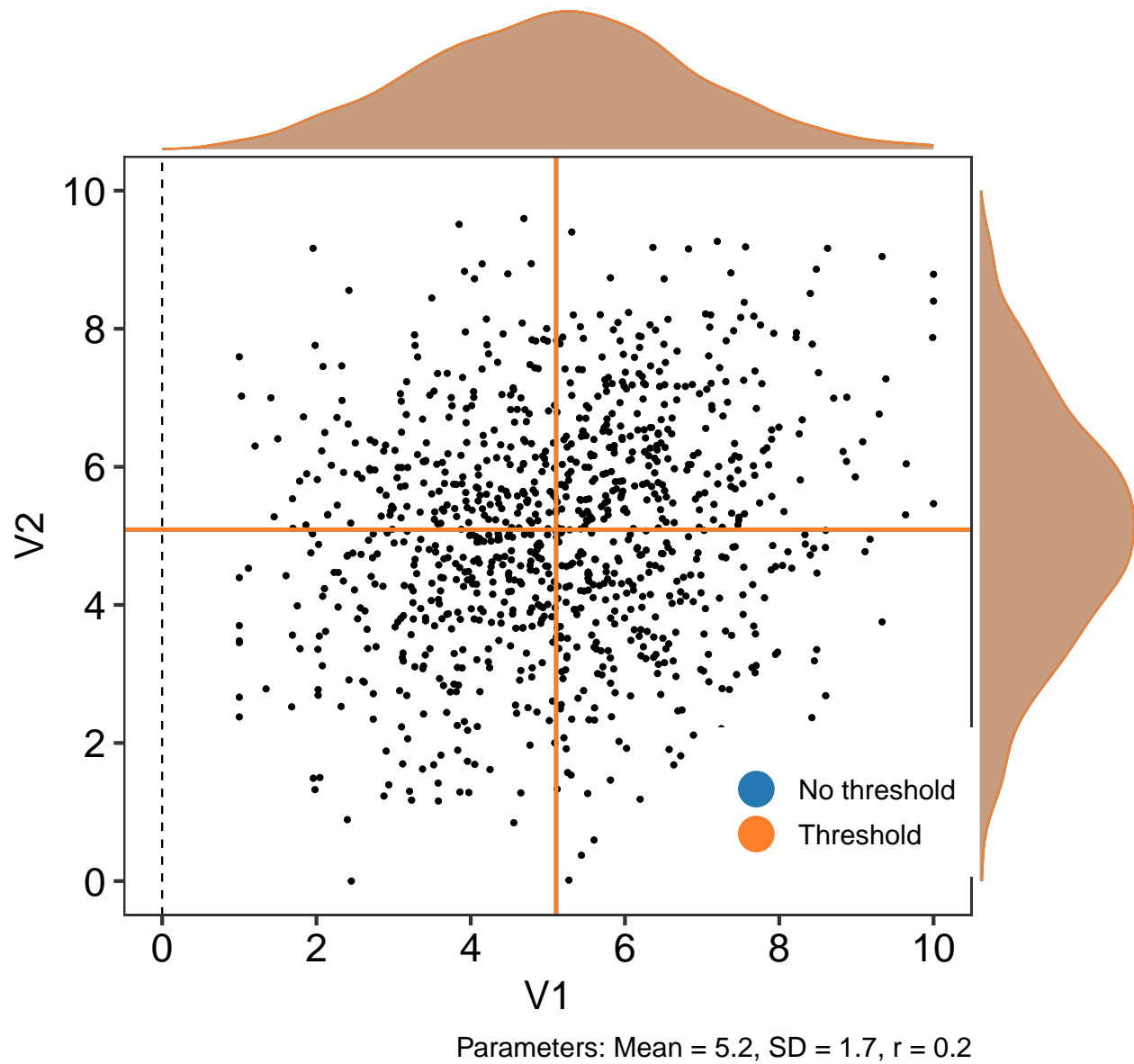
```

# 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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
groupColour = TRUE,
groupFill = TRUE)

```

A: Baseline pain threshold = 0



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

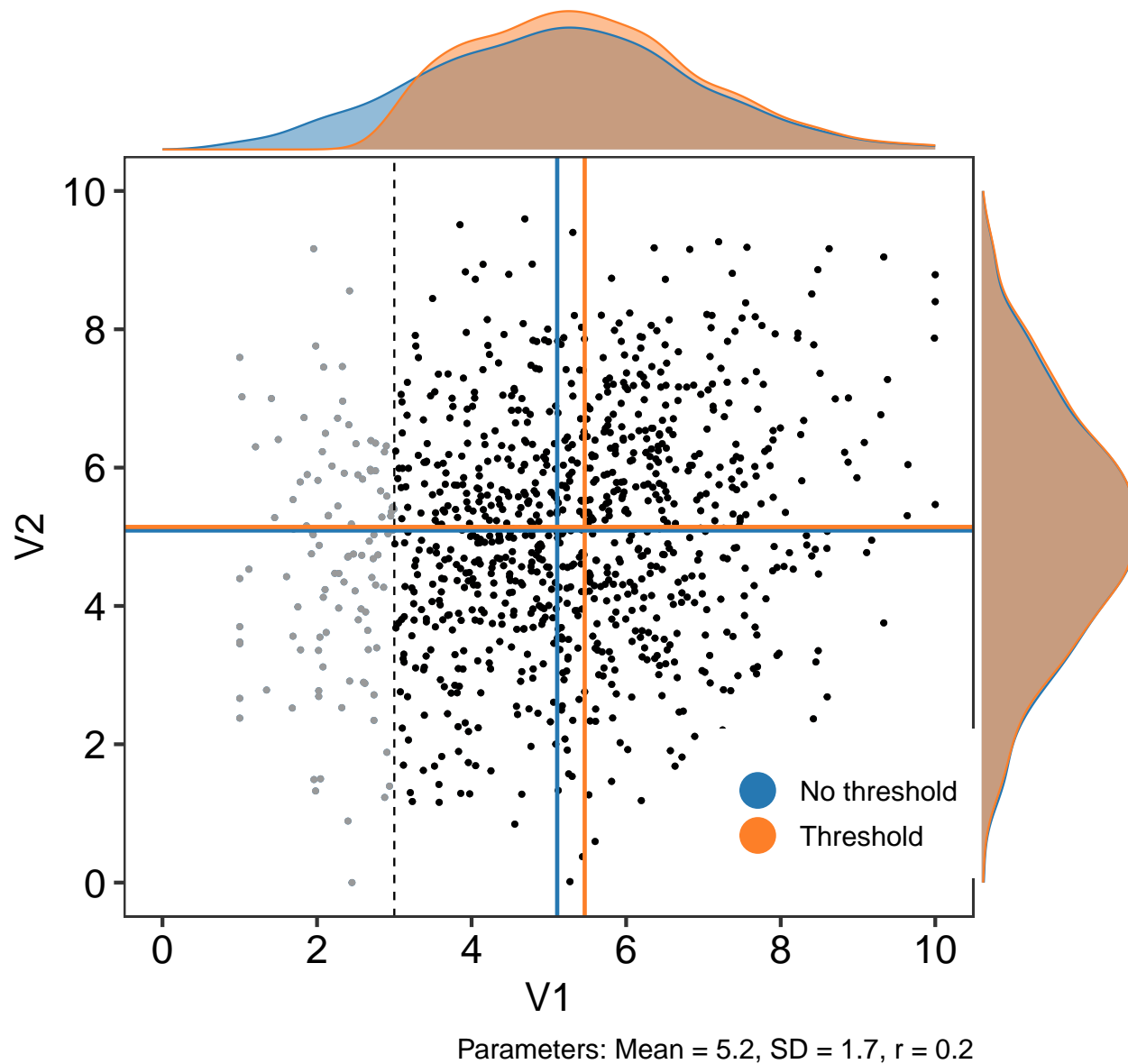
kable(diff_2.3)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

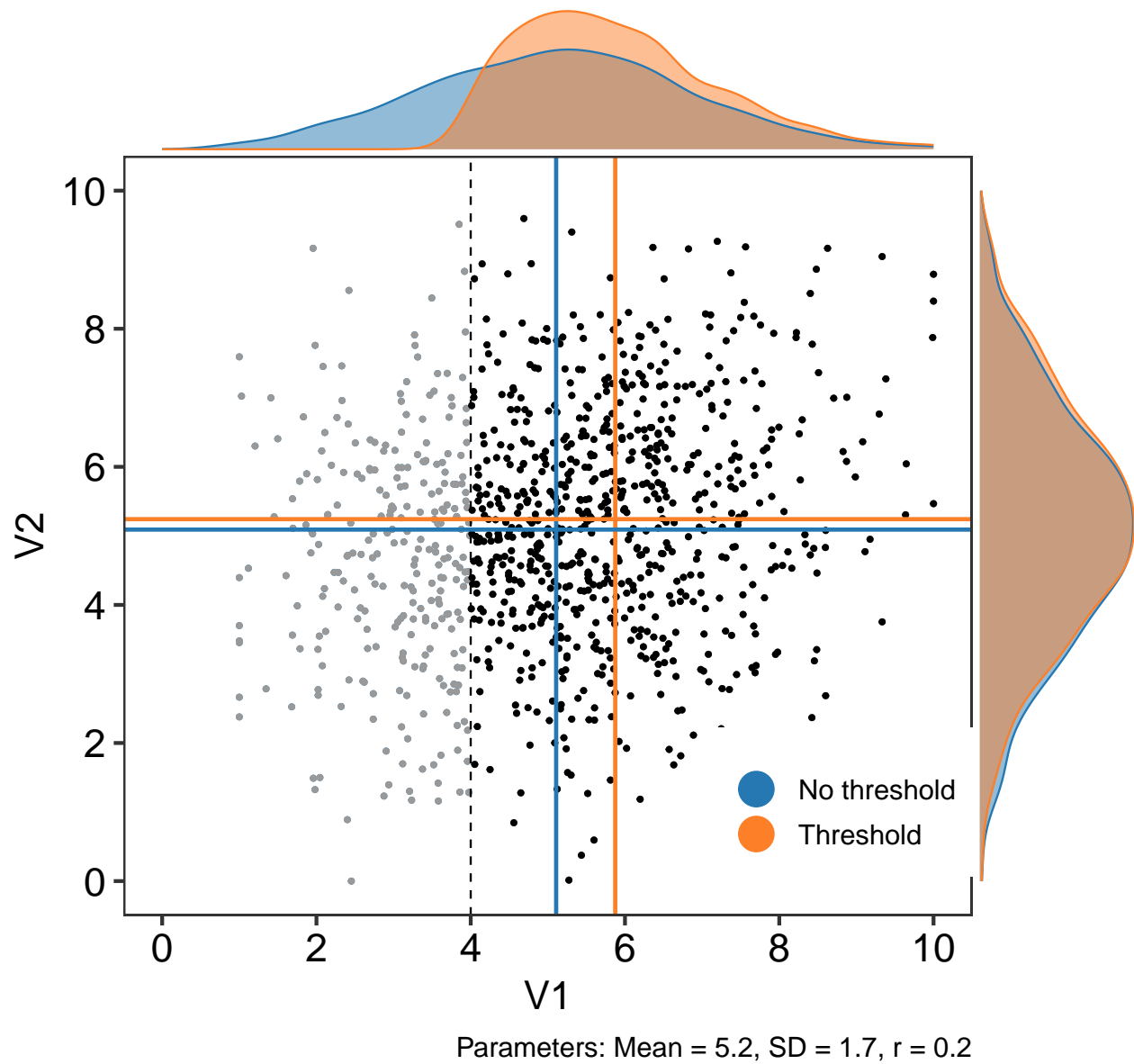
kable(diff_2.4)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

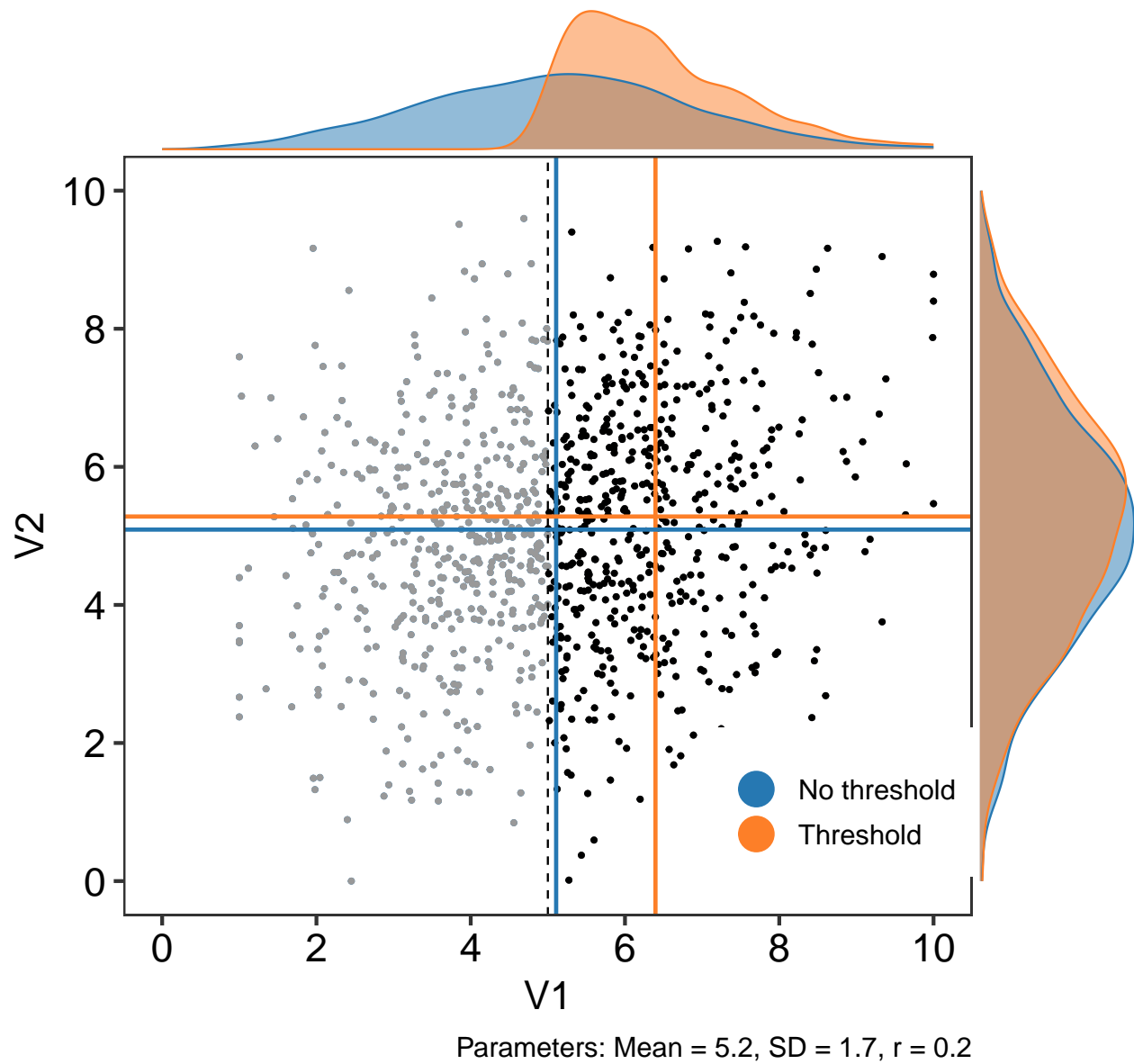
kable(diff_2.5)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
  groupColour = TRUE,
  groupFill = TRUE)
```

D: Baseline pain threshold = 5



Publication plots

```
# Plot 1
shift_2 <- five_2V1.df %>%
  bind_rows(five_2V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
       x = cutoff2,
       fill = time) +
  geom_hline(yintercept = 5.2,
             linetype = 2) +
  geom_point(shape = 21,
```

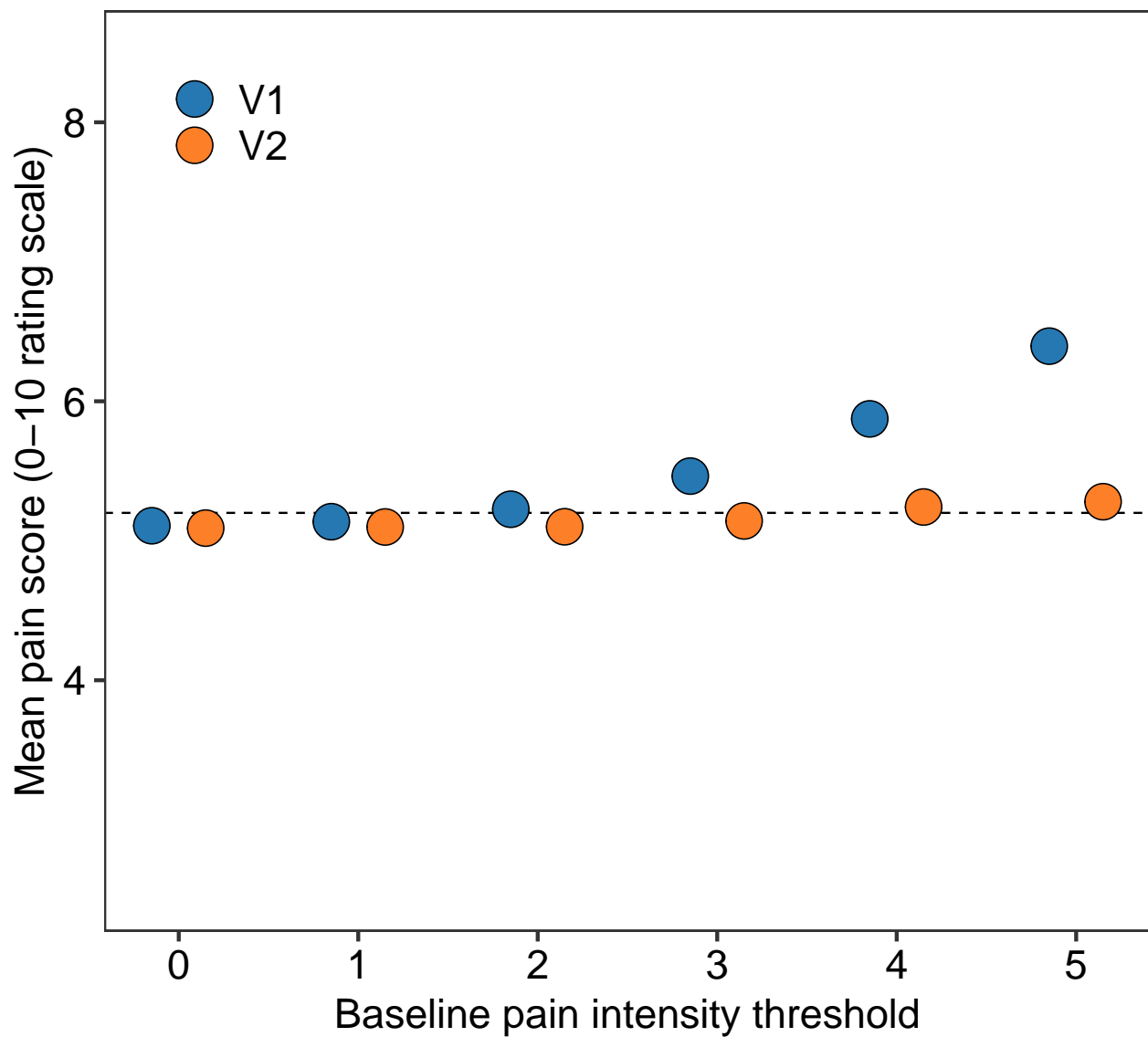
```

      size = 8) +
labs(title = 'B',
      subtitle = 'Parameters: Mean = 5.2, SD = 1.7, r = 0.2',
      x = 'Baseline pain intensity threshold',
      y = 'Mean pain score (0-10 rating scale)') +
scale_x_continuous(breaks = 0:5) +
scale_y_continuous(limits = c(2.5, 8.5)) +
scale_fill_manual(values = pal) +
theme(legend.title = element_blank(),
      legend.position = c(0.12, 0.89),
      legend.text = element_text(size = 20)); shift_2

```

B

Parameters: Mean = 5.2, SD = 1.7, $r = 0.2$



```

# Plot 2
# Bind diff_*. dataframes

```

```

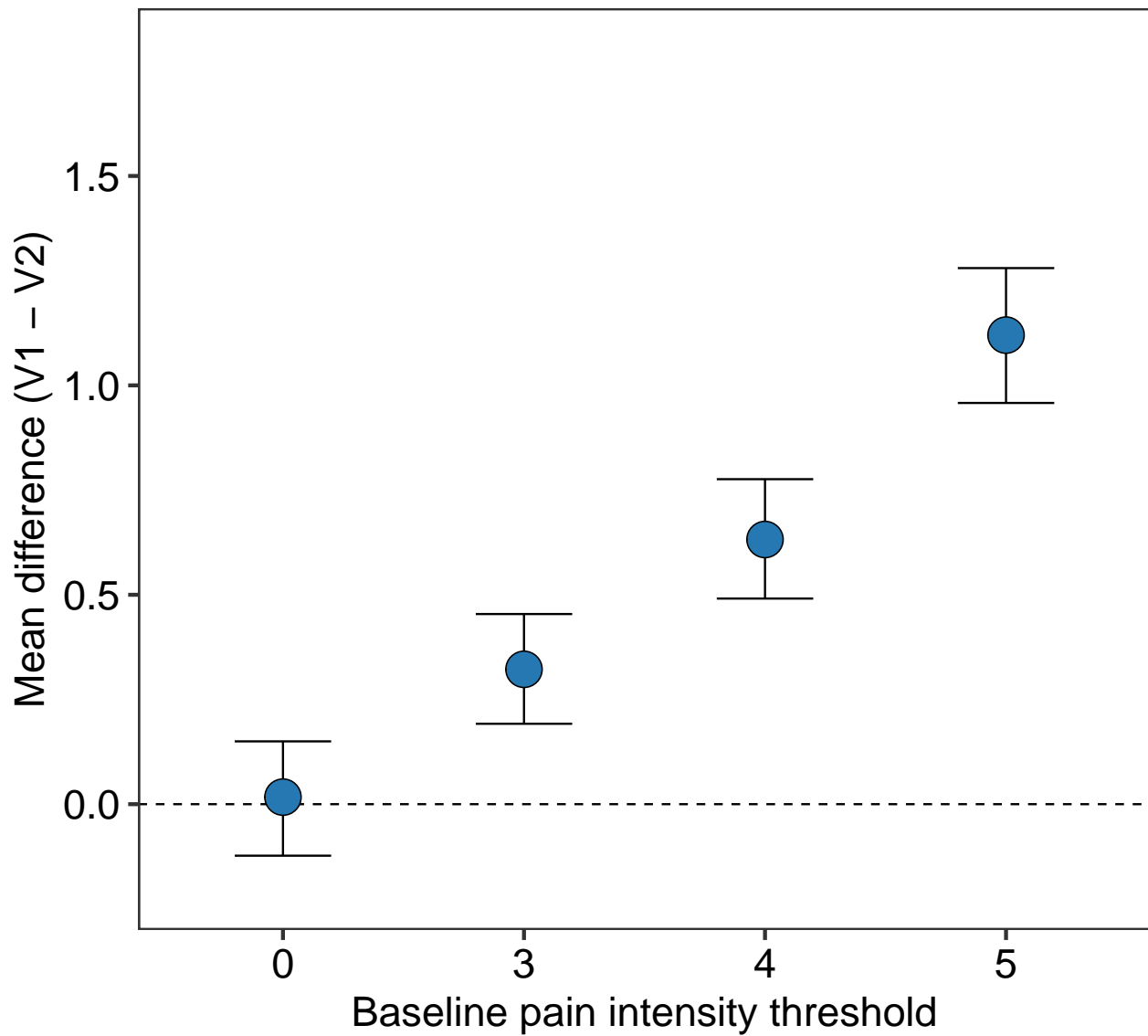
diff_all_2 <- diff_2.0 %>%
  bind_rows(diff_2.3, diff_2.4, diff_2.5)

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, r = 0.2',
       x = 'Baseline pain intensity threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_2

```

B

Parameters: Mean = 5.2, SD = 1.7, $r = 0.2$



Mean = 5.2, SD = 2.2, $r = 0.2$

Generate and summarise data

Base data

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

# Generate the data
```



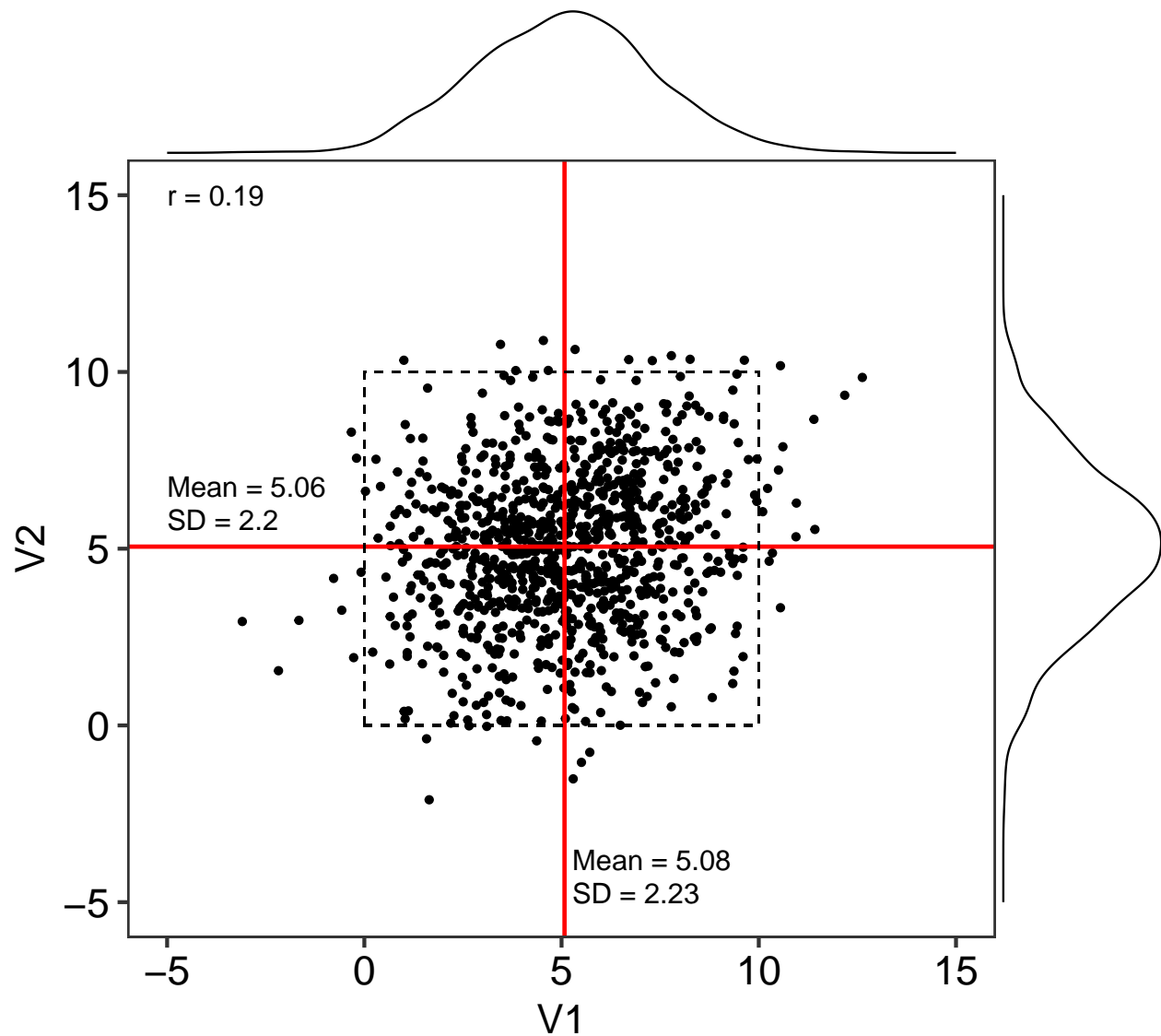
```

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, r = 0.2') +
  scale_y_continuous(limits = c(-5, 15)) +
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))

```

A: Unconstrained



Parameters: Mean = 5.2, SD = 2.2, $r = 0.2$

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)) %>%
```

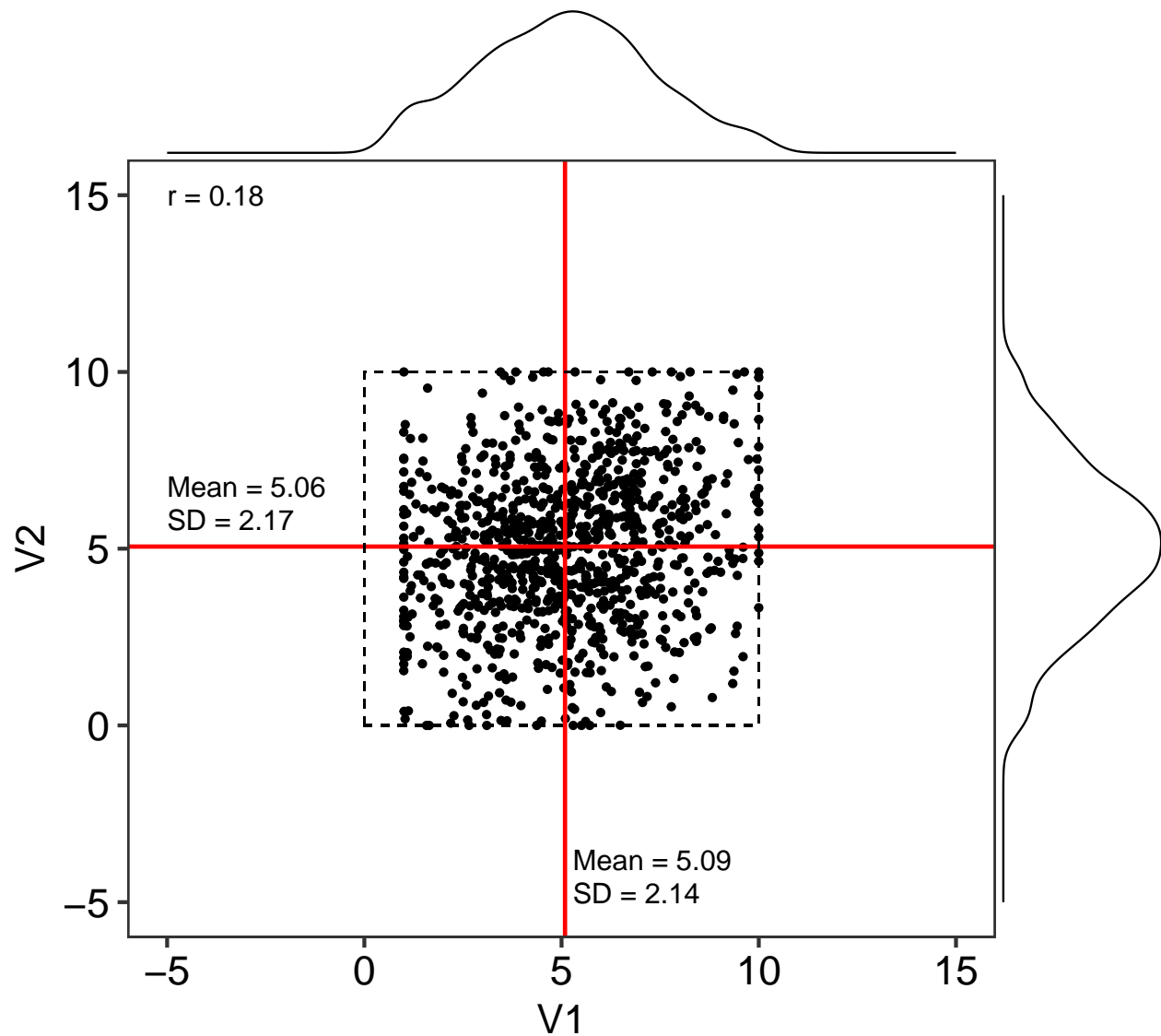
```

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, r = 0.2') +
  scale_y_continuous(limits = c(-5, 15)) +
  scale_x_continuous(limits = c(-5, 15)) +
  theme(plot.caption = element_text(size = 14)))

```

B: Constrained (0–10 range)



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])})

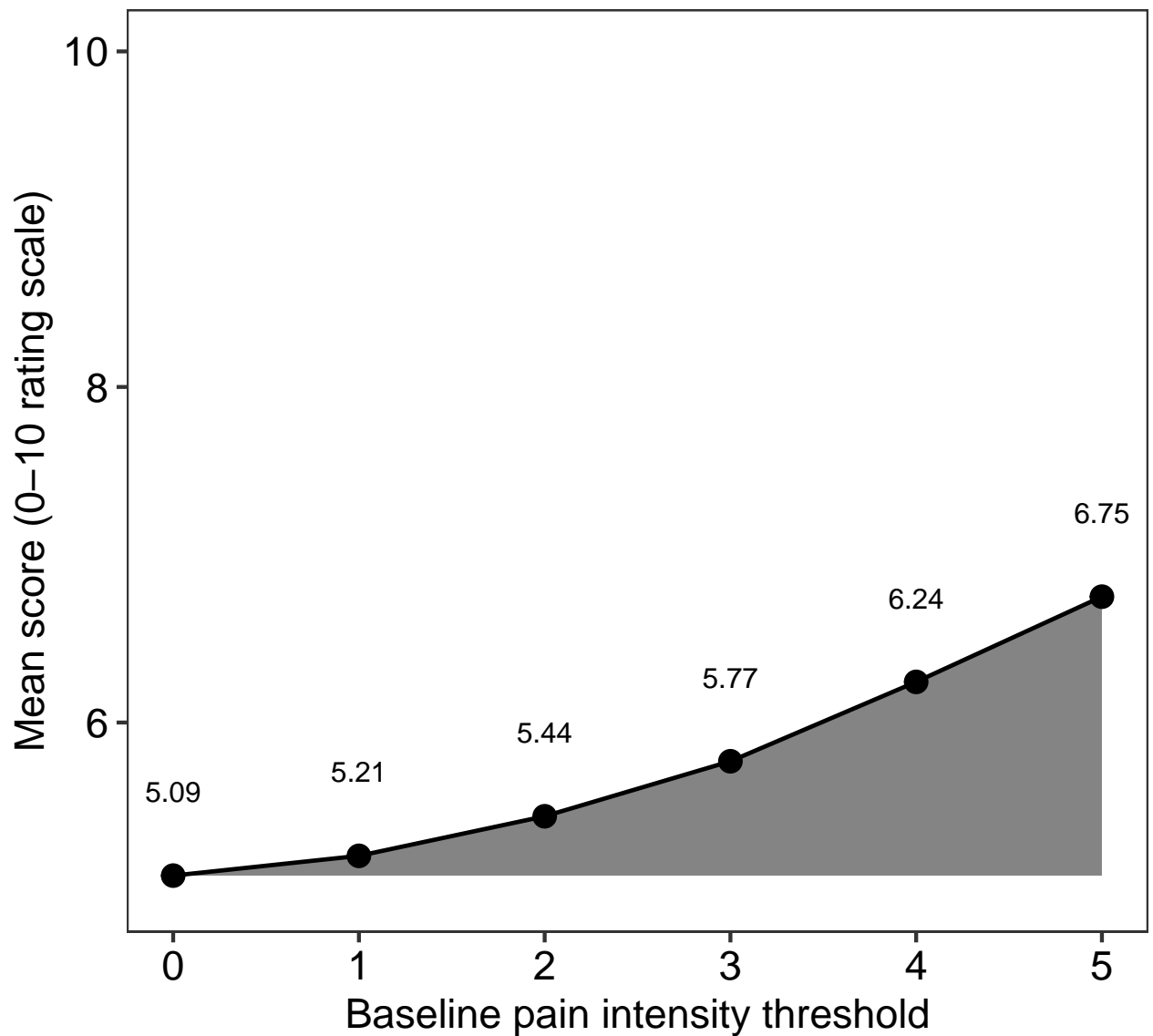
# Calculate deviation
```

```
(five_3V1.df <- data.frame(time = 'V1',
                           cutoff = cutoff,
                           cutoff2 = cutoff - 0.15,
                           mean = five_3V1.shift) %>%
  mutate(deviation = mean - mean(five_3V1),
         time = as.character(time)))
```

```
##   time cutoff cutoff2    mean deviation
## 1  V1      0   -0.15 5.087557 0.0000000
## 2  V1      1    0.85 5.205305 0.1177486
## 3  V1      2    1.85 5.441239 0.3536825
## 4  V1      3    2.85 5.769110 0.6815530
## 5  V1      4    3.85 6.242234 1.1546772
## 6  V1      5    4.85 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, r = 0.2',
       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 = 5.2, SD = 2.2, $r = 0.2$

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(.))

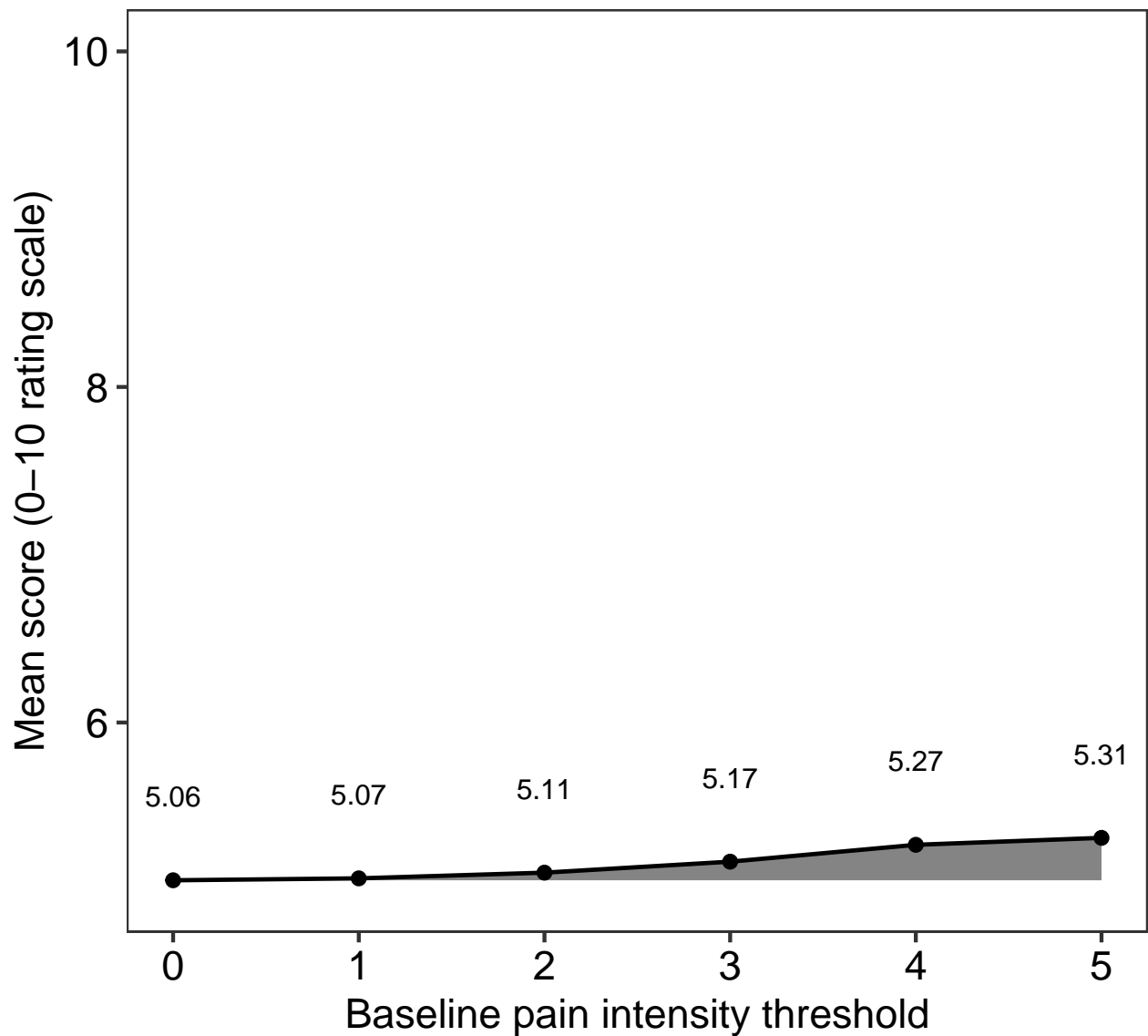
(five_3V2.df <- data.frame(time = 'V2',
                           cutoff = cutoff,
                           cutoff2 = cutoff + 0.15,
                           mean = five_3V2.shift) %>%
  mutate(deviation = mean - mean(five_3V1),
         time = as.character(time)))

##   time cutoff cutoff2    mean  deviation
## 1  V2      0    0.15 5.059701 -0.02785545
## 2  V2      1    1.15 5.071221 -0.01633581
## 3  V2      2    2.15 5.105124  0.01756751
## 4  V2      3    3.15 5.170639  0.08308252
## 5  V2      4    4.15 5.271064  0.18350676
## 6  V2      5    5.15 5.312531  0.22497378

# 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, r = 0.2',
       x = 'Baseline pain intensity threshold',
       y = 'Mean score (0-10 rating scale)') +
  theme(plot.caption = element_text(size = 14))

```

B: Shift in V2 mean with increasing V1 threshold value



Parameters: Mean = 5.2, SD = 2.2, $r = 0.2$

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

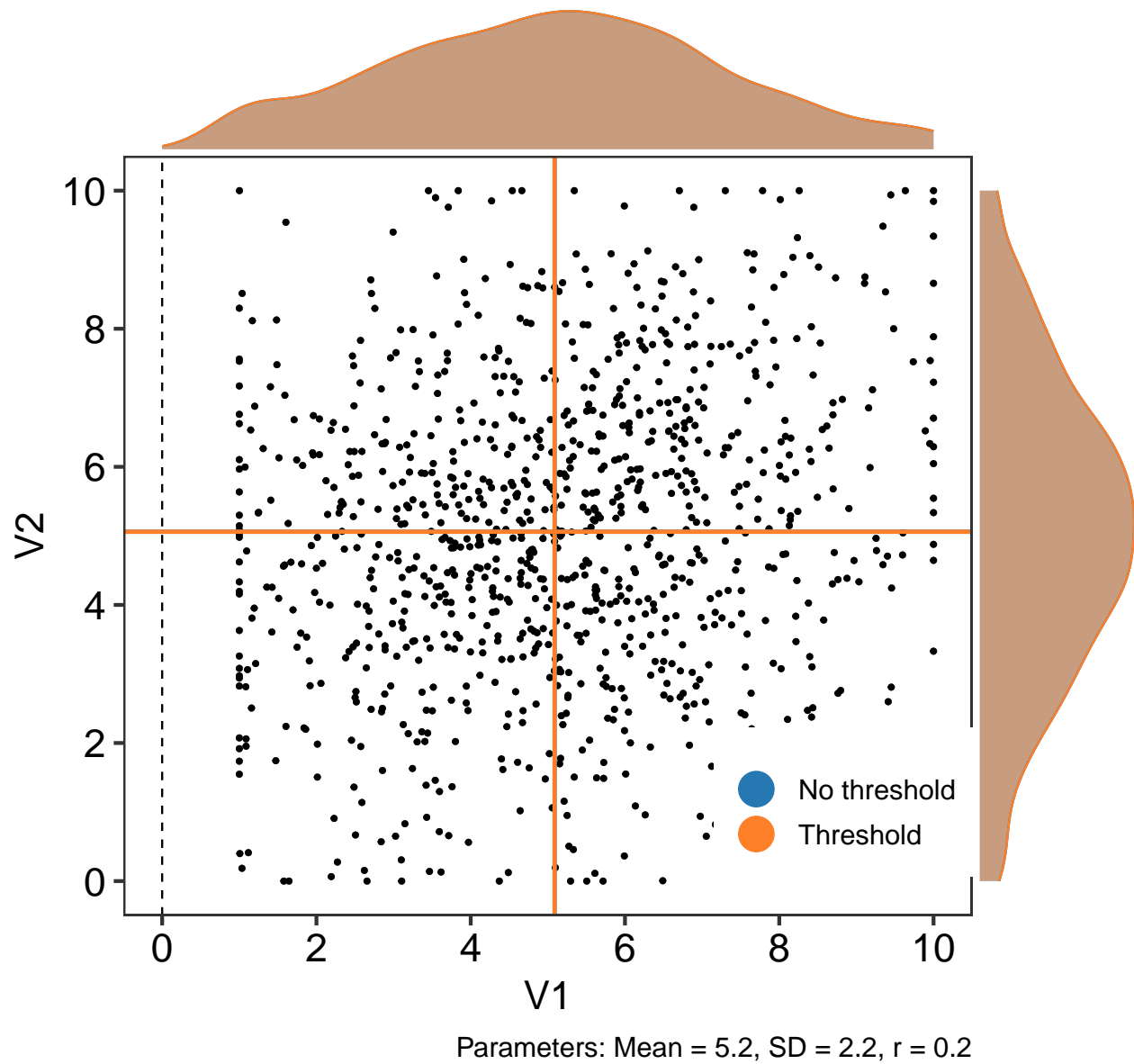
```

# 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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
groupColour = TRUE,
groupFill = TRUE)

```

A: Baseline pain threshold = 0



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

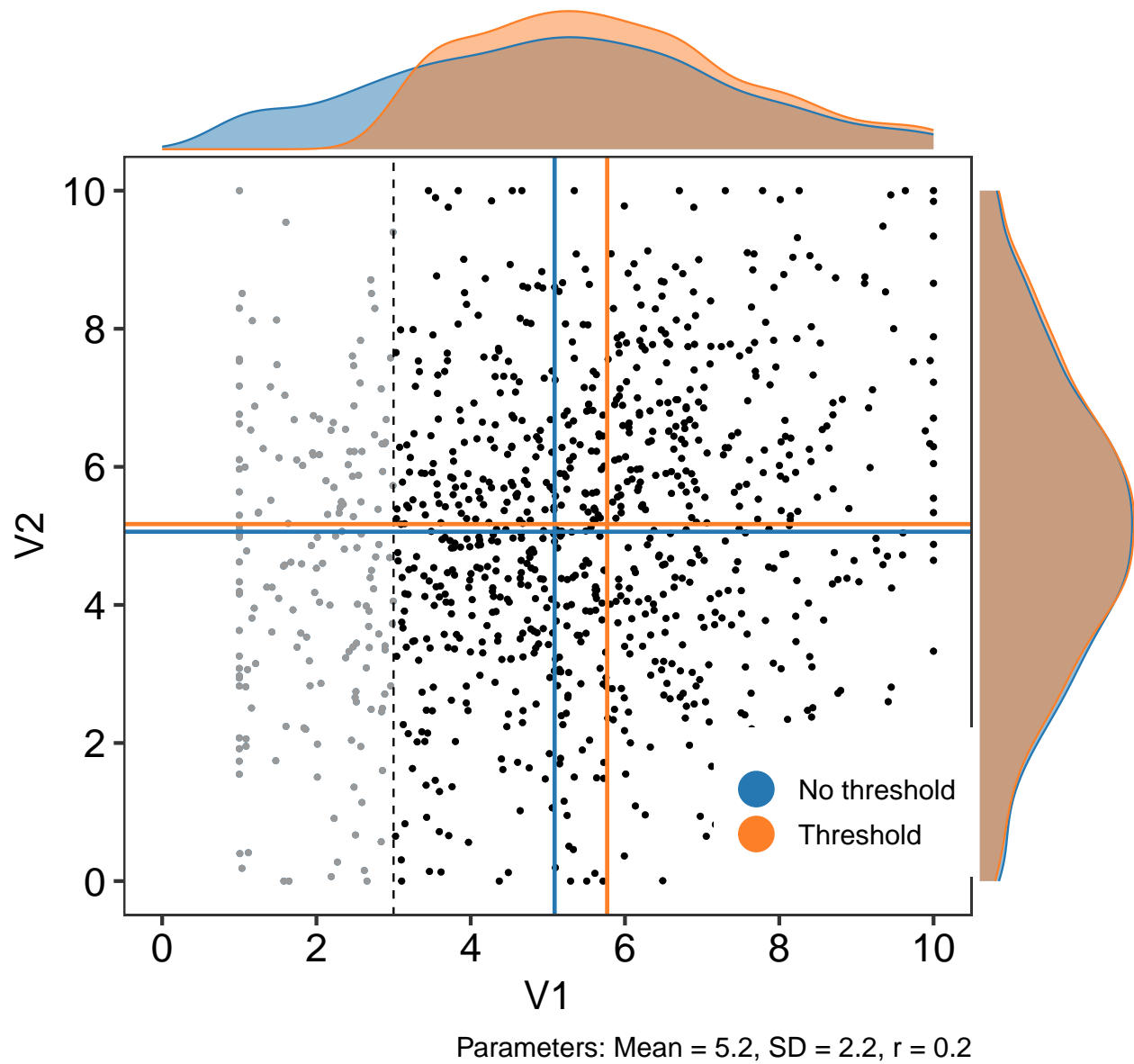
kable(diff_3.3)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

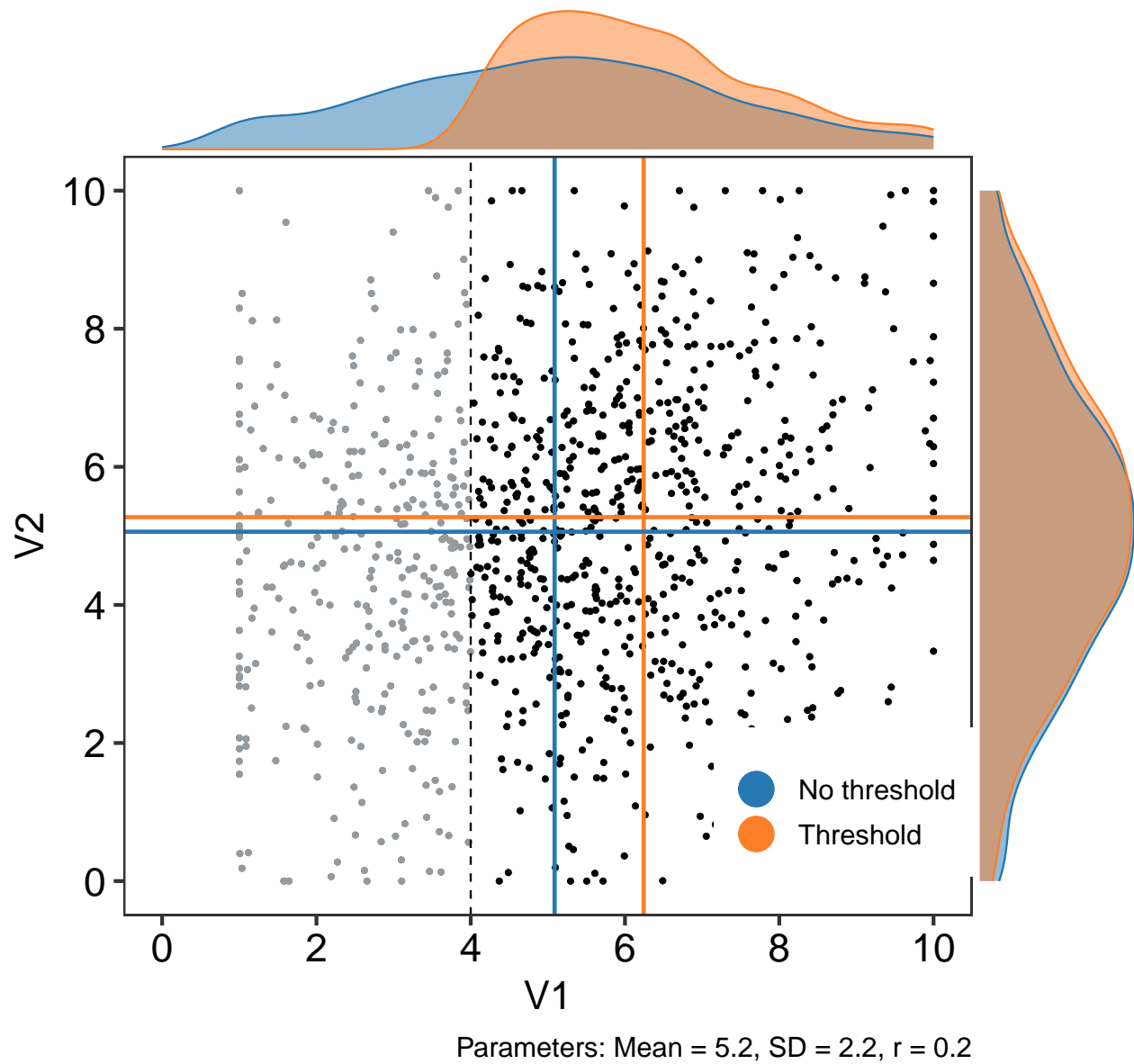
kable(diff_3.4)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  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



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

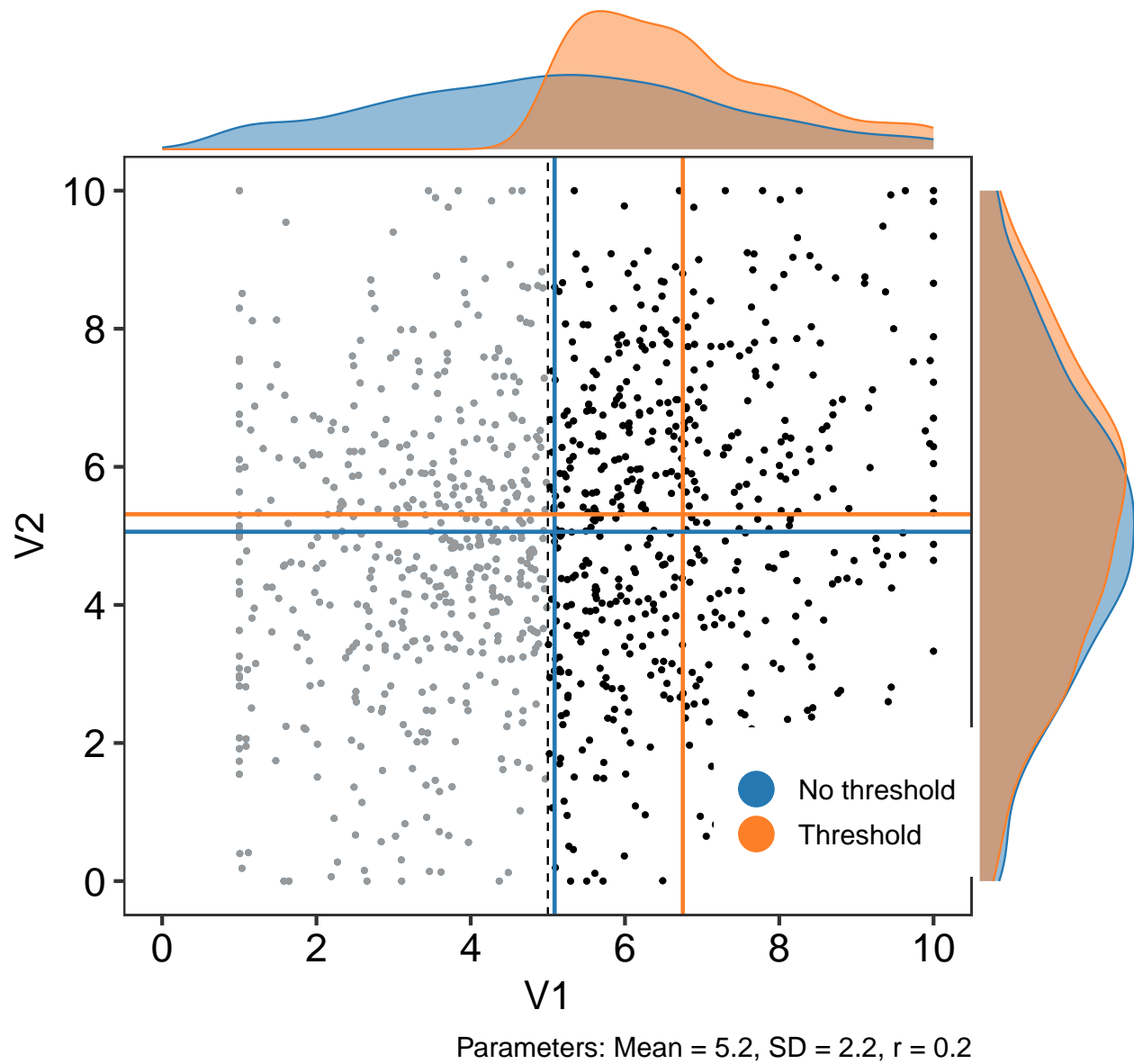
kable(diff_3.5)
```

.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) +
  guides(colour = guide_legend(override.aes = list(size = 8))) +
  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, r = 0.2') +
  theme(legend.title = element_blank(),
    legend.position = c(0.85, 0.15),
    plot.caption = element_text(size = 14)),
  groupColour = TRUE,
  groupFill = TRUE)
```

D: Baseline pain threshold = 5



Publication plots

```
# Plot 1
shift_3 <- five_3V1.df %>%
  bind_rows(five_3V2.df) %>%
  ggplot(data = .) +
  aes(y = mean,
       x = cutoff2,
       fill = time) +
  geom_hline(yintercept = 5.2,
             linetype = 2) +
  geom_point(shape = 21,
```



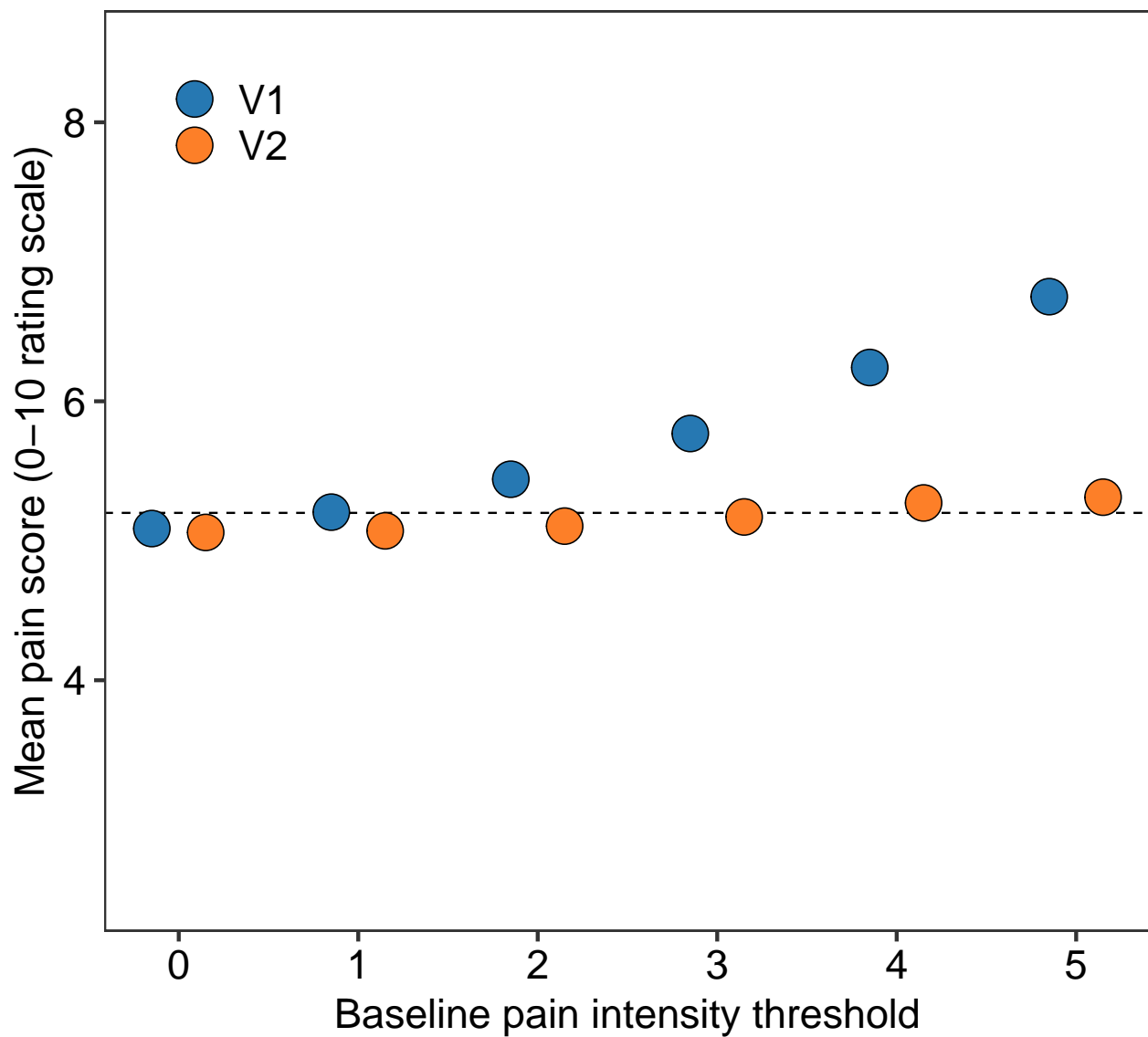
```

      size = 8) +
labs(title = 'C',
      subtitle = 'Parameters: Mean = 5.2, SD = 2.2, r = 0.2',
      x = 'Baseline pain intensity threshold',
      y = 'Mean pain score (0-10 rating scale)') +
scale_x_continuous(breaks = 0:5) +
scale_y_continuous(limits = c(2.5, 8.5)) +
scale_fill_manual(values = pal) +
theme(legend.title = element_blank(),
      legend.position = c(0.12, 0.89),
      legend.text = element_text(size = 20)); shift_3

```

C

Parameters: Mean = 5.2, SD = 2.2, $r = 0.2$



```

# Plot 2
# Bind diff_*. dataframes

```

```

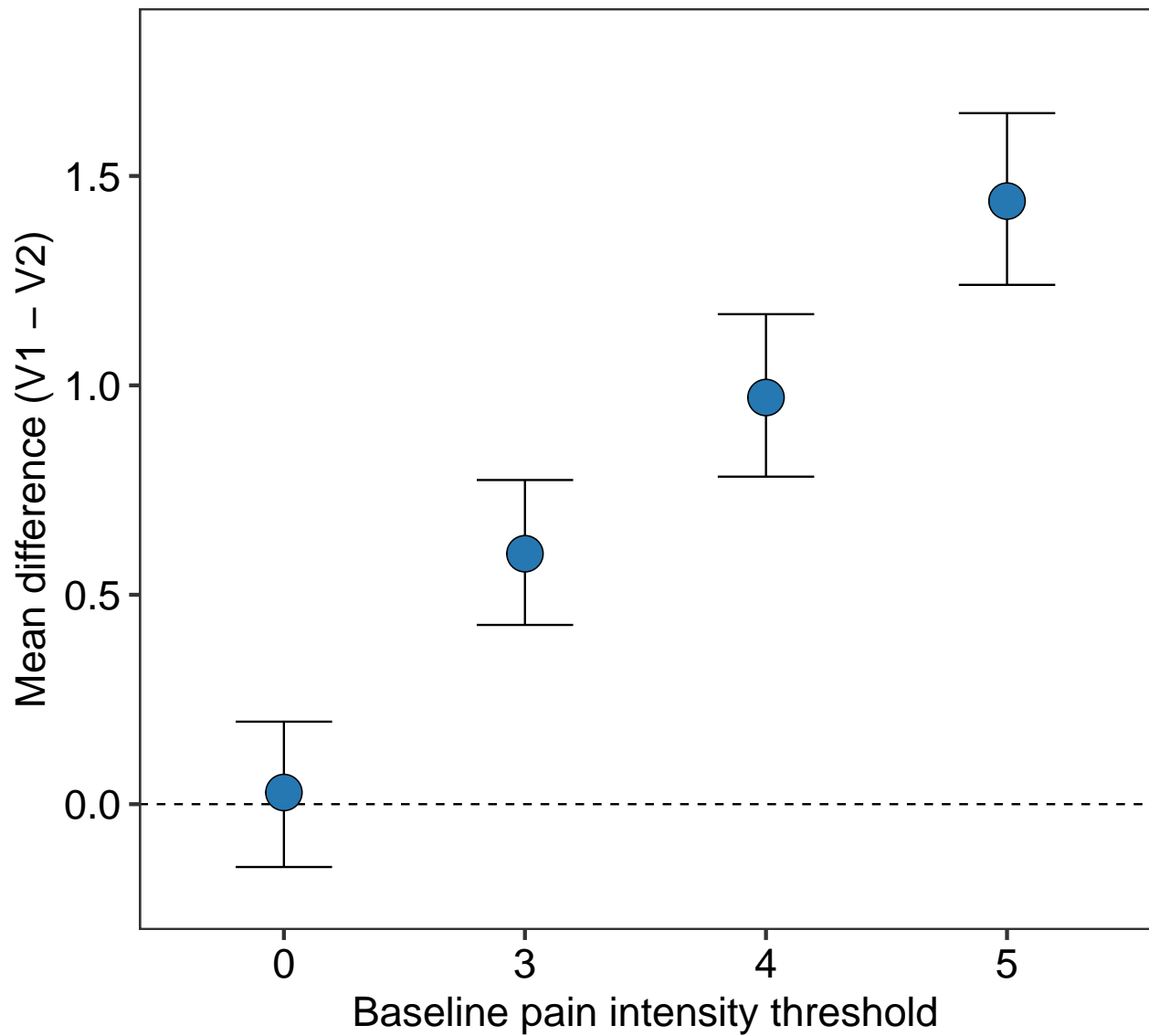
diff_all_3 <- diff_3.0 %>%
  bind_rows(diff_3.3, diff_3.4, diff_3.5)

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, r = 0.2',
       x = 'Baseline pain intensity threshold',
       y = 'Mean difference (V1 - V2)') +
  scale_y_continuous(limits = c(-0.2, 1.8)); pp_3

```

C

Parameters: Mean = 5.2, SD = 2.2, $r = 0.2$



Publication composite plots

```
shift_4 <- shift_1 + shift_2 + shift_3
ggsave('figures/5.2_0.2a.png', shift_4, width = 17, height = 7)

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

Session information

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## 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_1.25      MBESS_4.6.0     ggExtra_0.9
## [5] rcompanion_2.3.7 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_1.0.0      tibble_2.1.3    ggplot2_3.2.1   tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] httr_1.4.1      jsonlite_1.6     splines_3.6.1
## [4] modelr_0.1.5    shiny_1.3.2      assertthat_0.2.1
## [7] expm_0.999-4    highr_0.8        stats4_3.6.1
## [10] coin_1.3-1      cellranger_1.1.0 yaml_2.2.0
## [13] pillar_1.4.2    backports_1.1.5  lattice_0.20-38
## [16] glue_1.3.1      digest_0.6.21    promises_1.1.0
## [19] rvest_0.3.4     colorspace_1.4-1 sandwich_2.5-1
## [22] httpuv_1.5.2    htmltools_0.4.0  Matrix_1.2-17
## [25] plyr_1.8.4      pkgconfig_2.0.3  broom_0.5.2
## [28] haven_2.1.1     EMT_1.1          xtable_1.8-4
## [31] mvtnorm_1.0-11  scales_1.0.0     later_1.0.0
## [34] generics_0.0.2  TH.data_1.0-10   withr_2.1.2
## [37] lazyeval_0.2.2  cli_1.1.0        mime_0.7
## [40] survival_2.44-1.1 crayon_1.3.4     readxl_1.3.1
## [43] evaluate_0.14   nlme_3.1-141     xml2_1.2.2
## [46] tools_3.6.1     hms_0.5.1        lifecycle_0.1.0
## [49] matrixStats_0.55.0 multcomp_1.4-10  munsell_0.5.0
## [52] compiler_3.6.1  multcompView_0.1-7 rlang_0.4.0
## [55] grid_3.6.1      rstudioapi_0.10  miniUI_0.1.1.1
## [58] labeling_0.3     rmarkdown_1.16   boot_1.3-23
## [61] DescTools_0.99.29 gtable_0.3.0     codetools_0.2-16
## [64] R6_2.4.0         zoo_1.8-6         lubridate_1.7.4
## [67] zeallot_0.1.0    nortest_1.0-4     libcoin_1.0-5
## [70] modeltools_0.2-22 stringi_1.4.3     parallel_3.6.1
## [73] Rcpp_1.0.2       vctrs_0.2.0      tidysselect_0.2.5
## [76] xfun_0.10        lmtest_0.9-37
```