

Regression to the mean modeling

Mean pain rating of 6.2 at 0.8 correlation

Peter Kamerman

Last knitted: 16 August 2019

Contents

Generate 2x2 covariance matrix	2
Mean = 6.2, SD = 1.2, Cor = 0.8	2
Generate and summarise data	2
Base data	2
Constrain values to 0-10 range	5
Model mean of V1 with increasing V1 thresholds from 0 to 5	7
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
Difference plot	18
Mean = 6.2, SD = 1.7, Cor = 0.8	20
Generate and summarise data	20
Base data	20
Constrain values to 0-10 range	22
Model mean of V1 with increasing V1 thresholds from 0 to 5	24
Model mean of V2 with increasing V1 thresholds from 0 to 5	25
Placebo response	27
threshold: 0	27
threshold: 3	29
threshold: 4	31
threshold: 5	33
Difference plot	35
Mean = 6.2, SD = 2.2, Cor = 0.8	37
Generate and summarise data	37
Base data	37
Constrain values to 0-10 range	39
Model mean of V1 with increasing V1 thresholds from 0 to 5	41
Model mean of V2 with increasing V1 thresholds from 0 to 5	42
Placebo response	44
threshold: 0	44
threshold: 3	46
threshold: 4	48
threshold: 5	50
Difference plot	52
Publication plot	54
Session information	54

Generate 2x2 covariance matrix

Generate a covariance matrix using an SD of 1.2, 1.7, and 2.2, and correlation of 0.8.

```
cor <- matrix(c(1, 0.8, 0.8, 1), ncol = 2)
std_1 <- c(1.2, 1.2)
std_2 <- c(1.7, 1.7)
std_3 <- c(2.2, 2.2)
cov_1 <- cor2cov(cor.mat = cor,
                sd = std_1)
cov_1

##          [,1] [,2]
## [1,] 1.440 1.152
## [2,] 1.152 1.440

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

##          [,1] [,2]
## [1,] 2.890 2.312
## [2,] 2.312 2.890

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

##          [,1] [,2]
## [1,] 4.840 3.872
## [2,] 3.872 4.840
```

Mean = 6.2, SD = 1.2, Cor = 0.8

Generate and summarise data

Base data

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

# Generate the data
six_1.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_1))

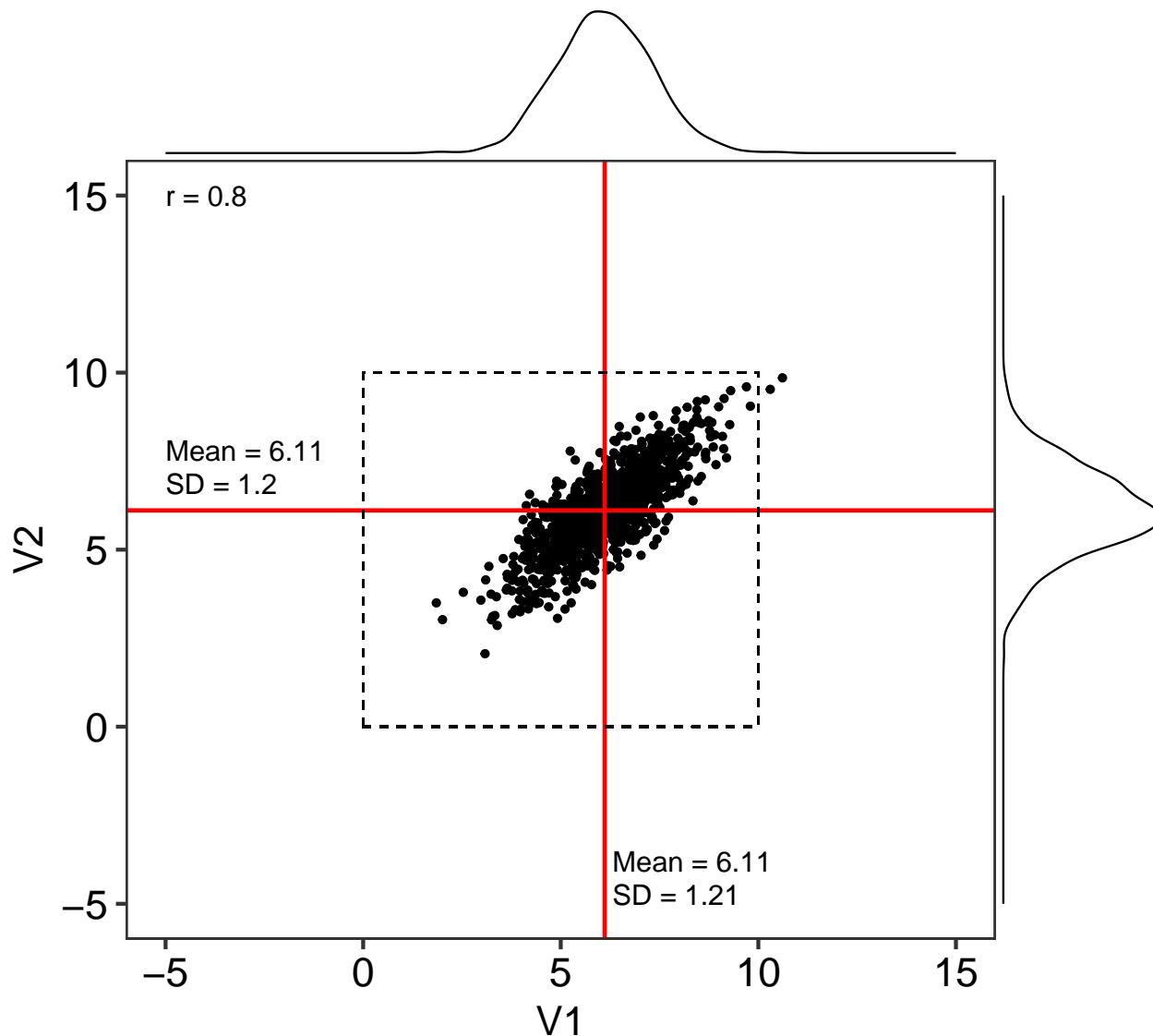
# Plot base data
ggMarginal(ggplot(data = six_1.base) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_1.base$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_1.base$V1),
    colour = 'red', size = 1) +
  geom_rect(ymin = 0, ymax = 10,
```

```

      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(six_1.base$V1, six_1.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_1.base$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_1.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_1.base$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_1.base$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_1.base$V1) + 0.2, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_1.base$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_1.base$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_1.base$V1), 2)}")) +
  labs(title = 'A: Unconstained',
    caption = 'Parameters: Mean = 6.2, SD = 1.2, Cor = 0.8') +
  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 = 6.2, SD = 1.2, Cor = 0.8

```
# Linear regression
summary(lm(V2 ~ V1, data = six_1.base))

##
## Call:
## lm(formula = V2 ~ V1, data = six_1.base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1052 -0.4681  0.0142  0.4799  2.3636
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.29254    0.11850   10.91  <2e-16 ***
```

```
## V1          0.78780    0.01901    41.43    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7271 on 998 degrees of freedom
## Multiple R-squared:  0.6323, Adjusted R-squared:  0.632
## F-statistic: 1716 on 1 and 998 DF,  p-value: < 2.2e-16
```

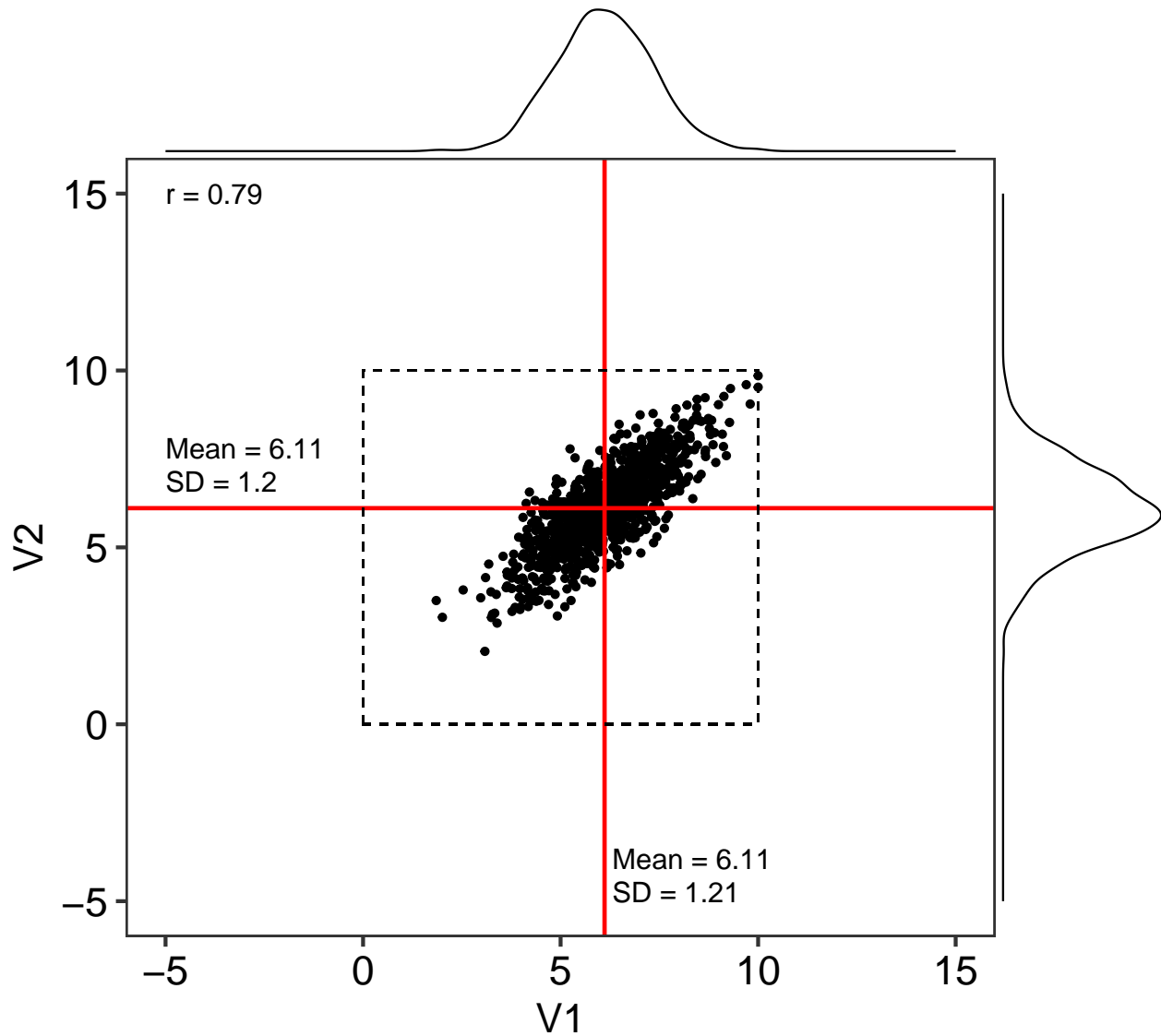
Constrain values to 0-10 range

```
# Process data
six_1 <- six_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 = six_1) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_1$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_1$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(six_1$V1, six_1$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_1$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_1$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_1$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_1$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_1$V1) + 0.2, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_1$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_1$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_1$V1), 2)}")) +
  labs(title = 'B: Constrained (0-10 range)',
    caption = 'Parameters: Mean = 6.2, SD = 1.2, Cor = 0.8') +
  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 = 6.2, SD = 1.2, Cor = 0.8

```
# Linear regression
summary(lm(V2 ~ V1, data = six_1))

##
## Call:
## lm(formula = V2 ~ V1, data = six_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10376 -0.47056  0.01237  0.48060  2.36441
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 1.28218 0.11884 10.79 <2e-16 ***
## V1          0.78961 0.01907 41.40 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7274 on 998 degrees of freedom
## Multiple R-squared:  0.632, Adjusted R-squared:  0.6316
## F-statistic: 1714 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
six_1V1 <- six_1$V1

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

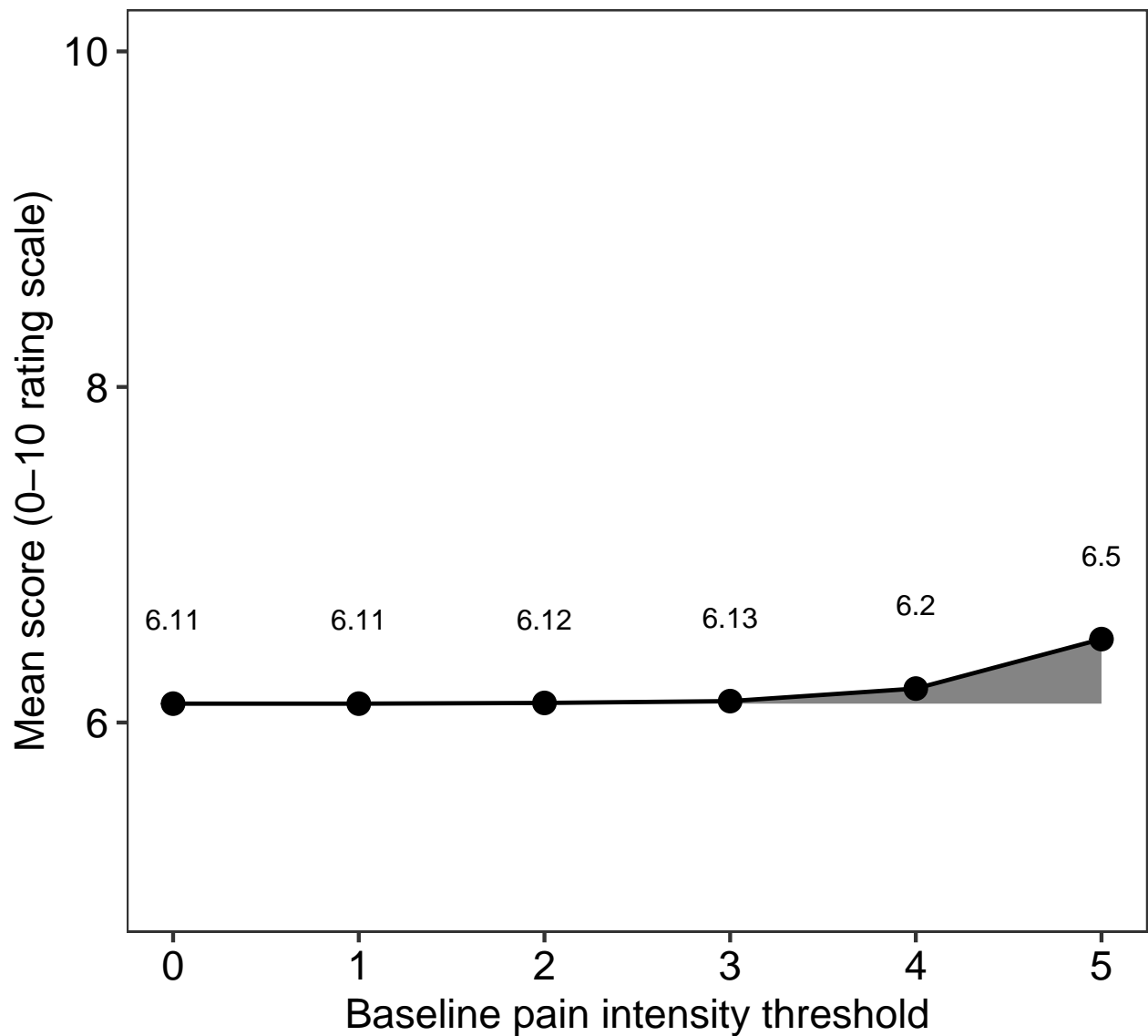
# Generate a vector of V1 means at each V1 threshold
six_1V1.shift <- sapply(cutoff, function(x){mean(six_1V1[six_1V1 > x])})

# Calculate deviation
(six_1V1.df <- data.frame(cutoff = cutoff,
                          mean = six_1V1.shift) %>%
  mutate(deviation = mean - mean(six_1V1)))

##   cutoff    mean deviation
## 1      0 6.112437 0.000000000
## 2      1 6.112437 0.000000000
## 3      2 6.116702 0.004264682
## 4      3 6.127575 0.015138609
## 5      4 6.202336 0.089898860
## 6      5 6.497302 0.384865053

# Plot data
ggplot(data = six_1V1.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 1.2, Cor = 0.8',
       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 = 6.2, SD = 1.2, Cor = 0.8

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

```
# Extract visit 2 data
six_1V2 <- six_1$V2

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

# Generate a vector of V2 means at each V1 threshold
six_1V2.shift <- map_dbl(.x = cutoff,
  ~ six_1 %>%
    filter(V1 > .x) %>%
    .$V2 %>%
```



```

mean(.))

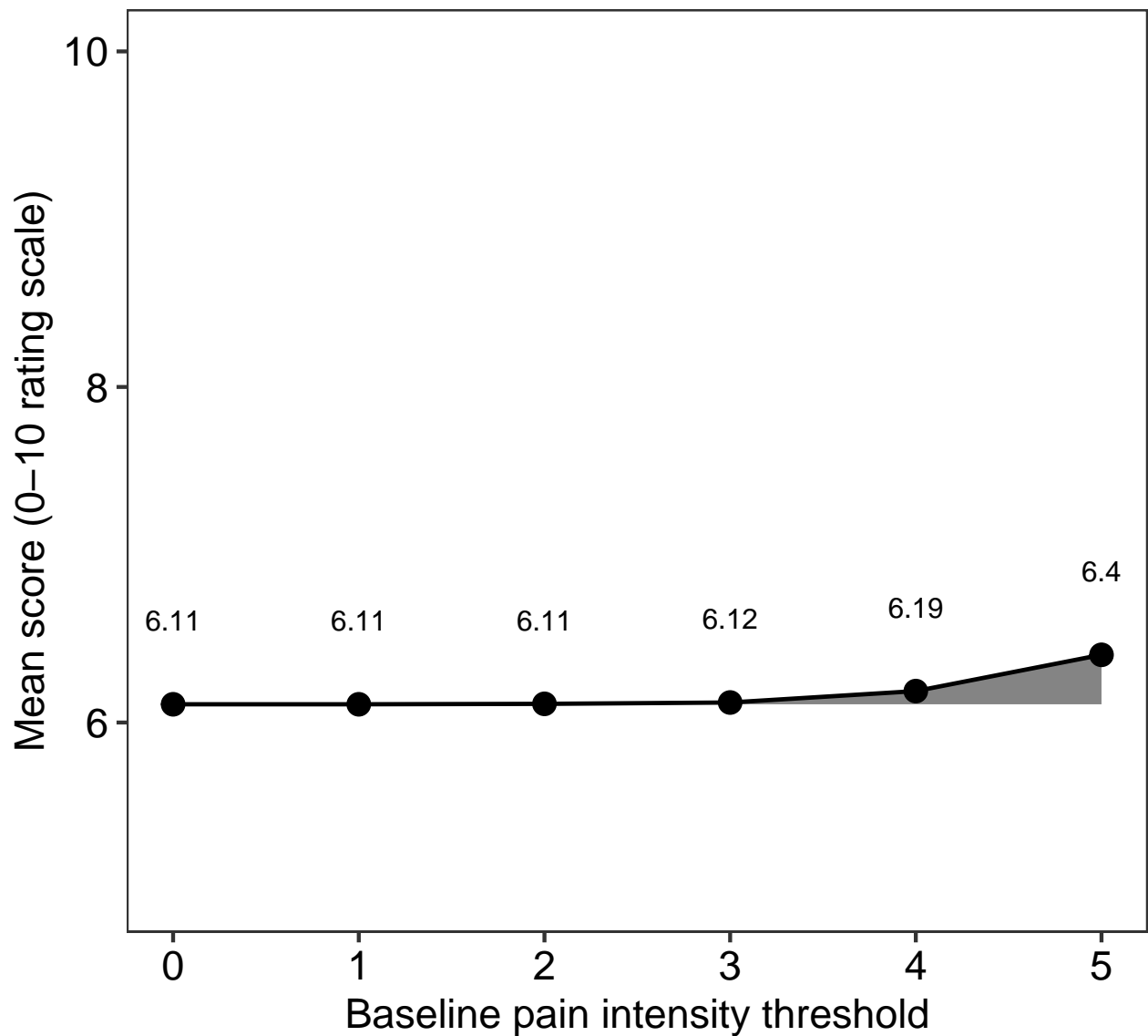
# Calculate deviation
(six_1V2.df <- data.frame(cutoff = cutoff,
                          mean = six_1V2.shift) %>%
  mutate(deviation = mean - mean(six_1V2)))

##   cutoff    mean  deviation
## 1      0 6.108642 0.000000000
## 2      1 6.108642 0.000000000
## 3      2 6.111256 0.002613545
## 4      3 6.119226 0.010583871
## 5      4 6.187290 0.078648120
## 6      5 6.403452 0.294810070

# Plot data
ggplot(data = six_1V2.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 1.2, Cor = 0.8',
       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 = 6.2, SD = 1.2, Cor = 0.8

Placebo response

threshold: 0

```
# Process data
placebo_1.0 <- six_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)
```

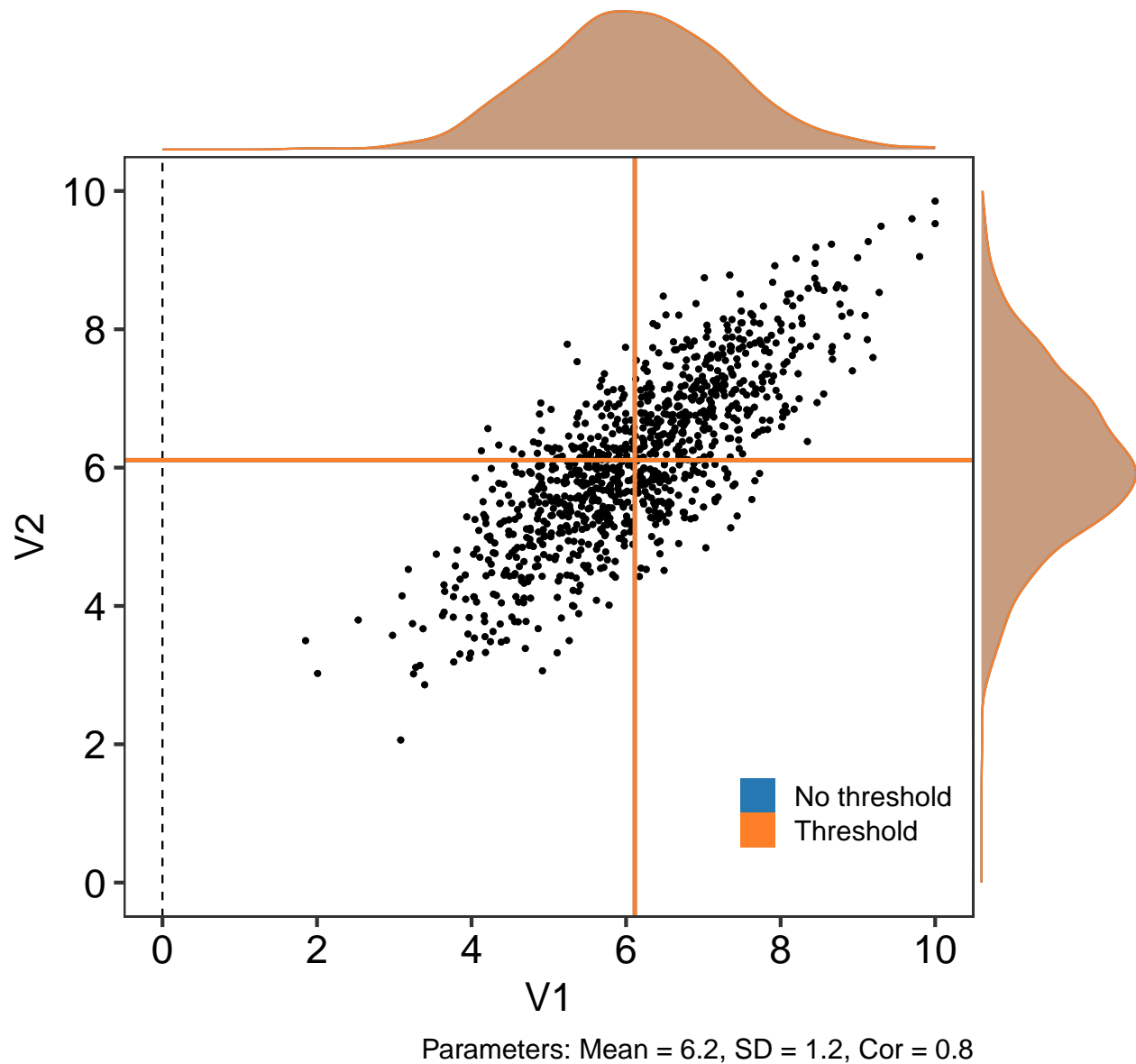
.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.00379	0.95	-0.046	0.0509

```
# Plot the data
```

```
ggMarginal(placebo_1.0[, 1:3] %>%
  bind_rows(six_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 = six_1,
            colour = '#999999',
            size = 1) +
  geom_point(data = placebo_1.0,
            size = 1,
            colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.2, Cor = 0.8') +
  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 <- six_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	996	0.00835	0.95	-0.0382	0.0552

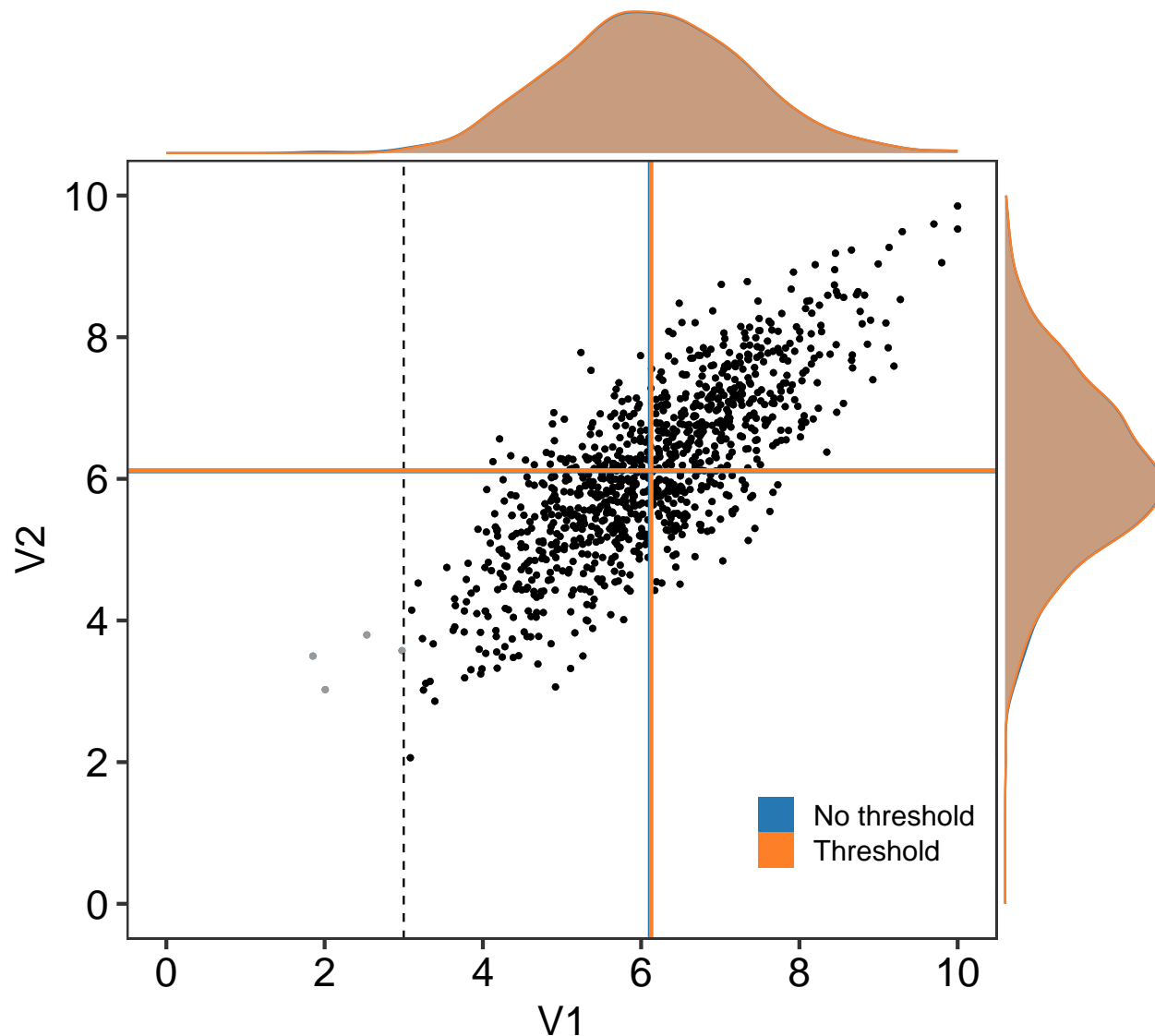
```

# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
  bind_rows(six_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 = six_1,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_1.3,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.2, Cor = 0.8') +
  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 = 6.2, SD = 1.2, Cor = 0.8

threshold: 4

```
# Process that data
placebo_1.4 <- six_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	967	0.015	0.95	-0.0331	0.0614

```
# Plot the data
```

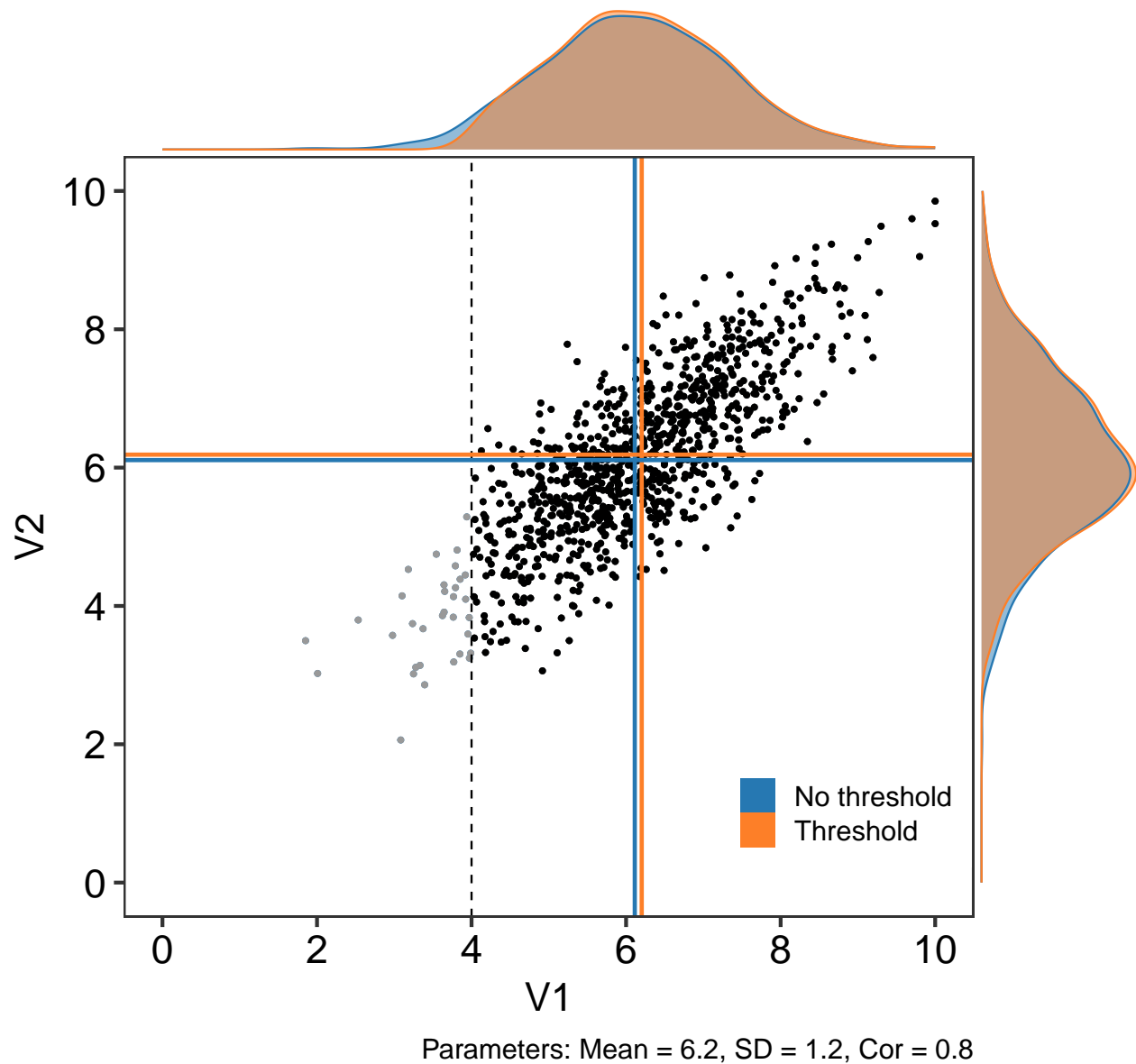
```

ggMarginal(placebo_1.4[, 1:3] %>%
  bind_rows(six_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 = six_1,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_1.4,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.2, Cor = 0.8') +
  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 <- six_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	819	0.0938	0.95	0.0421	0.146

```
# Plot the data
```

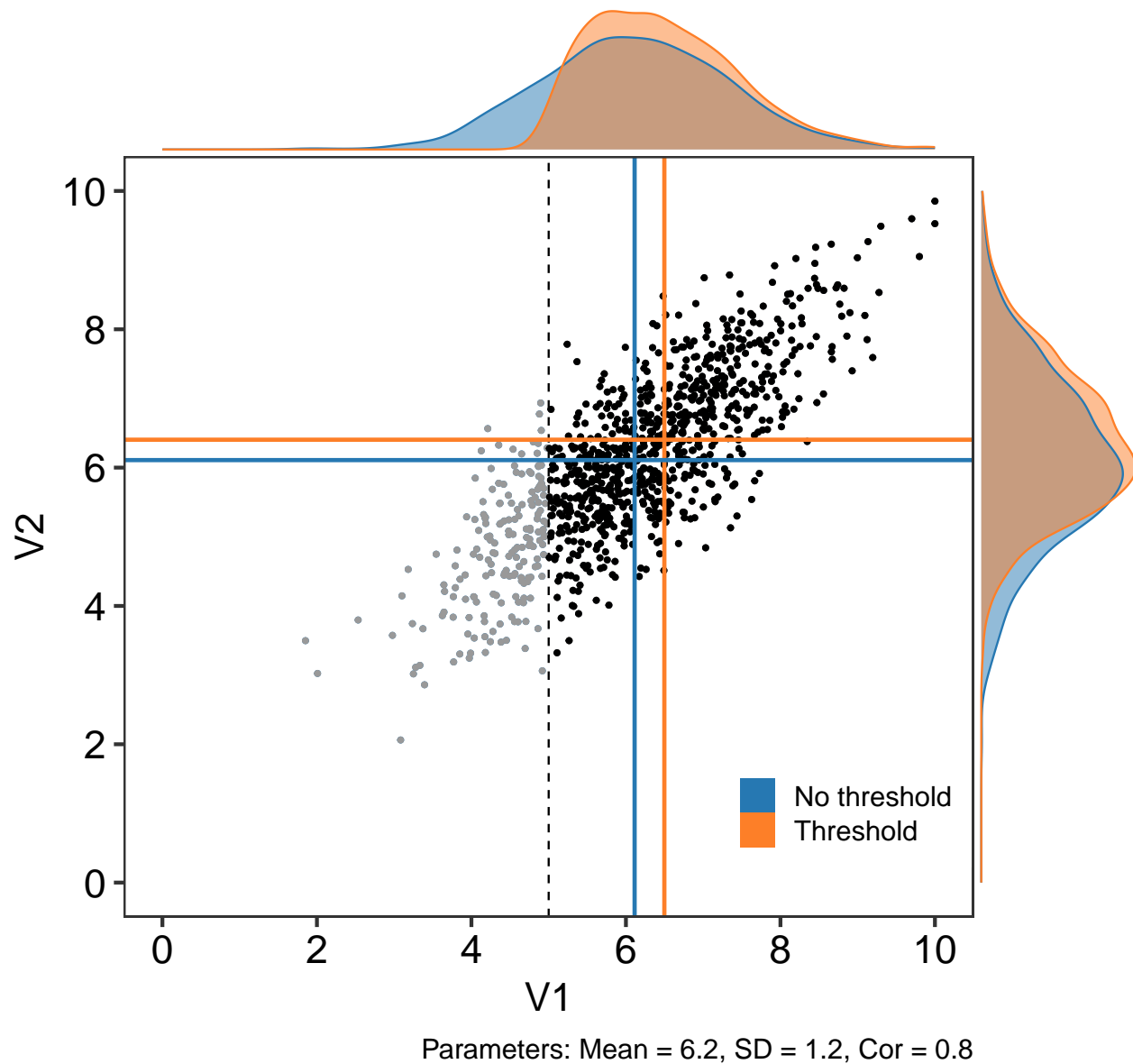
```

ggMarginal(placebo_1.5[, 1:3] %>%
  bind_rows(six_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 = six_1,
            colour = '#999999',
            size = 1) +
  geom_point(data = placebo_1.5,
            size = 1,
            colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.2, Cor = 0.8') +
  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



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) +
```

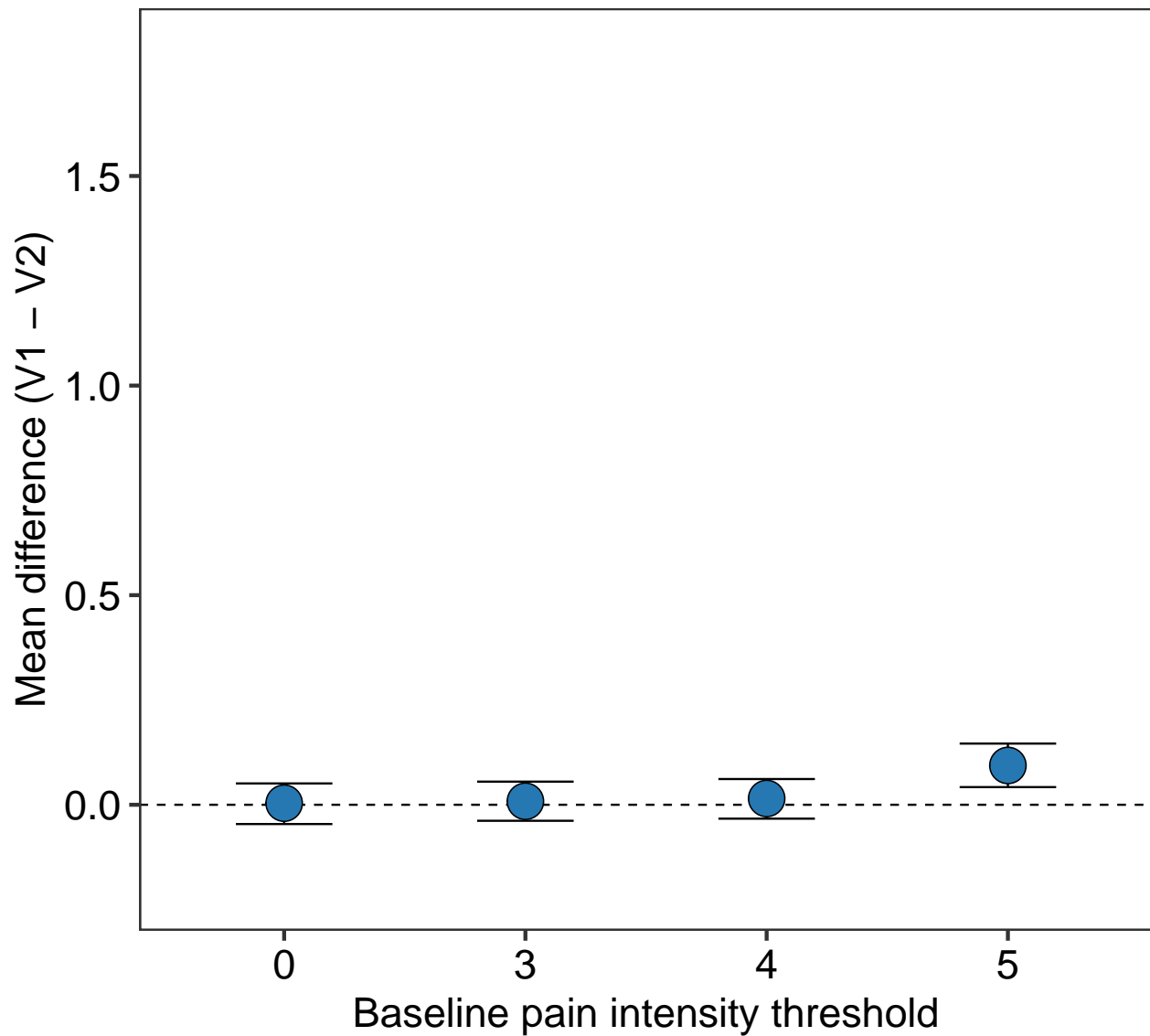
```

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 = 6.2, SD = 1.2, Cor = 0.8',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)' +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_1

```

A

Parameters: Mean = 6.2, SD = 1.2, Cor = 0.8



Mean = 6.2, SD = 1.7, Cor = 0.8

Generate and summarise data

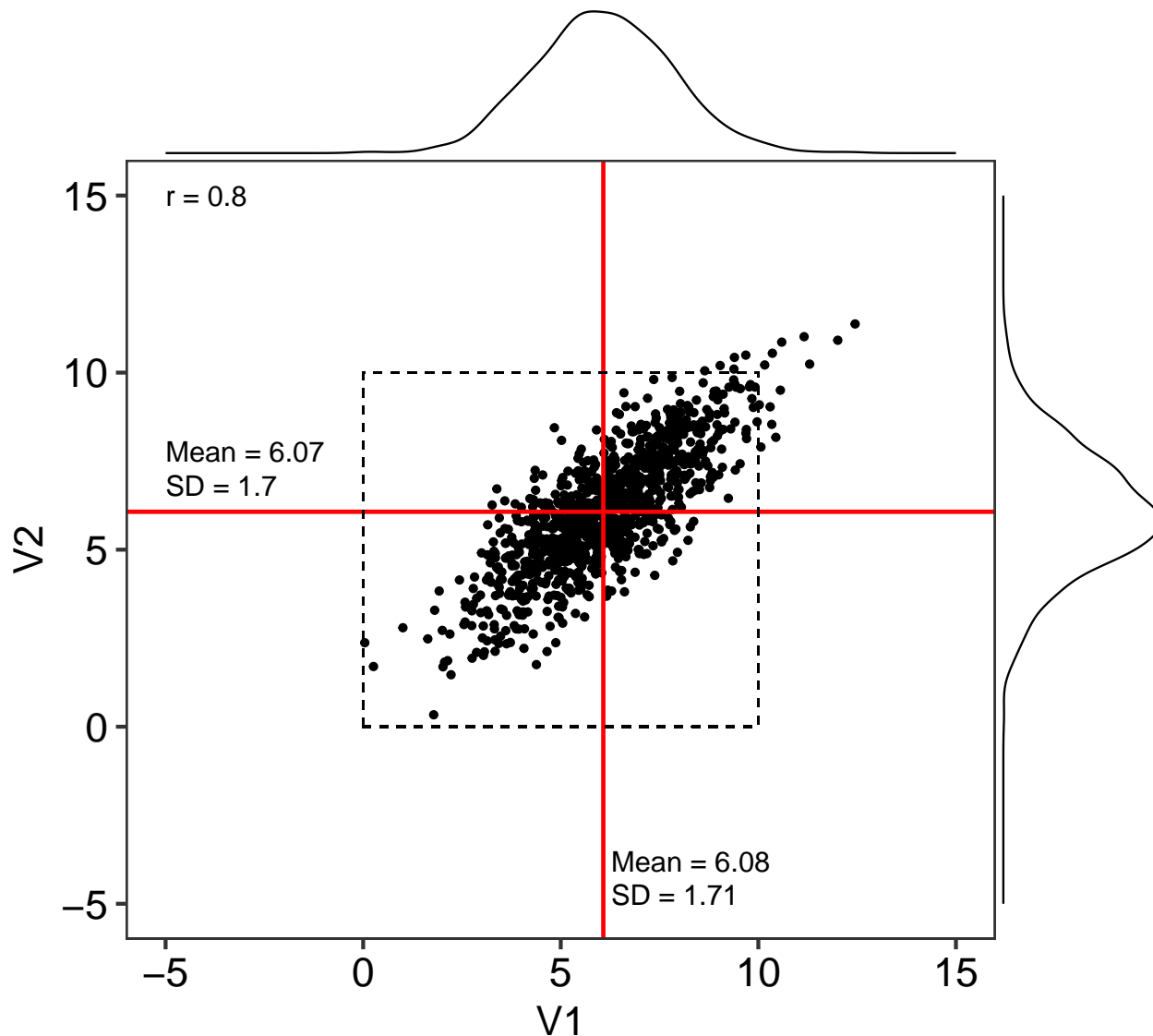
Base data

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

# Generate the data
six_2.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_2))

# Plot base data
ggMarginal(ggplot(data = six_2.base) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_2.base$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_2.base$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(six_2.base$V1, six_2.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_2.base$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_2.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_2.base$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_2.base$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_2.base$V1) + 0.2, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_2.base$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_2.base$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_2.base$V1), 2)}")) +
  labs(title = 'A: Unconstrained',
    caption = 'Parameters: Mean = 6.2, SD = 1.7, Cor = 0.8') +
  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 = 6.2, SD = 1.7, Cor = 0.8

```
# Linear regression
summary(lm(V2 ~ V1, data = six_2.base))

##
## Call:
## lm(formula = V2 ~ V1, data = six_2.base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9824 -0.6632  0.0201  0.6799  3.3484
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.28292    0.12006   10.69  <2e-16 ***
```

```
## V1          0.78780    0.01901    41.43    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.03 on 998 degrees of freedom
## Multiple R-squared:  0.6323, Adjusted R-squared:  0.632
## F-statistic: 1716 on 1 and 998 DF,  p-value: < 2.2e-16
```

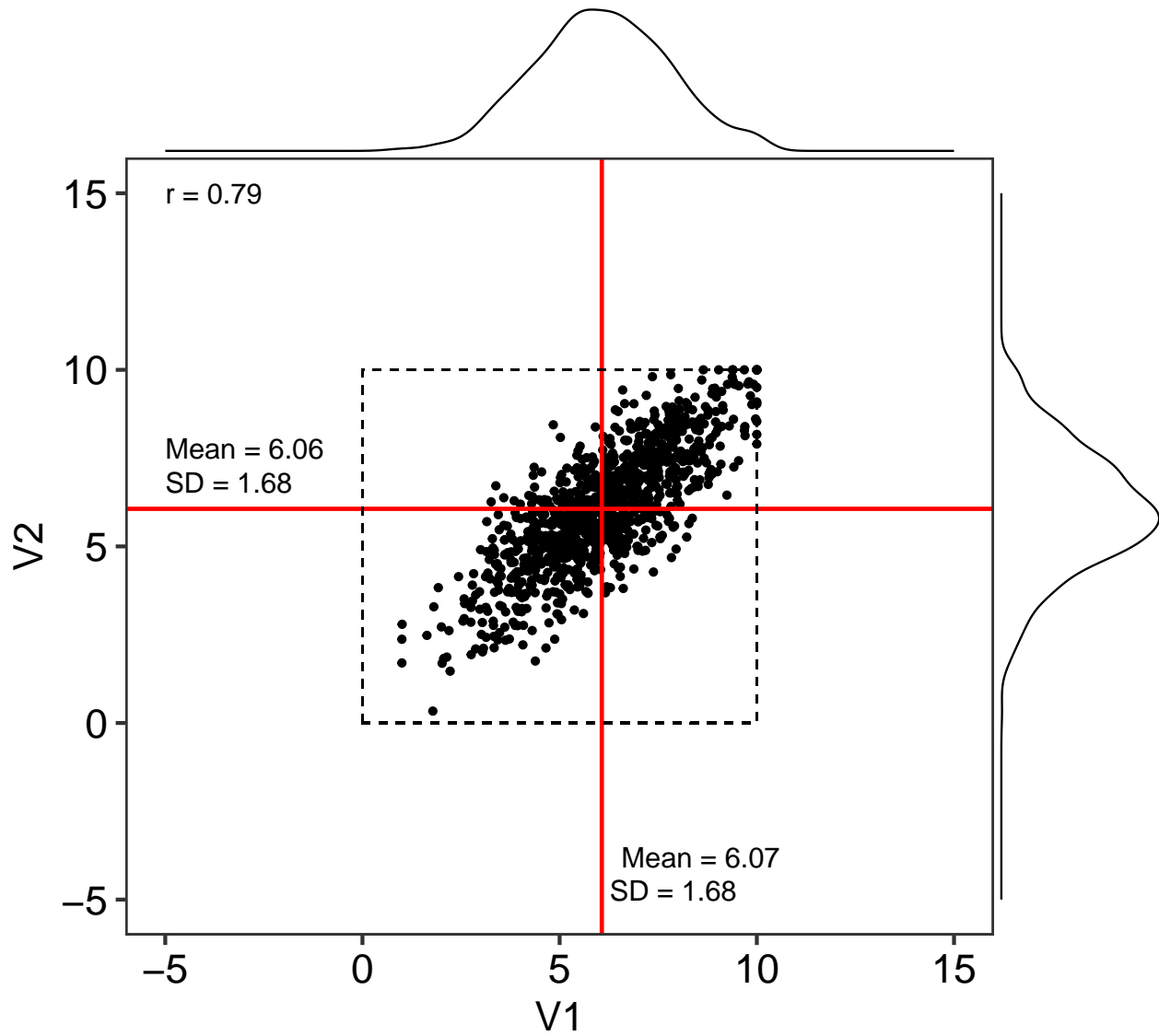
Constrain values to 0-10 range

```
# Process data
six_2 <- six_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 = six_2) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_2$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_2$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(six_2$V1, six_2$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_2$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_2$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_2$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_2$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_2$V1) + 0.5, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_2$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_2$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_2$V1), 2)}")) +
  labs(title = 'B: Constrained (0-10 range)',
    caption = 'Parameters: Mean = 6.2, SD = 1.7, Cor = 0.8') +
  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 = 6.2, SD = 1.7, Cor = 0.8

```
# Linear regression
summary(lm(V2 ~ V1, data = six_2))

##
## Call:
## lm(formula = V2 ~ V1, data = six_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9756 -0.6595  0.0179  0.6843  3.3534
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 1.25863 0.12171 10.34 <2e-16 ***
## V1          0.79179 0.01933 40.97 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.027 on 998 degrees of freedom
## Multiple R-squared:  0.6271, Adjusted R-squared:  0.6267
## F-statistic: 1678 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
six_2V1 <- six_2$V1

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

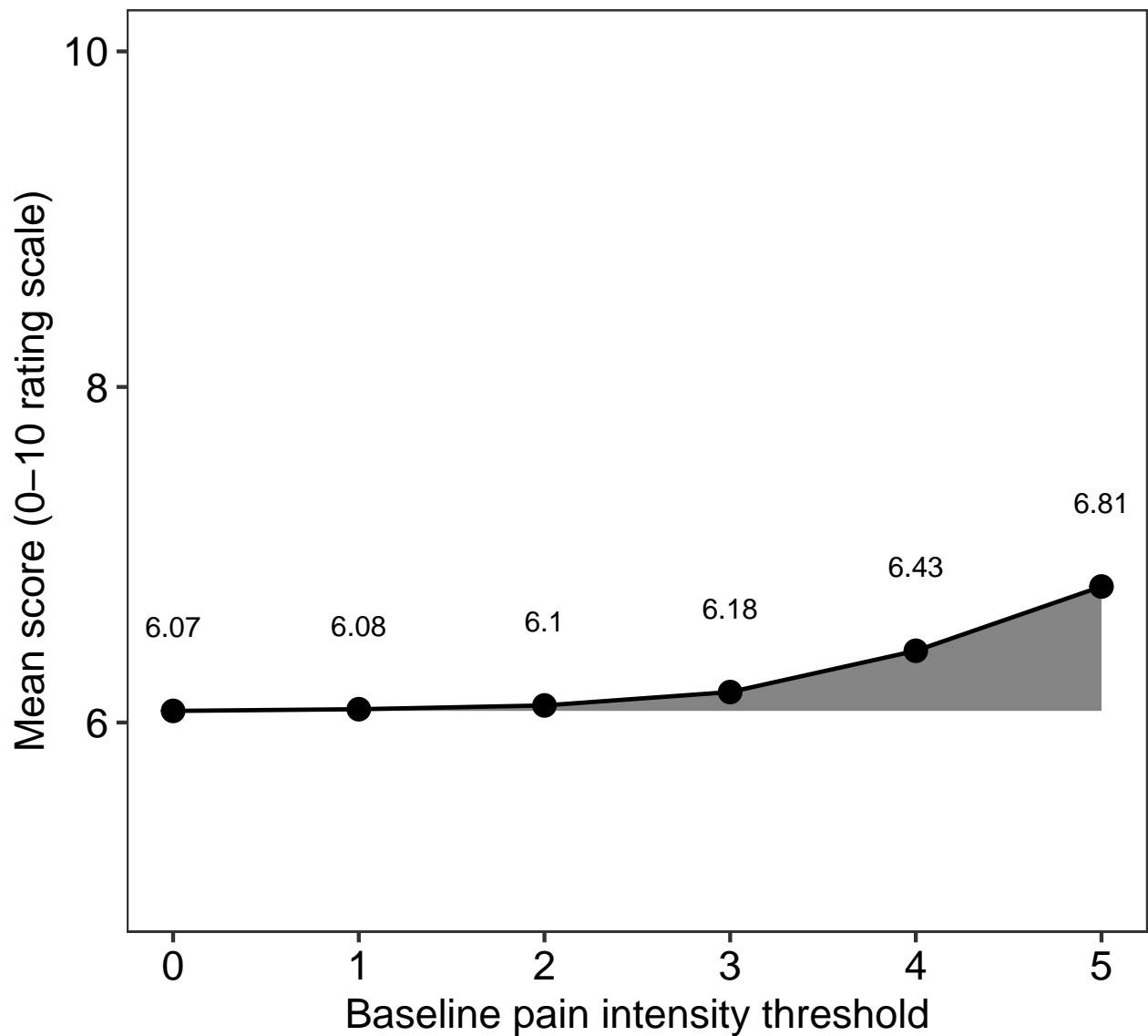
# Generate a vector of V1 means at each V1 threshold
six_2V1.shift <- sapply(cutoff, function(x){mean(six_2V1[six_2V1 > x])})

# Calculate deviation
(six_2V1.df <- data.frame(cutoff = cutoff,
                          mean = six_2V1.shift) %>%
  mutate(deviation = mean - mean(six_2V1)))

##   cutoff    mean deviation
## 1      0 6.069186 0.00000000
## 2      1 6.079345 0.01015869
## 3      2 6.101738 0.03255142
## 4      3 6.181574 0.11238789
## 5      4 6.427567 0.35838073
## 6      5 6.810760 0.74157394

# Plot data
ggplot(data = six_2V1.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 1.7, Cor = 0.8',
       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 = 6.2, SD = 1.7, Cor = 0.8

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

```
# Extract visit 2 data
six_2V2 <- six_2$V2

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

# Generate a vector of V2 means at each V1 threshold
six_2V2.shift <- map_dbl(.x = cutoff,
  ~ six_2 %>%
    filter(V1 > .x) %>%
    .$V2 %>%
```

```

mean(.))

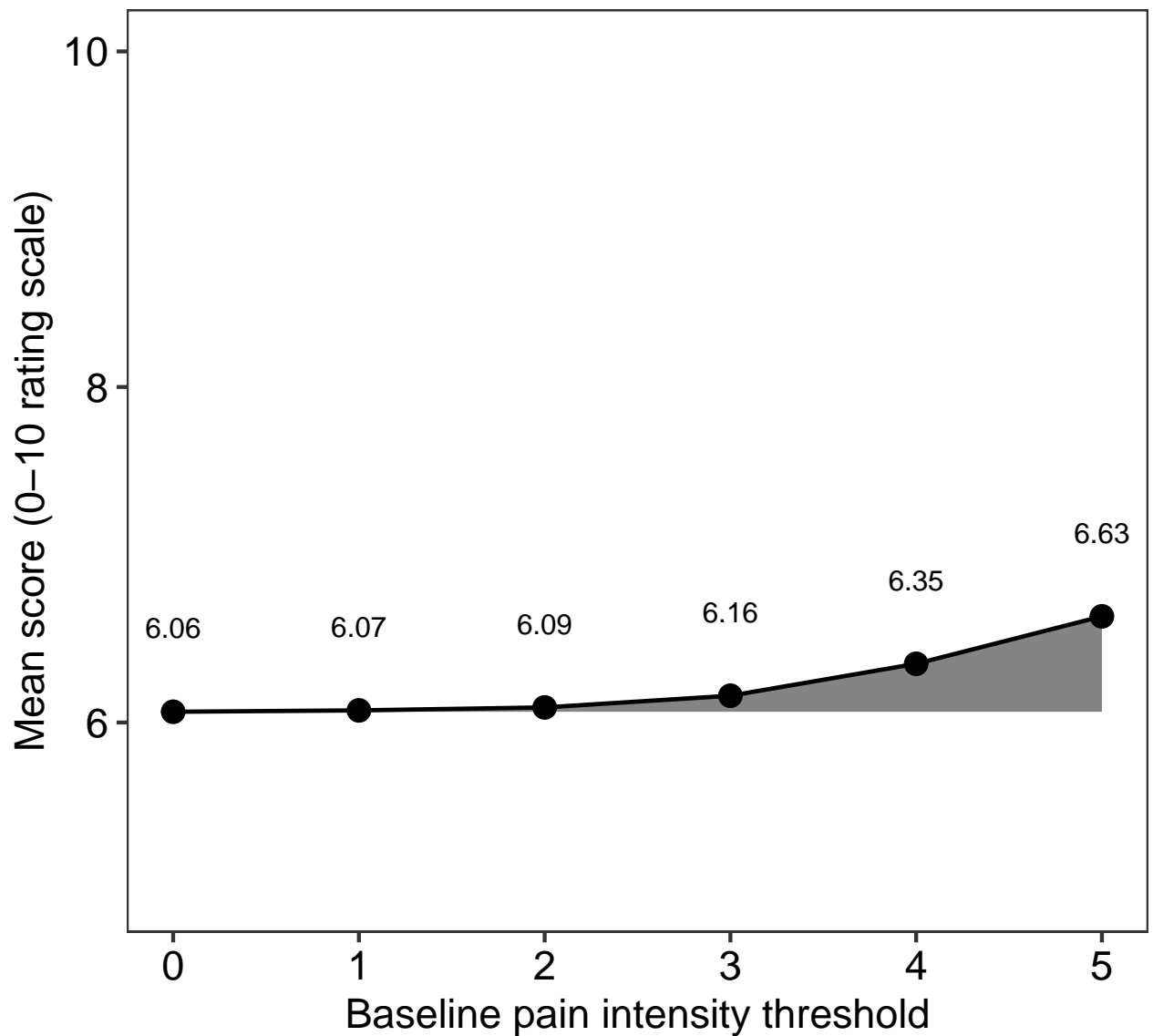
# Calculate deviation
(six_2V2.df <- data.frame(cutoff = cutoff,
                          mean = six_2V2.shift) %>%
  mutate(deviation = mean - mean(six_2V2)))

##   cutoff    mean  deviation
## 1      0 6.064135 0.000000000
## 2      1 6.072208 0.008072147
## 3      2 6.089957 0.025821792
## 4      3 6.159661 0.095525555
## 5      4 6.350046 0.285910207
## 6      5 6.632831 0.568695577

# Plot data
ggplot(data = six_2V2.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 1.7, Cor = 0.8',
       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 = 6.2, SD = 1.7, Cor = 0.8

Placebo response

threshold: 0

```
# Process data
placebo_2.0 <- six_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)
```

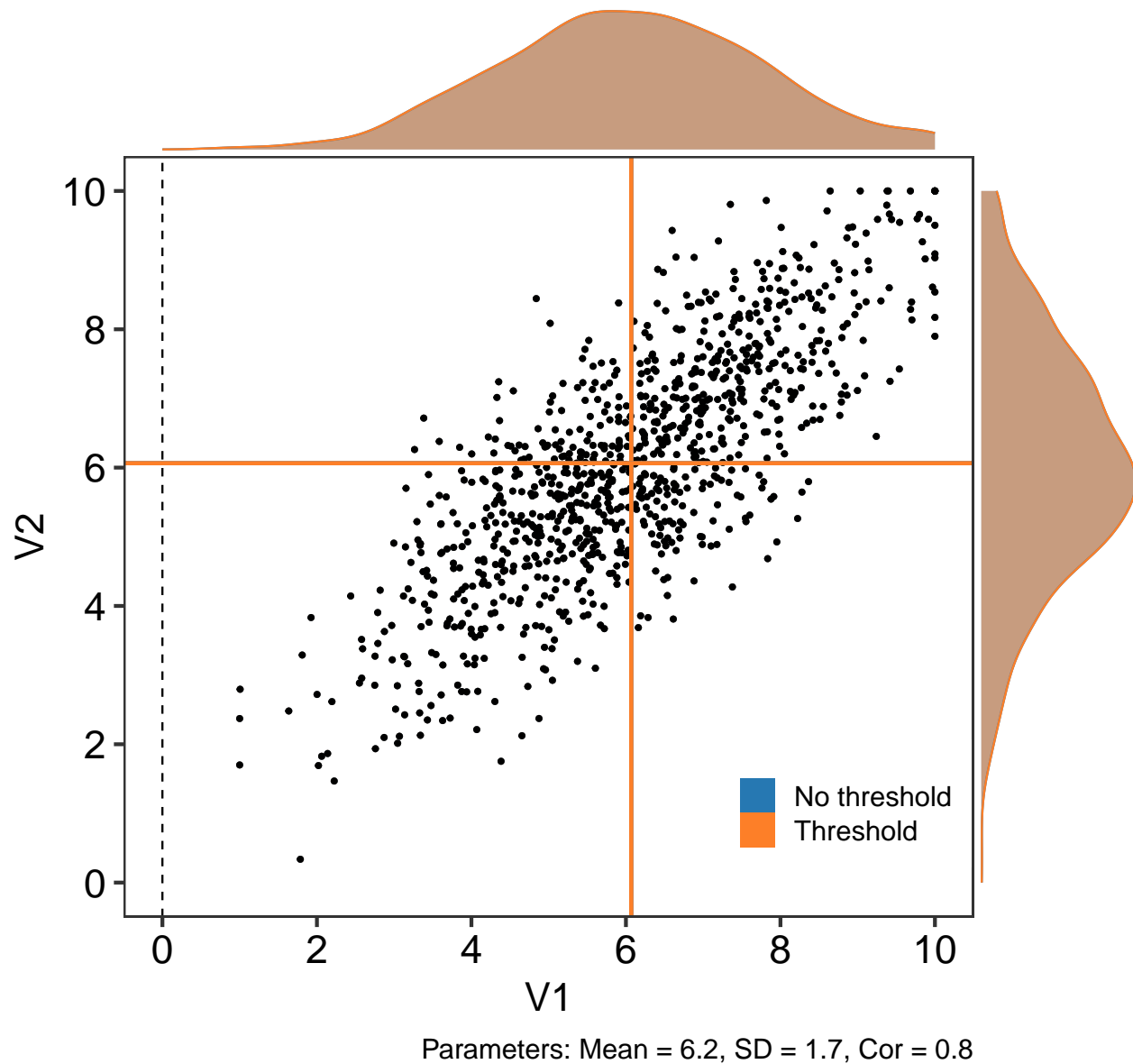
.id	n	Mean	Conf.level	Bca.lower	Bca.upper
0	1000	0.00505	0.95	-0.0641	0.0714

```
# Plot the data
```

```
ggMarginal(placebo_2.0[, 1:3] %>%
  bind_rows(six_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 = six_2,
            colour = '#999999',
            size = 1) +
  geom_point(data = placebo_2.0,
            size = 1,
            colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.7, Cor = 0.8') +
  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 <- six_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	971	0.0219	0.95	-0.0471	0.0894

```
# Plot the data
```

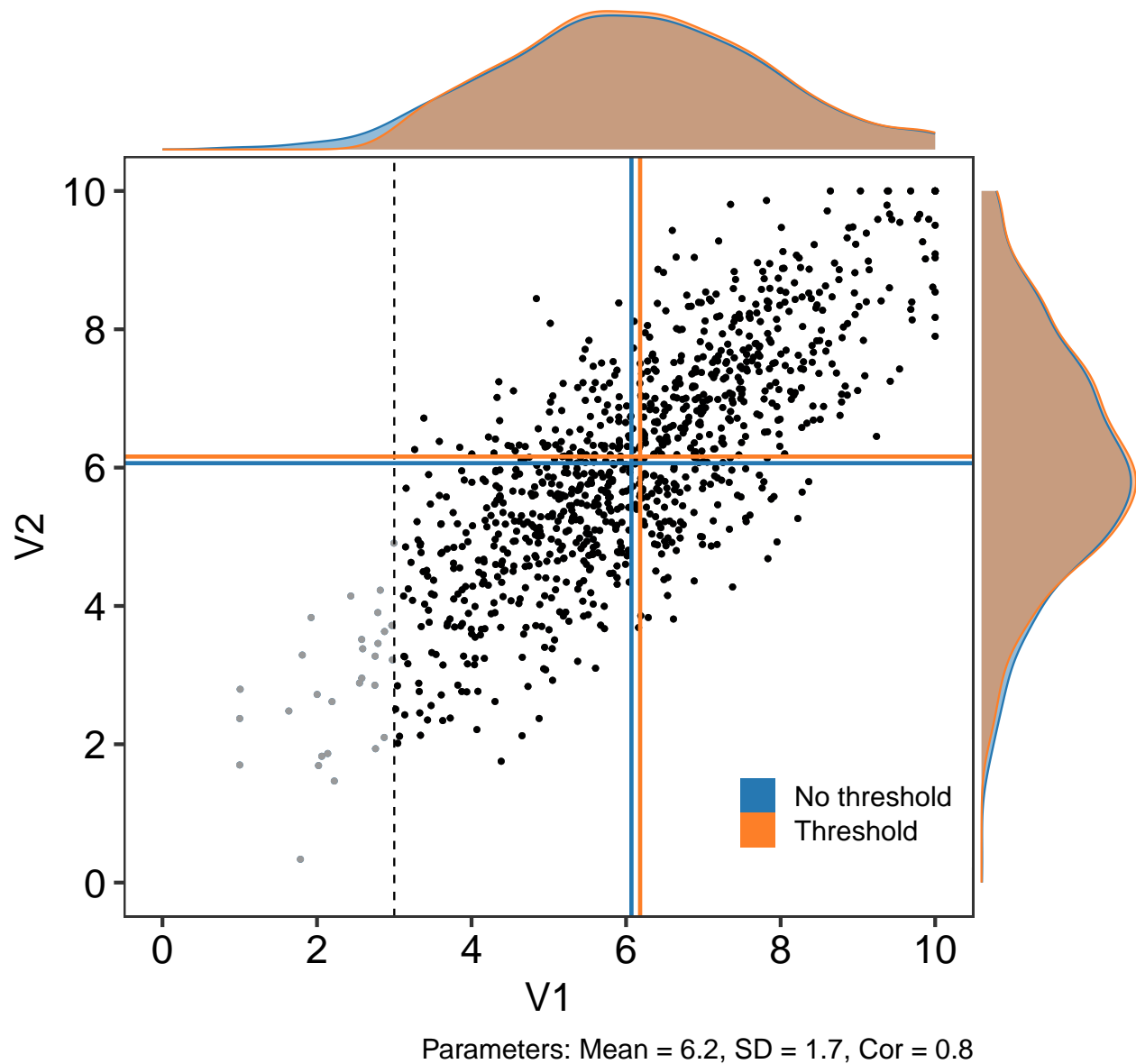
```

ggMarginal(placebo_2.3[, 1:3] %>%
  bind_rows(six_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 = six_2,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_2.3,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.7, Cor = 0.8') +
  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 <- six_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	888	0.0775	0.95	0.00769	0.149

```
# Plot the data
```

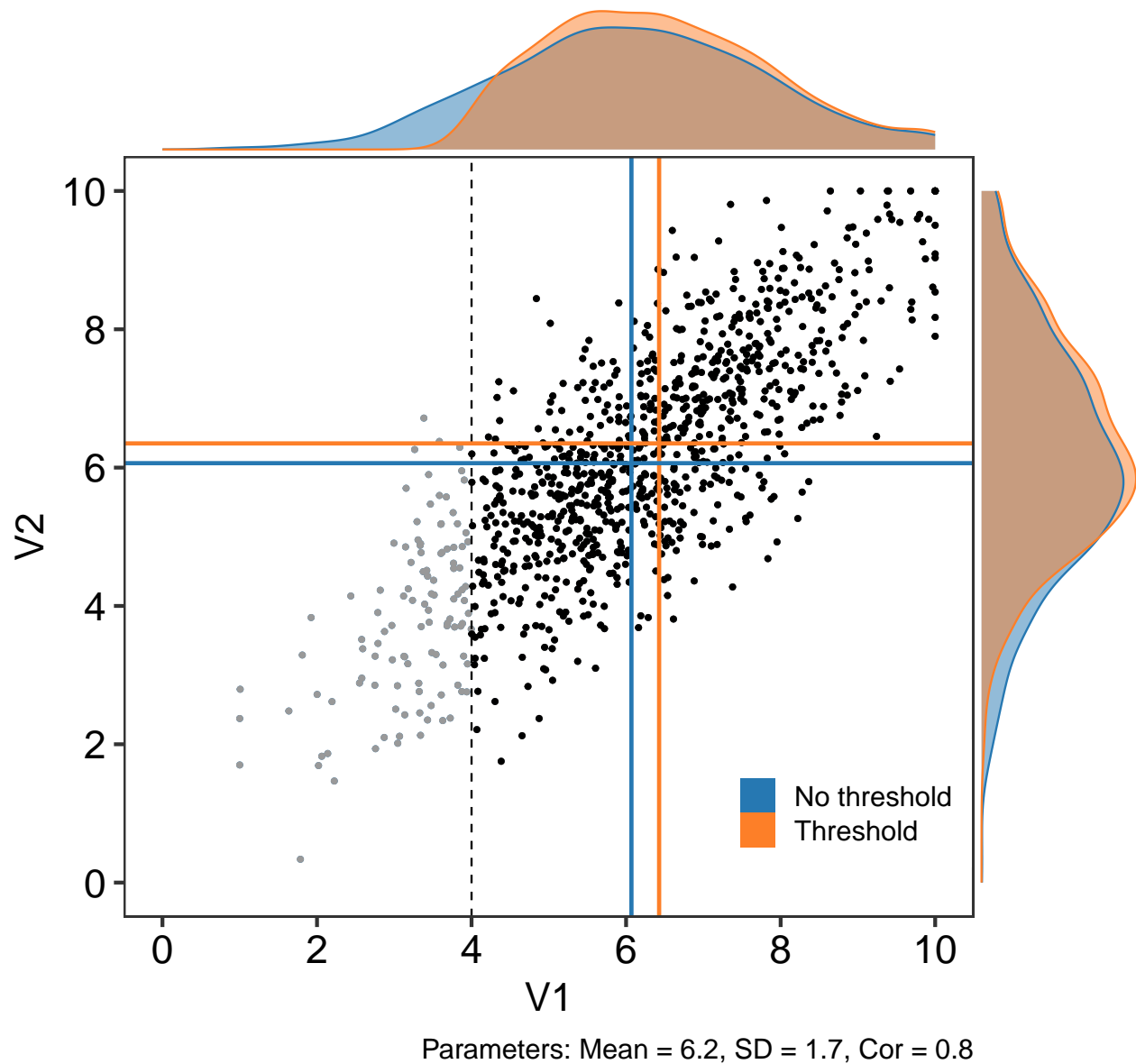
```

ggMarginal(placebo_2.4[, 1:3] %>%
  bind_rows(six_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 = six_2,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_2.4,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.7, Cor = 0.8') +
  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 <- six_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	739	0.178	0.95	0.103	0.253

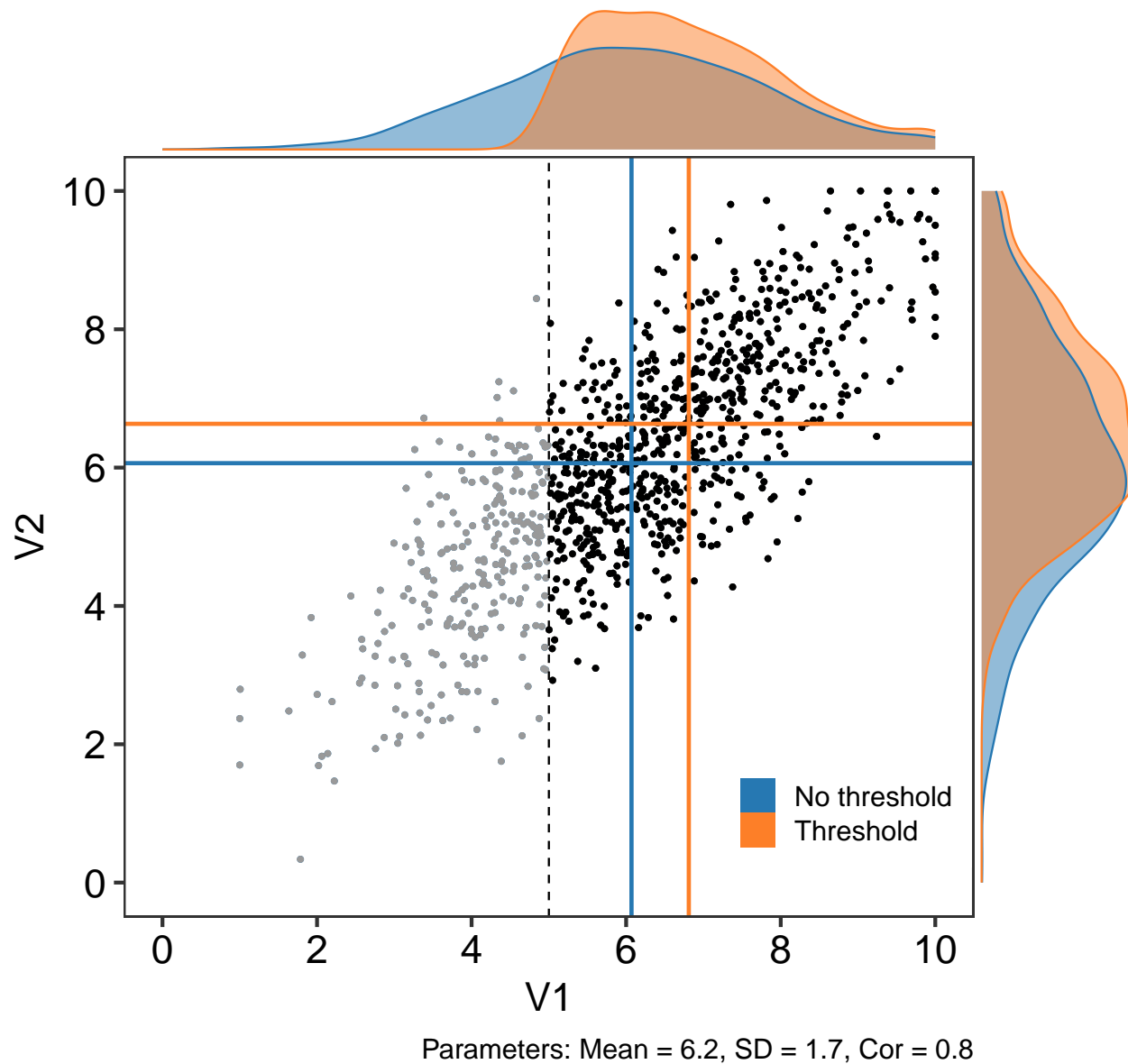
```

# Plot the data
ggMarginal(placebo_2.5[, 1:3] %>%
  bind_rows(six_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 = six_2,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_2.5,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 1.7, Cor = 0.8') +
  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



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) +
```

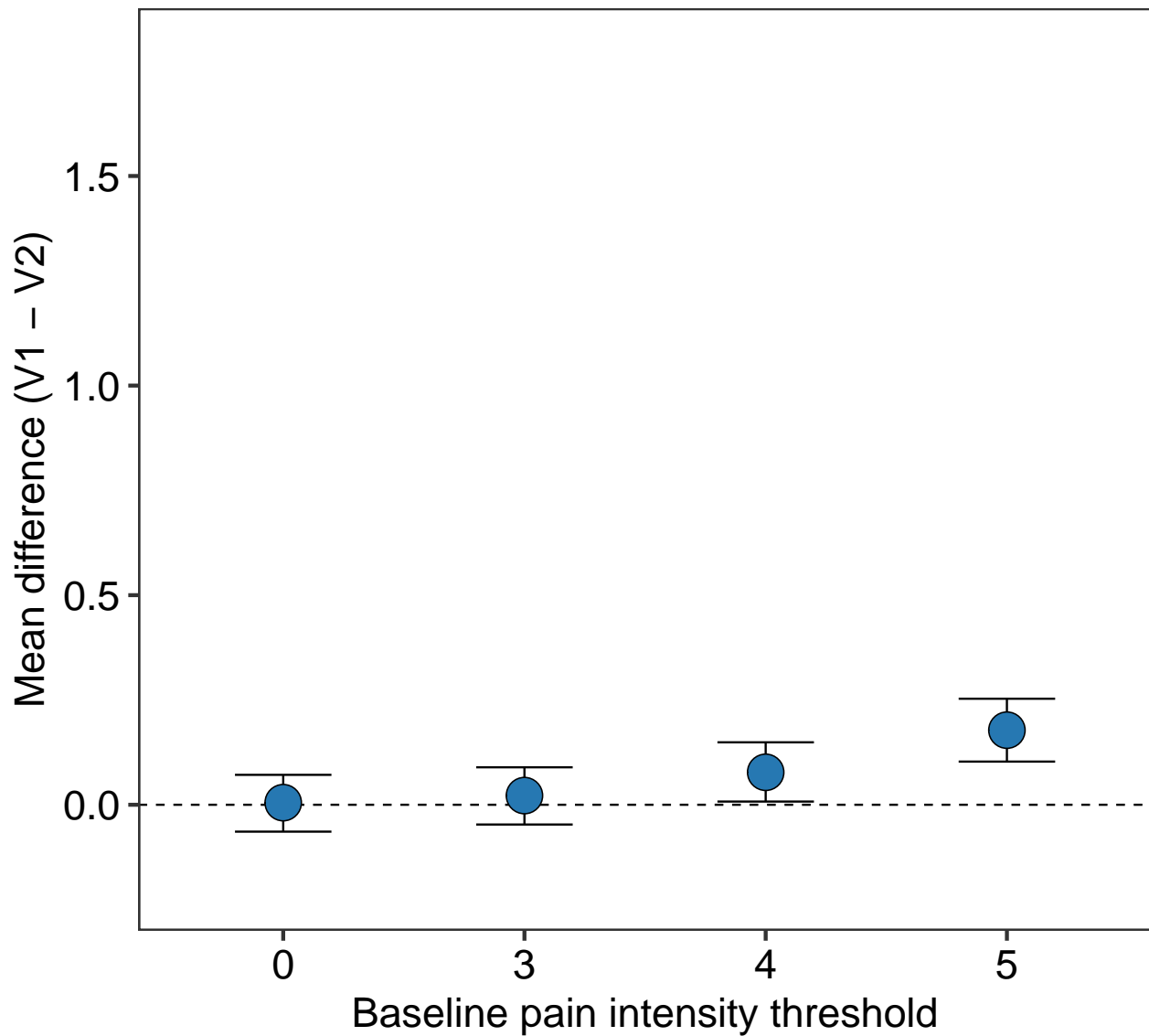
```

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 = 6.2, SD = 1.7, Cor = 0.8',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)' +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_2

```

B

Parameters: Mean = 6.2, SD = 1.7, Cor = 0.8



Mean = 6.2, SD = 2.2, Cor = 0.8

Generate and summarise data

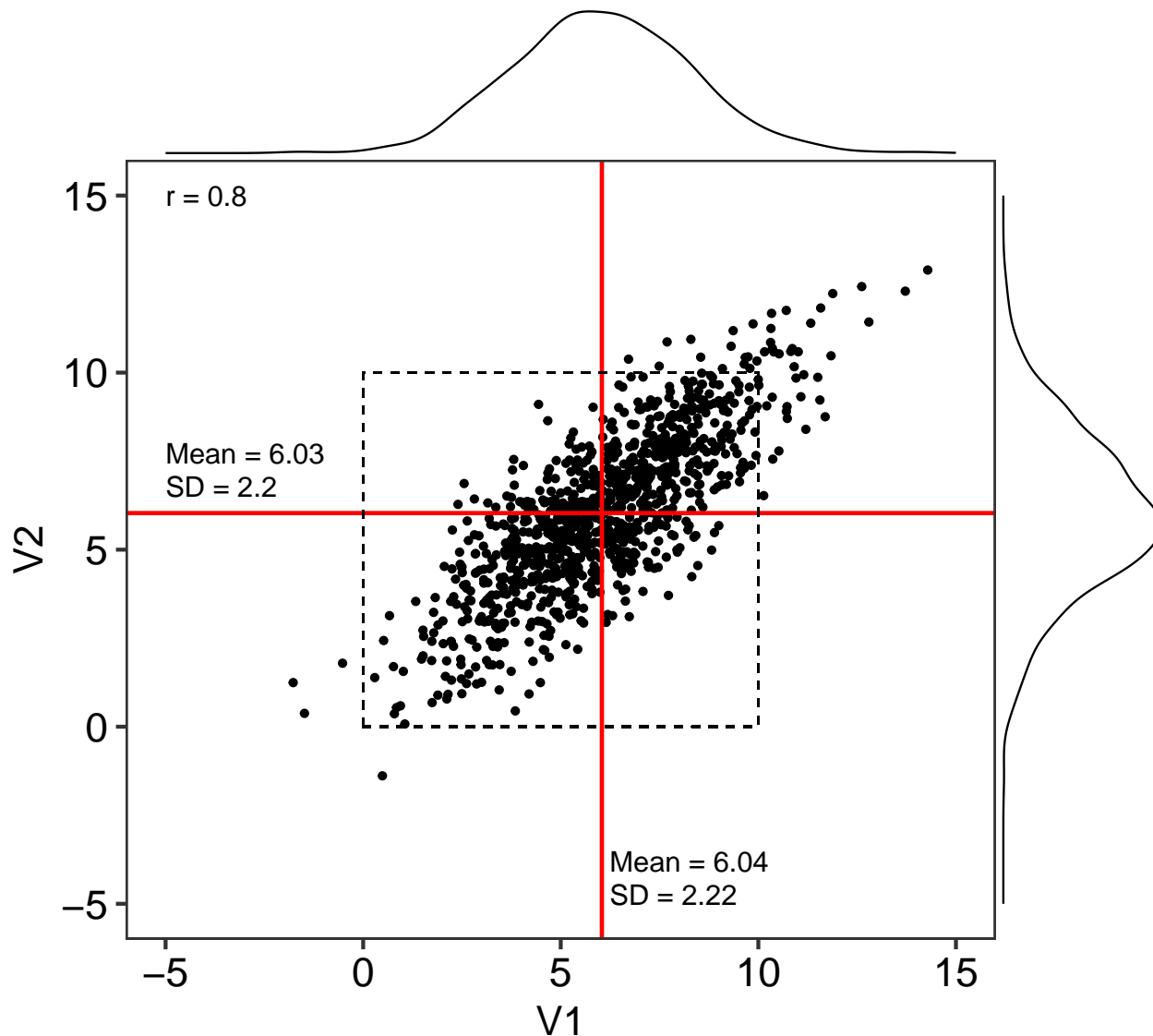
Base data

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

# Generate the data
six_3.base <- as.data.frame(mvrnorm(n = 1000, mu = c(6.2, 6.2), Sigma = cov_3))

# Plot base data
ggMarginal(ggplot(data = six_3.base) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_3.base$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_3.base$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(six_3.base$V1, six_3.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_3.base$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_3.base$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_3.base$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_3.base$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_3.base$V1) + 0.2, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_3.base$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_3.base$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_3.base$V1), 2)}")) +
  labs(title = 'A: Unconstrained',
    caption = 'Parameters: Mean = 6.2, SD = 2.2, Cor = 0.8') +
  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 = 6.2, SD = 2.2, Cor = 0.8

```
# Linear regression
summary(lm(V2 ~ V1, data = six_3.base))

##
## Call:
## lm(formula = V2 ~ V1, data = six_3.base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.8595 -0.8582  0.0260  0.8799  4.3332
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.27330    0.12236   10.41  <2e-16 ***
```

```
## V1          0.78780    0.01901    41.43    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.333 on 998 degrees of freedom
## Multiple R-squared:  0.6323, Adjusted R-squared:  0.632
## F-statistic: 1716 on 1 and 998 DF,  p-value: < 2.2e-16
```

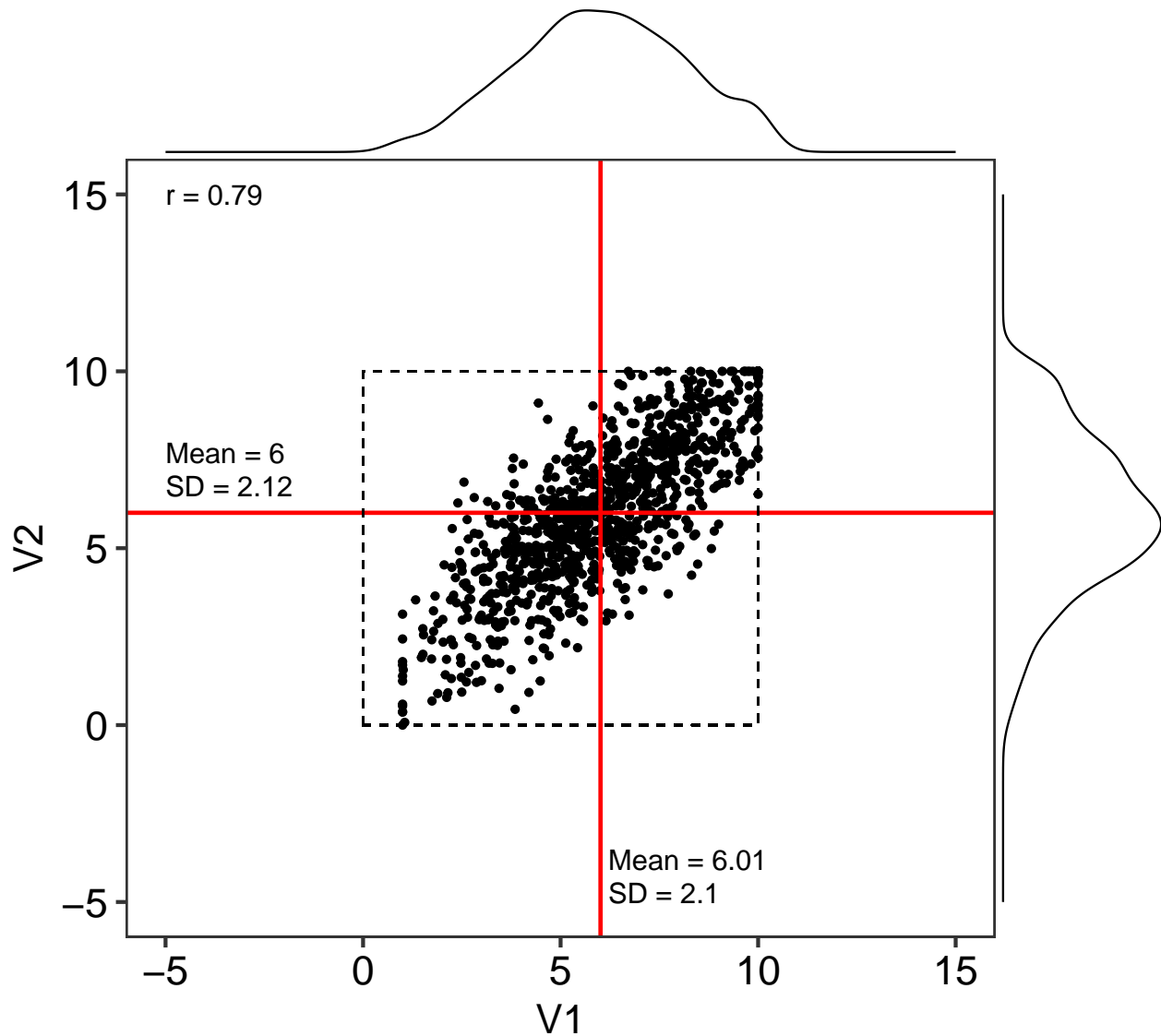
Constrain values to 0-10 range

```
# Process data
six_3 <- six_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 = six_3) +
  aes(x = V1, y = V2) +
  geom_point() +
  geom_hline(yintercept = mean(six_3$V2),
    colour = 'red', size = 1) +
  geom_vline(xintercept = mean(six_3$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(six_3$V1, six_3$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_3$V2) + 1.7,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_3$V2), 2)}")) +
  annotate(geom = 'text', x = -5, y = mean(six_3$V2) + 0.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_3$V2), 2)}")) +
  annotate(geom = 'text', x = mean(six_3$V1) + 0.2, y = -3.8,
    hjust = 0, size = 5,
    label = str_glue("Mean = {round(mean(six_3$V1), 2)}")) +
  annotate(geom = 'text', x = mean(six_3$V1) + 0.2, y = -4.75,
    hjust = 0, size = 5,
    label = str_glue("SD = {round(sd(six_3$V1), 2)}")) +
  labs(title = 'B: Constrained (0-10 range)',
    caption = 'Parameters: Mean = 6.2, SD = 2.2, Cor = 0.8') +
  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 = 6.2, SD = 2.2, Cor = 0.8

```
# Linear regression  
summary(lm(V2 ~ V1, data = six_3))  
  
##  
## Call:  
## lm(formula = V2 ~ V1, data = six_3)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -3.8375 -0.8391  0.0385  0.8526  4.3513   
##  
## Coefficients:
```



```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.22547    0.12557    9.76  <2e-16 ***
## V1          0.79450    0.01972   40.29  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.31 on 998 degrees of freedom
## Multiple R-squared:  0.6192, Adjusted R-squared:  0.6189
## F-statistic: 1623 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
six_3V1 <- six_3$V1

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

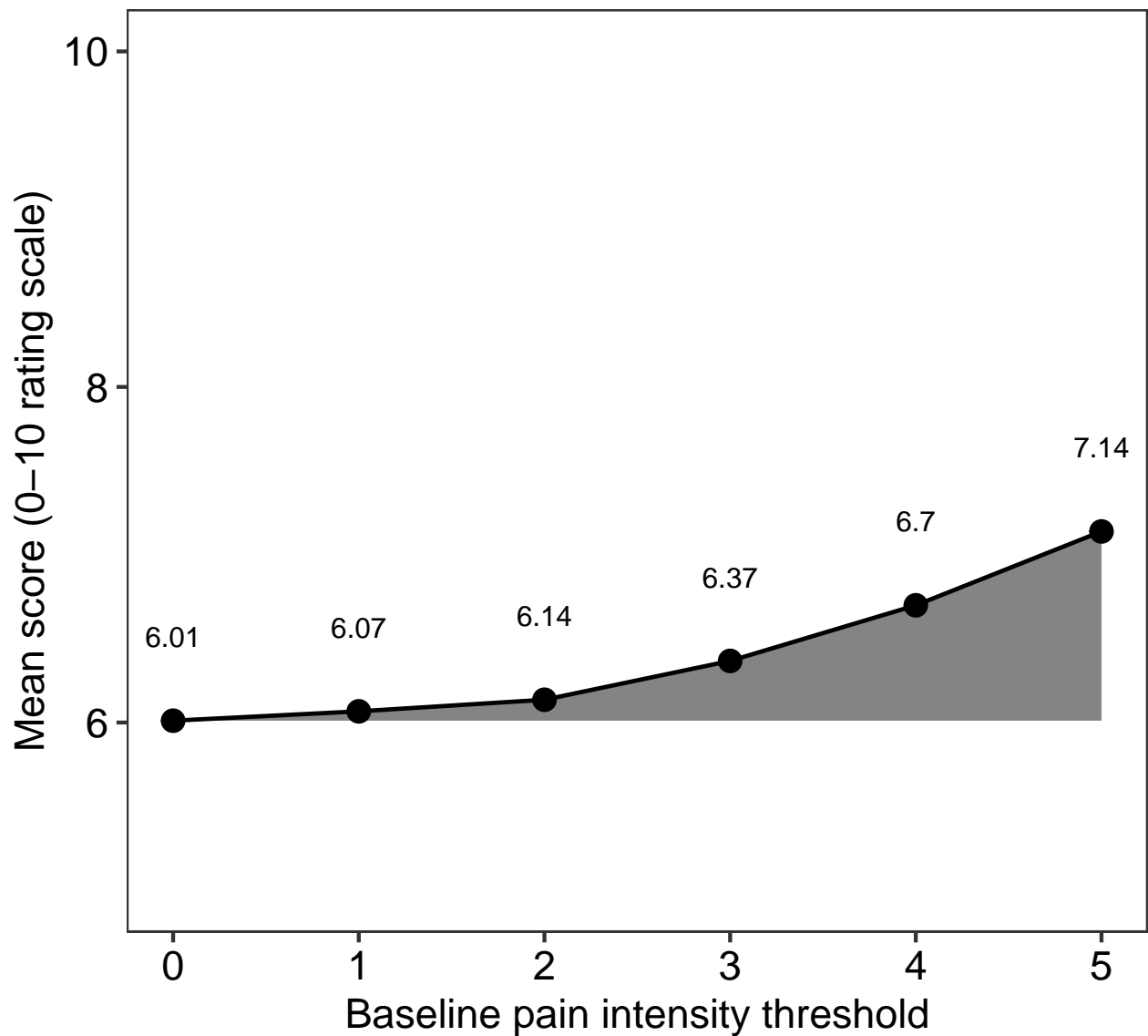
# Generate a vector of V1 means at each V1 threshold
six_3V1.shift <- sapply(cutoff, function(x){mean(six_3V1[six_3V1 > x])})

# Calculate deviation
(six_3V1.df <- data.frame(cutoff = cutoff,
                          mean = six_3V1.shift) %>%
  mutate(deviation = mean - mean(six_3V1)))

##   cutoff    mean deviation
## 1      0 6.010948 0.00000000
## 2      1 6.066681 0.05573349
## 3      2 6.135638 0.12469053
## 4      3 6.367139 0.35619093
## 5      4 6.698671 0.68772344
## 6      5 7.138821 1.12787318

# Plot data
ggplot(data = six_3V1.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 2.2, Cor = 0.8',
       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 = 6.2, SD = 2.2, Cor = 0.8

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

```
# Extract visit 2 data
six_3V2 <- six_3$V2

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

# Generate a vector of V2 means at each V1 threshold
six_3V2.shift <- map_dbl(.x = cutoff,
  ~ six_3 %>%
    filter(V1 > .x) %>%
    .$V2 %>%
```

```

mean(.))

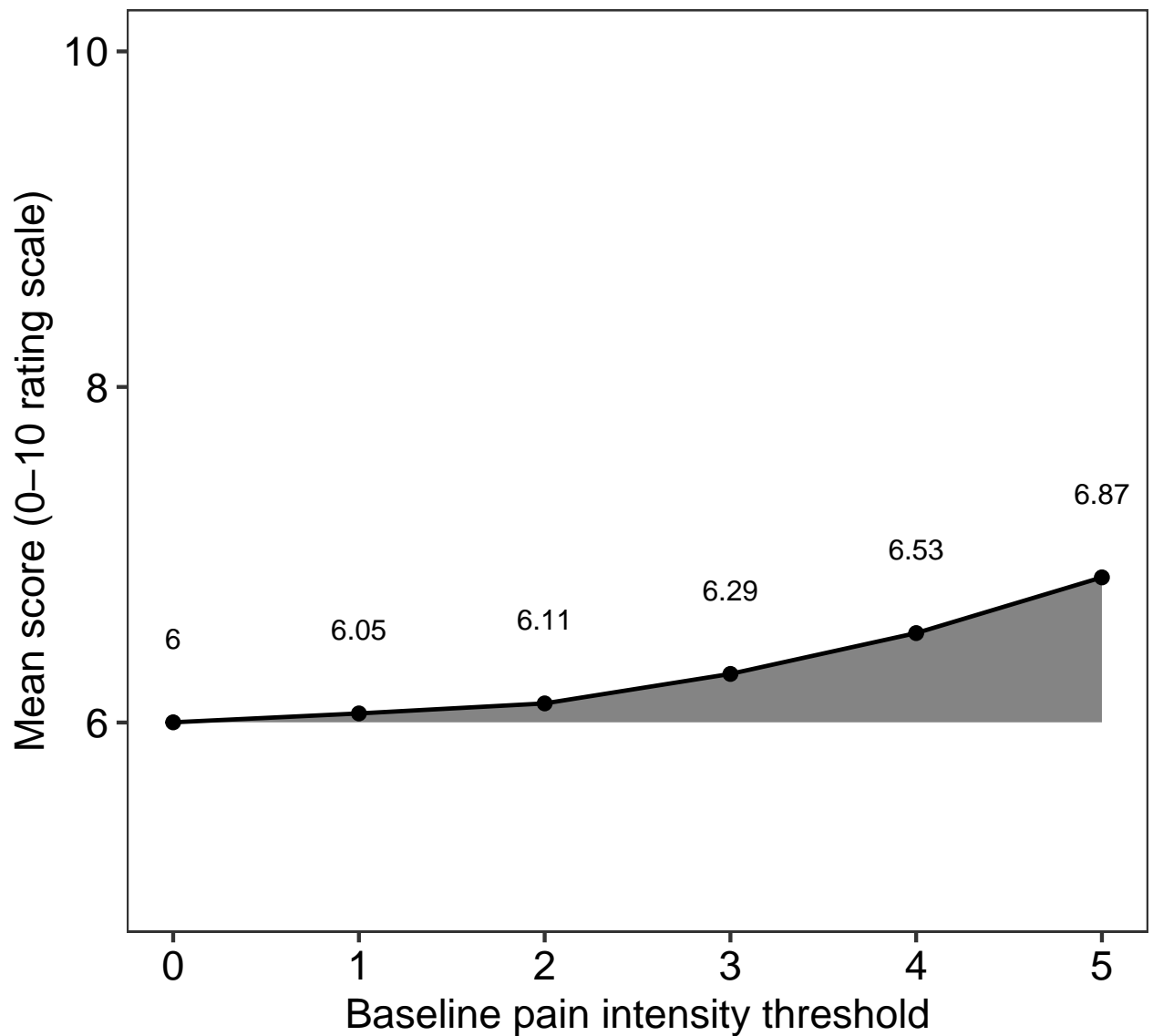
# Calculate deviation
(six_3V2.df <- data.frame(cutoff = cutoff,
                          mean = six_3V2.shift) %>%
  mutate(deviation = mean - mean(six_3V2)))

##   cutoff    mean deviation
## 1      0 6.001181 0.00000000
## 2      1 6.054209 0.05302876
## 3      2 6.113952 0.11277140
## 4      3 6.289654 0.28847314
## 5      4 6.533048 0.53186734
## 6      5 6.865184 0.86400355

# Plot data
ggplot(data = six_3V2.df) +
  aes(x = cutoff, y = mean, ymin = mean(six_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 = 6.2, SD = 2.2, Cor = 0.8',
       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



Placebo response

threshold: 0

```
# Process data
placebo_3.0 <- six_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	0.00977	0.95	-0.0779	0.0943

```
# Plot the data
```

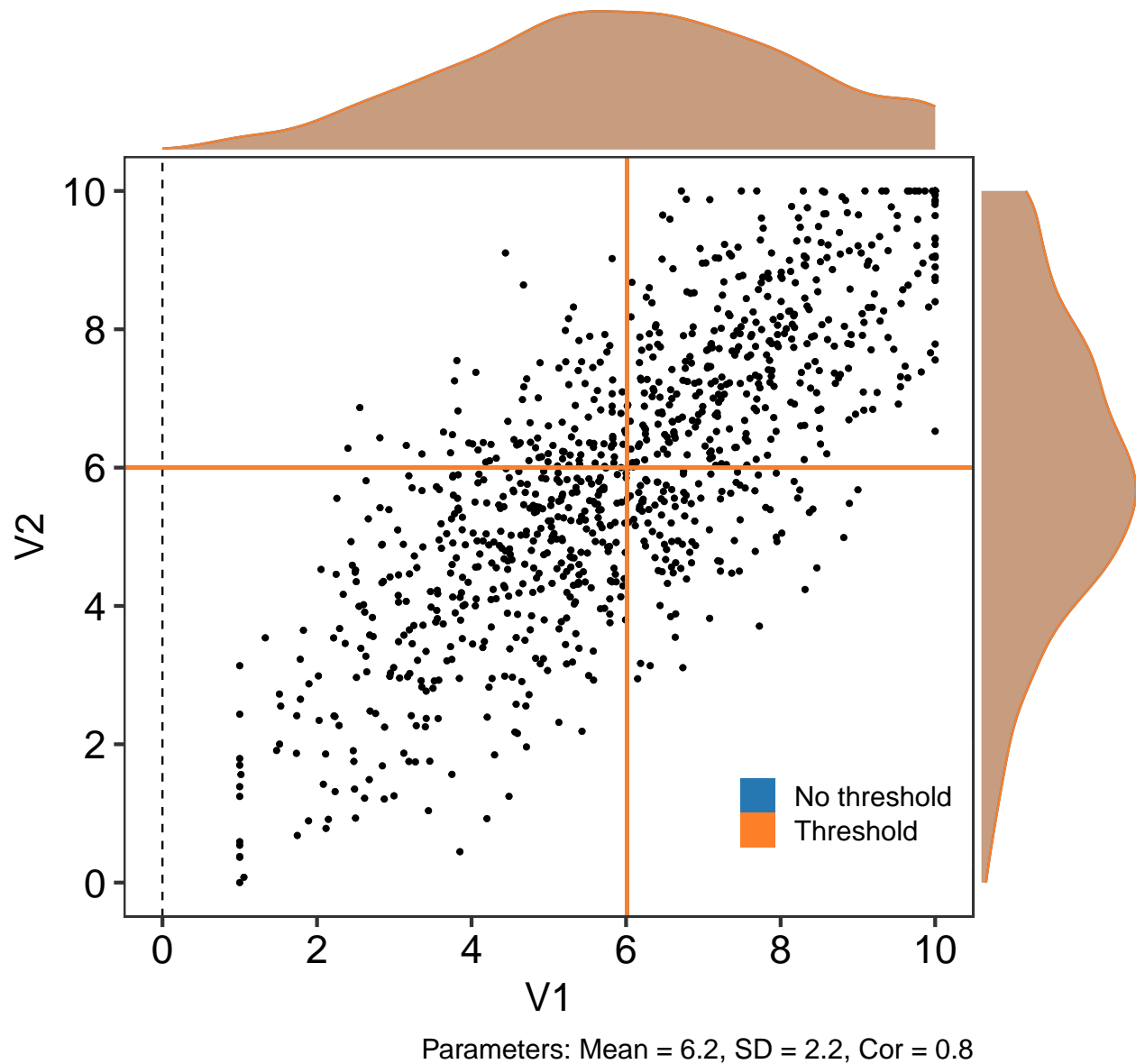
```

ggMarginal(placebo_3.0[, 1:3] %>%
  bind_rows(six_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 = six_3,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_3.0,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 2.2, Cor = 0.8') +
  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 <- six_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	915	0.0775	0.95	-0.00917	0.171

```
# Plot the data
```

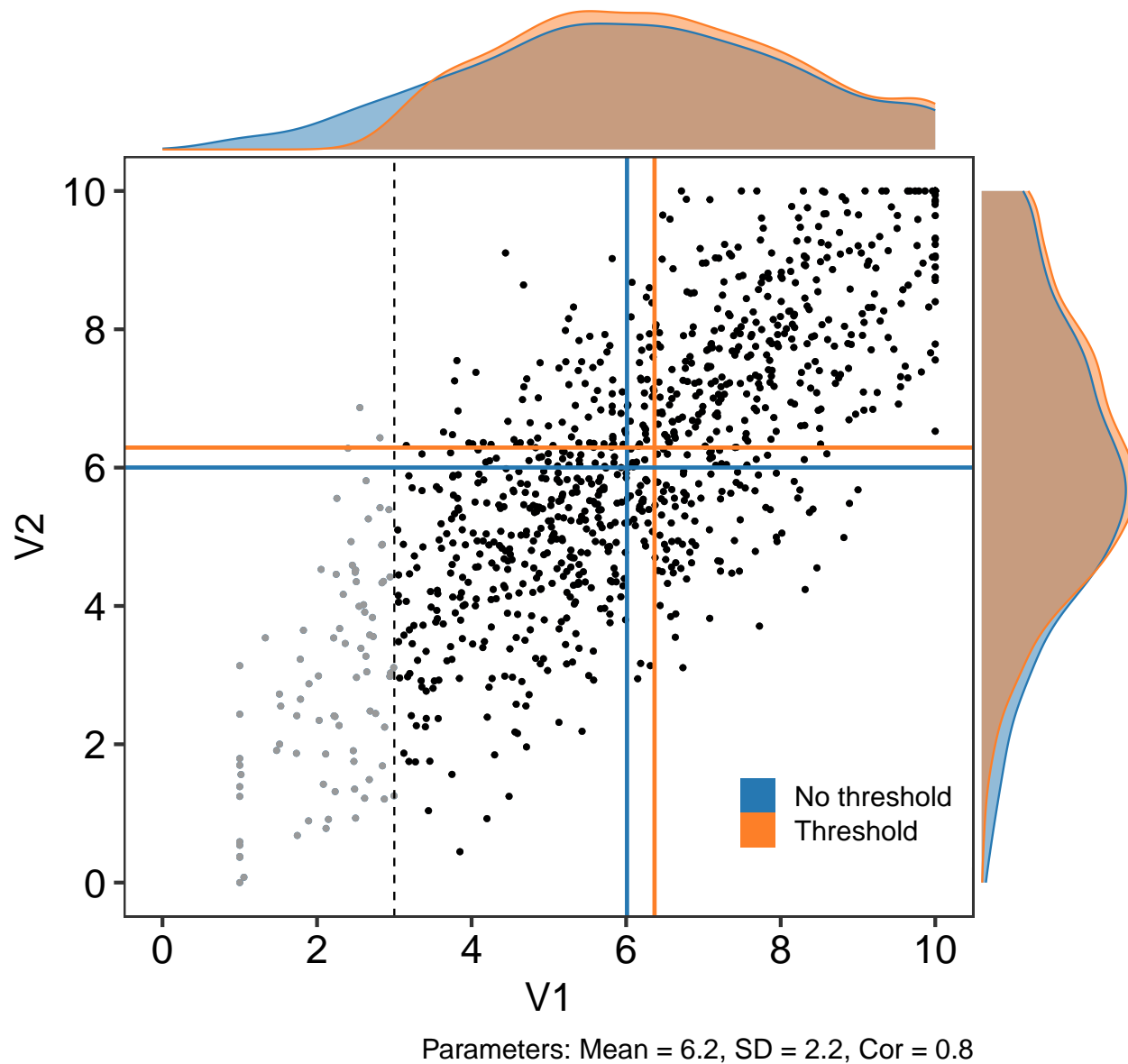
```

ggMarginal(placebo_3.3[, 1:3] %>%
  bind_rows(six_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 = six_3,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_3.3,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 2.2, Cor = 0.8') +
  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 <- six_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	819	0.166	0.95	0.0702	0.259

```
# Plot the data
```

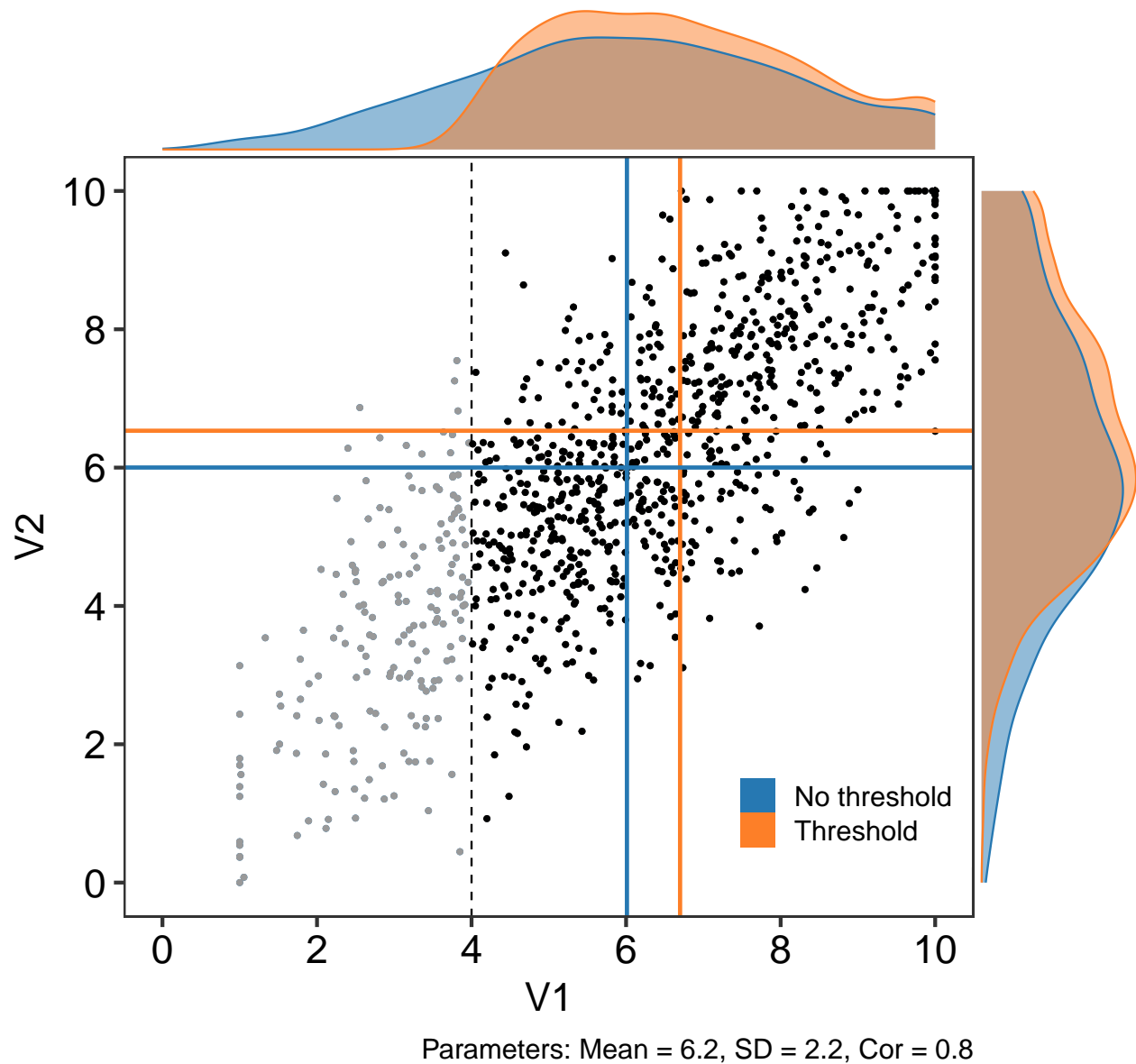
```

ggMarginal(placebo_3.4[, 1:3] %>%
  bind_rows(six_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 = six_3,
            colour = '#999999',
            size = 1) +
  geom_point(data = placebo_3.4,
            size = 1,
            colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 2.2, Cor = 0.8') +
  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 <- six_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	680	0.274	0.95	0.174	0.371

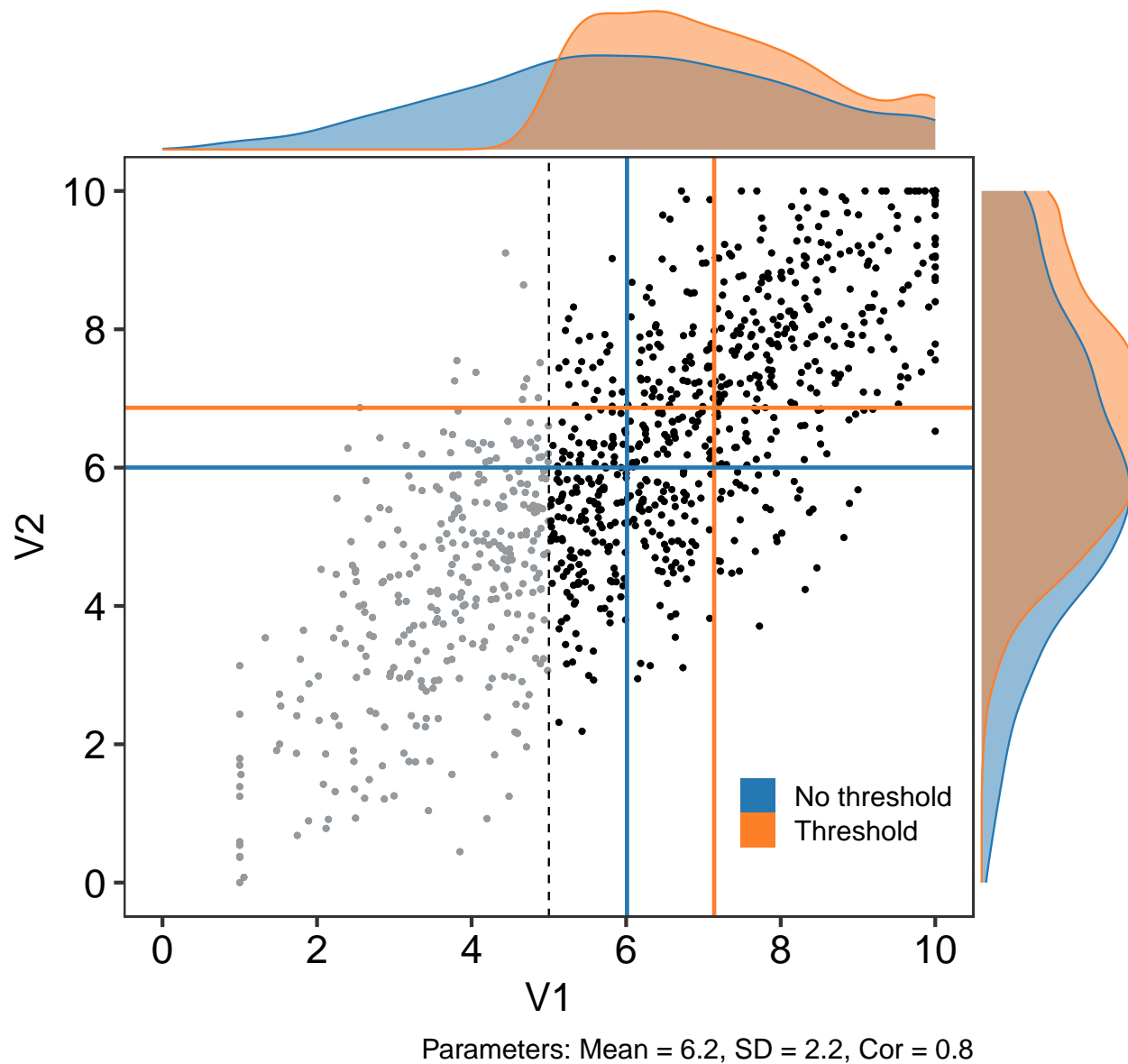
```

# Plot the data
ggMarginal(placebo_3.5[, 1:3] %>%
  bind_rows(six_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 = six_3,
    colour = '#999999',
    size = 1) +
  geom_point(data = placebo_3.5,
    size = 1,
    colour = '#000000') +
  geom_vline(xintercept = mean(six_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(six_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 = 6.2, SD = 2.2, Cor = 0.8') +
  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



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) +
```

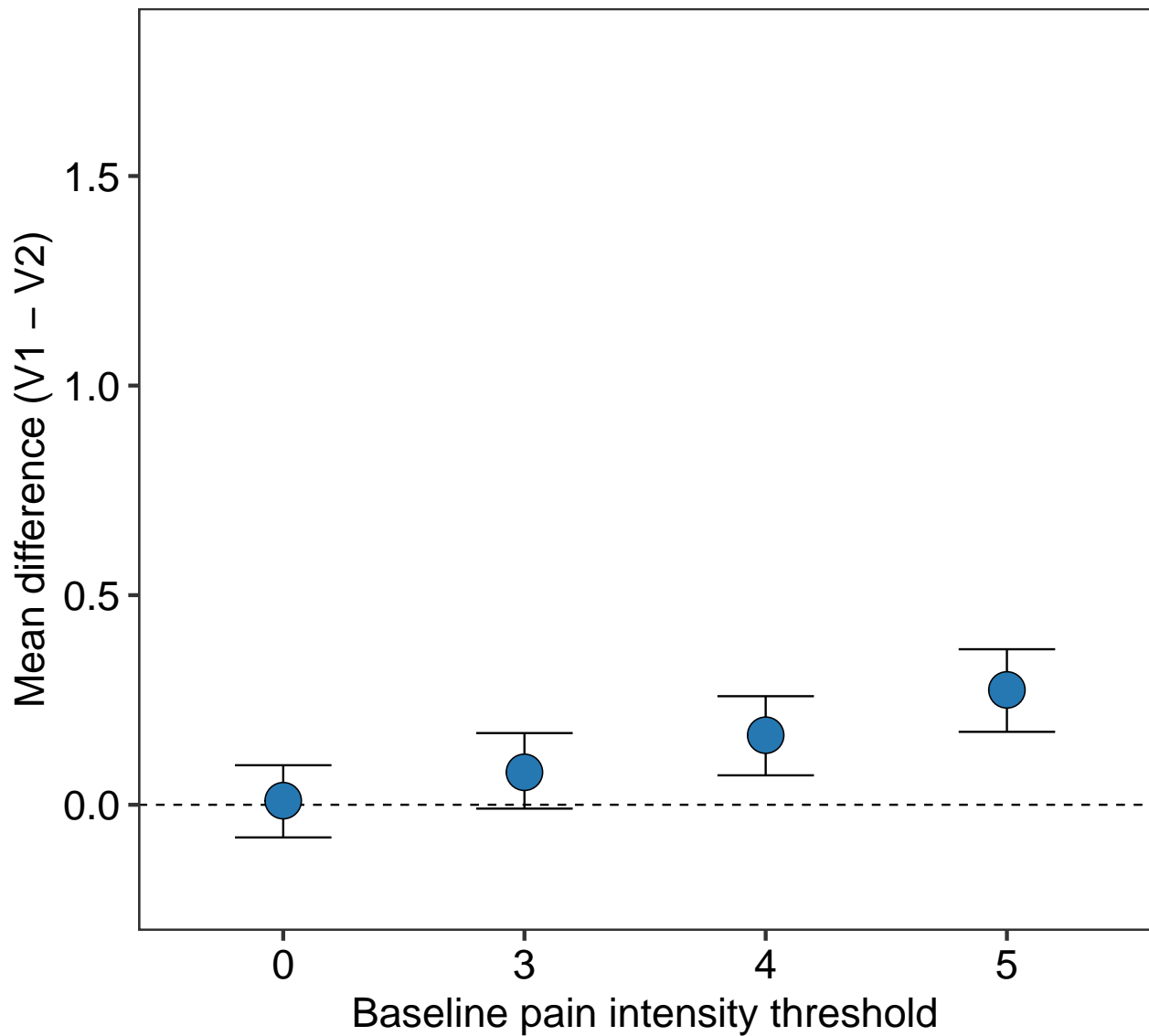
```

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 = 6.2, SD = 2.2, Cor = 0.8',
     x = 'Baseline pain intensity threshold',
     y = 'Mean difference (V1 - V2)' +
scale_y_continuous(limits = c(-0.2, 1.8)); pp_3

```

C

Parameters: Mean = 6.2, SD = 2.2, Cor = 0.8



Publication plot

```
pp_4 <- pp_1 + pp_2 + pp_3
ggsave('figures/6.2_0.8.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
```