Regression to the mean modeling

Mean NRS pain rating of 6.2 at 0.8 correlation

Peter Kamerman

Last knitted: 02 August 2019

Contents

Generate 2x2 covariance matrix

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

```
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
```

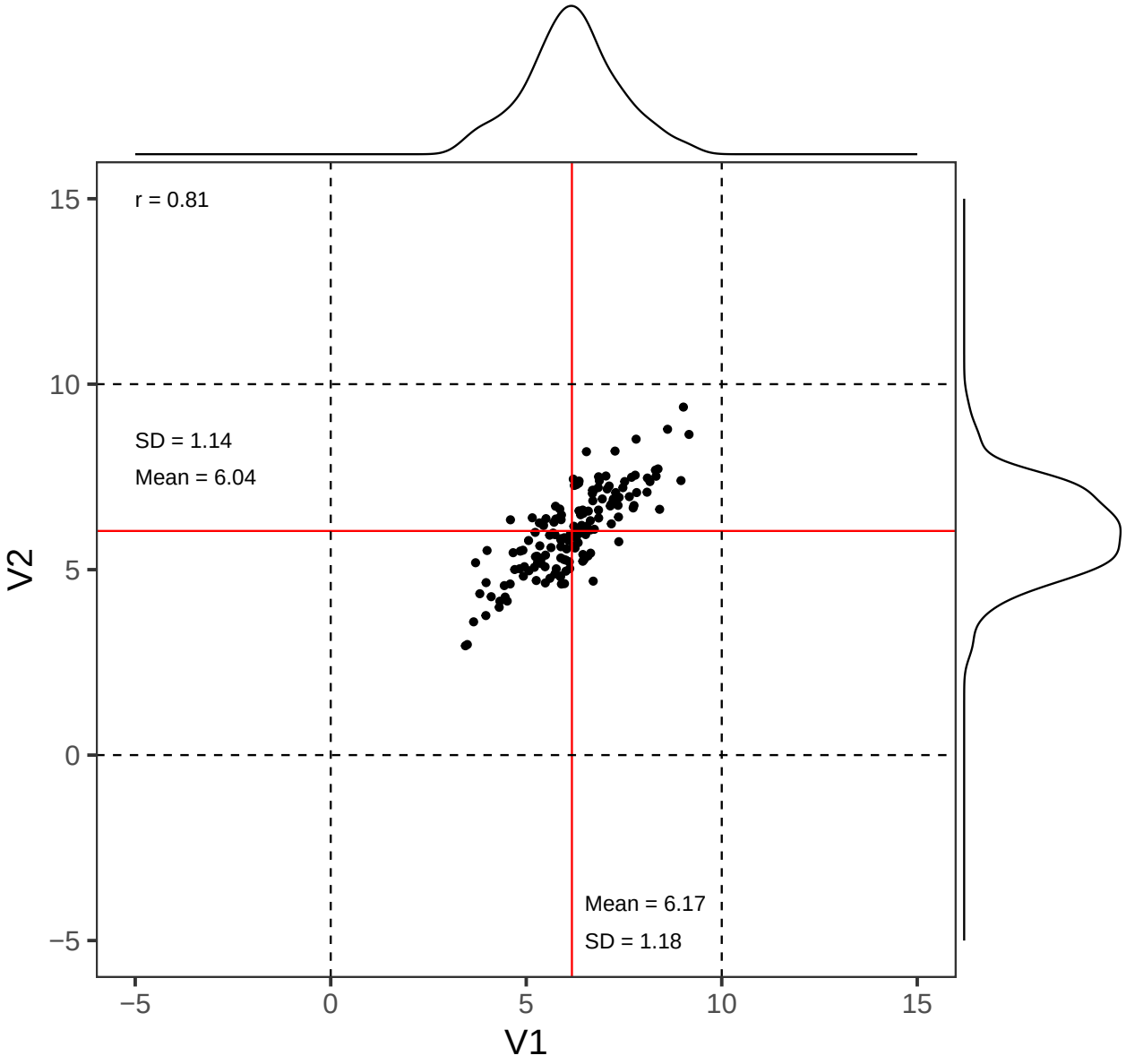
Mean = 6.2, SD = 1.2, cor = 0.8

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 = 150, 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$V2), colour = 'red') +
               geom_vline(xintercept = mean(six_1.base$V1), colour = 'red') +
               geom_hline(yintercept = 0, linetype = 2) +
               geom_hline(yintercept = 10, linetype = 2) +
               geom_vline(xintercept = 0, linetype = 2) +
```

```
geom_vline(xintercept = 10, linetype = 2) +
annotate(geom = 'text', x = -5, y = 15, hjust = 0,
         label = str_glue("r = {round(cor(six_1.base$V1, six_1.base$V2), 2)}")) +
annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
         label = str_glue("Mean = {round(mean(six_1.base$V2), 2)}")) +
annotate(geom = 'text', x = -5, y = 8.5, hjust = 0,
         label = str_glue("SD = {round(sd(six_1.base$V2),2)}")) +
annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
         label = str_glue("Mean = {round(mean(six_1.base$V1), 2)}")) +
annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
         label = str_glue("SD = {round(sd(six_1.base$V1), 2)}")) +
labs(subtitle = 'Mean = 6.2, SD = 1.2, cor = 0.8') +
scale_y_continuous(limits = c(-5, 15)) +
scale_x_continuous(limits = c(-5, 15)) +
theme(panel.grid = element_blank(),
      plot.subtitle = element_text(size = 16)))
```



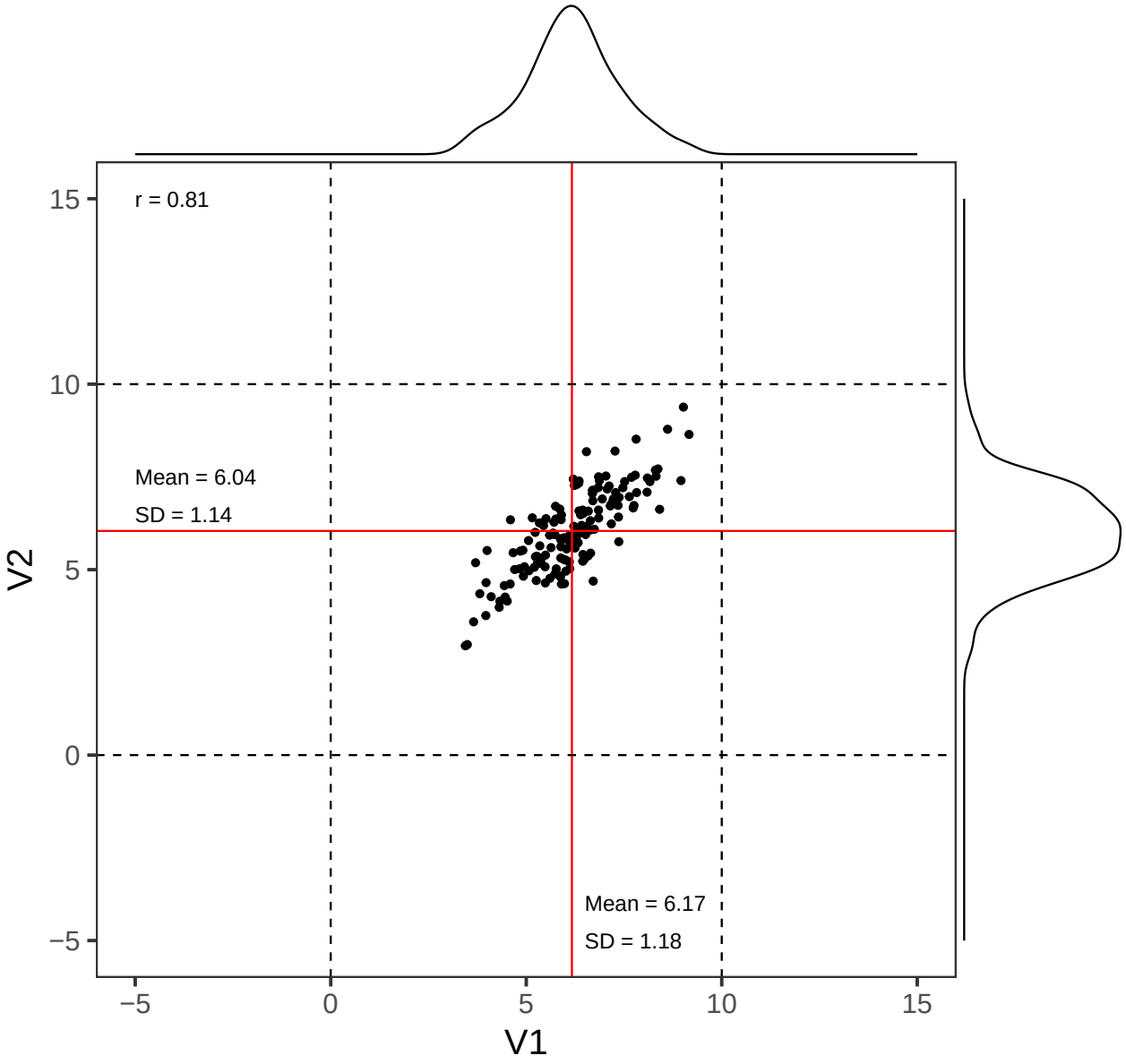

```
# Linear regression
summary(lm(V2 ~ V1, data = six_1.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_1.base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78128 -0.44281 -0.06263  0.45286  1.84462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.20139    0.29131   4.124 6.18e-05 ***
## V1           0.78499    0.04639  16.920  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6682 on 148 degrees of freedom
## Multiple R-squared:  0.6592, Adjusted R-squared:  0.6569
## F-statistic: 286.3 on 1 and 148 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 cut-off')

# Plot processed data
ggMarginal(ggplot(data = six_1) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(six_1$V2), colour = 'red') +
               geom_vline(xintercept = mean(six_1$V1), colour = 'red') +
               geom_hline(yintercept = 0, linetype = 2) +
               geom_hline(yintercept = 10, linetype = 2) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_vline(xintercept = 10, linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0,
                        label = str_glue("r = {round(cor(six_1$V1, six_1$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_1$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 6.5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_1$V2),2)}")) +
               annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_1$V1), 2)}")) +
               annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_1$V1), 2)}")) +
               labs(subtitle = 'Mean = 6.2, SD = 1.2 (constrained), cor = 0.8') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)))
```




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

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78128 -0.44281 -0.06263  0.45286  1.84462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.20139    0.29131   4.124 6.18e-05 ***
## V1           0.78499    0.04639  16.920  < 2e-16 ***
```

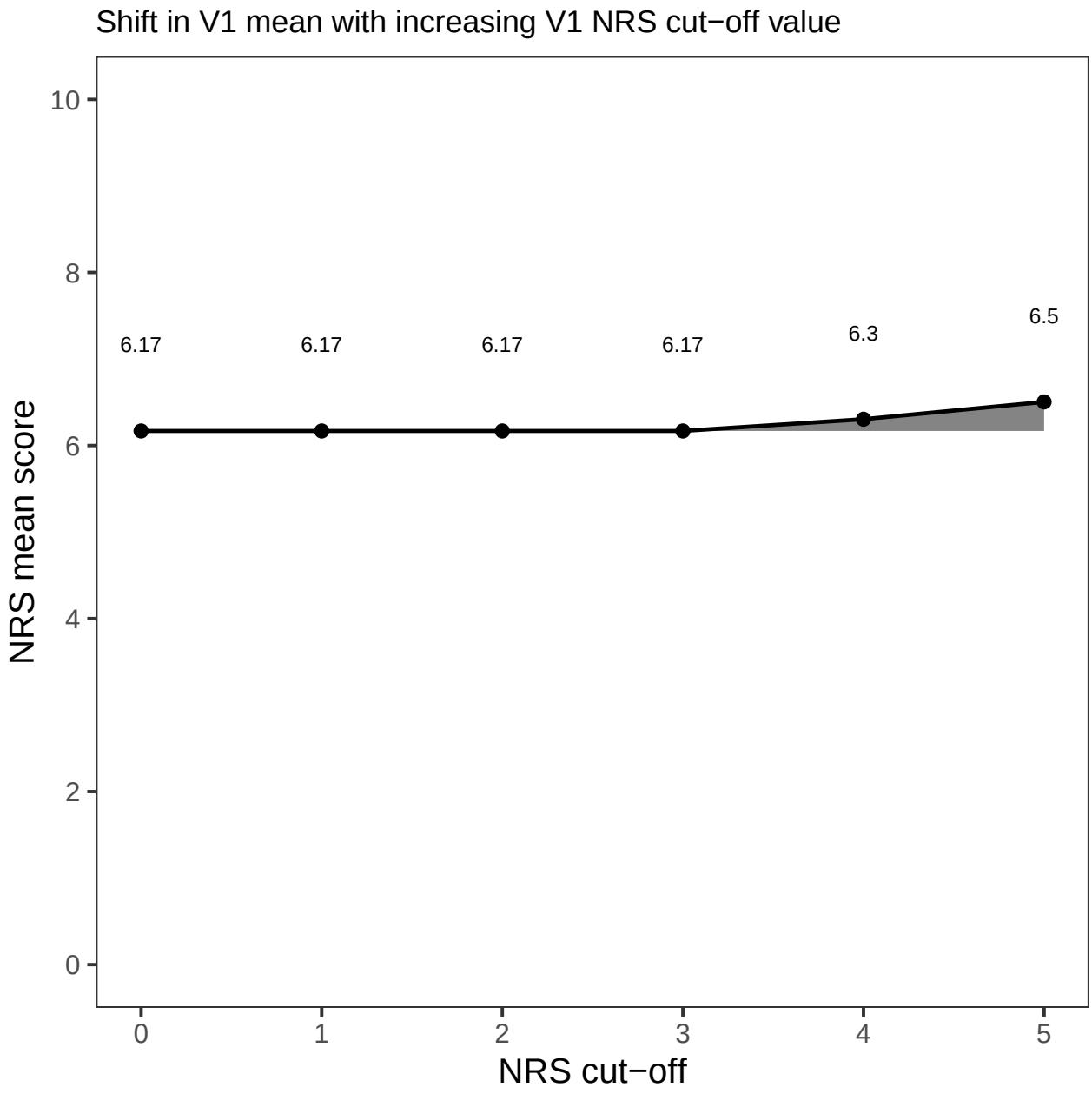
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6682 on 148 degrees of freedom
## Multiple R-squared:  0.6592, Adjusted R-squared:  0.6569
## F-statistic: 286.3 on 1 and 148 DF,  p-value: < 2.2e-16
```

Model mean of V1 with increasing V1 cut-offs from 0 to 5

```
# Extract visit 1 data
six_1V1 <- six_1$V1

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

# Generate a vector of V1 means at each V1 cut-off
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 6.168004 0.0000000
## 2      1 6.168004 0.0000000
## 3      2 6.168004 0.0000000
## 4      3 6.168004 0.0000000
## 5      4 6.303873 0.1358689
## 6      5 6.503691 0.3356872
```

```
# 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 = 3) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)), nudge_y = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V1 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



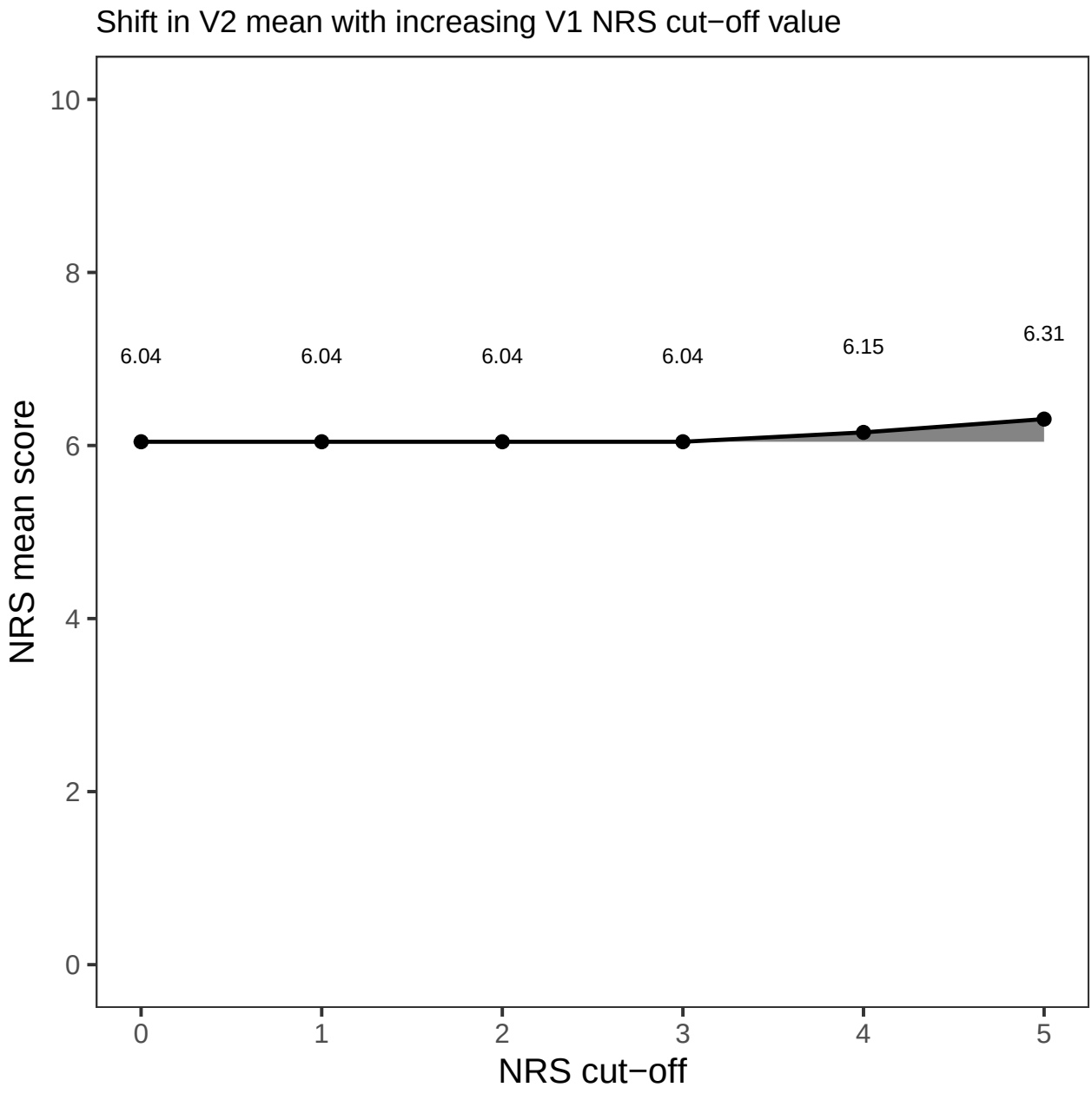
Model mean of V2 with increasing V1 cut-offs from 0 to 5

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

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

# Generate a vector of V2 means at each V1 cut-off
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 6.043226 0.0000000
## 2      1 6.043226 0.0000000
## 3      2 6.043226 0.0000000
## 4      3 6.043226 0.0000000
## 5      4 6.151412 0.1081861
## 6      5 6.305127 0.2619005
```

```
# 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 = 3) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)), nudge_y = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V2 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



Placebo response

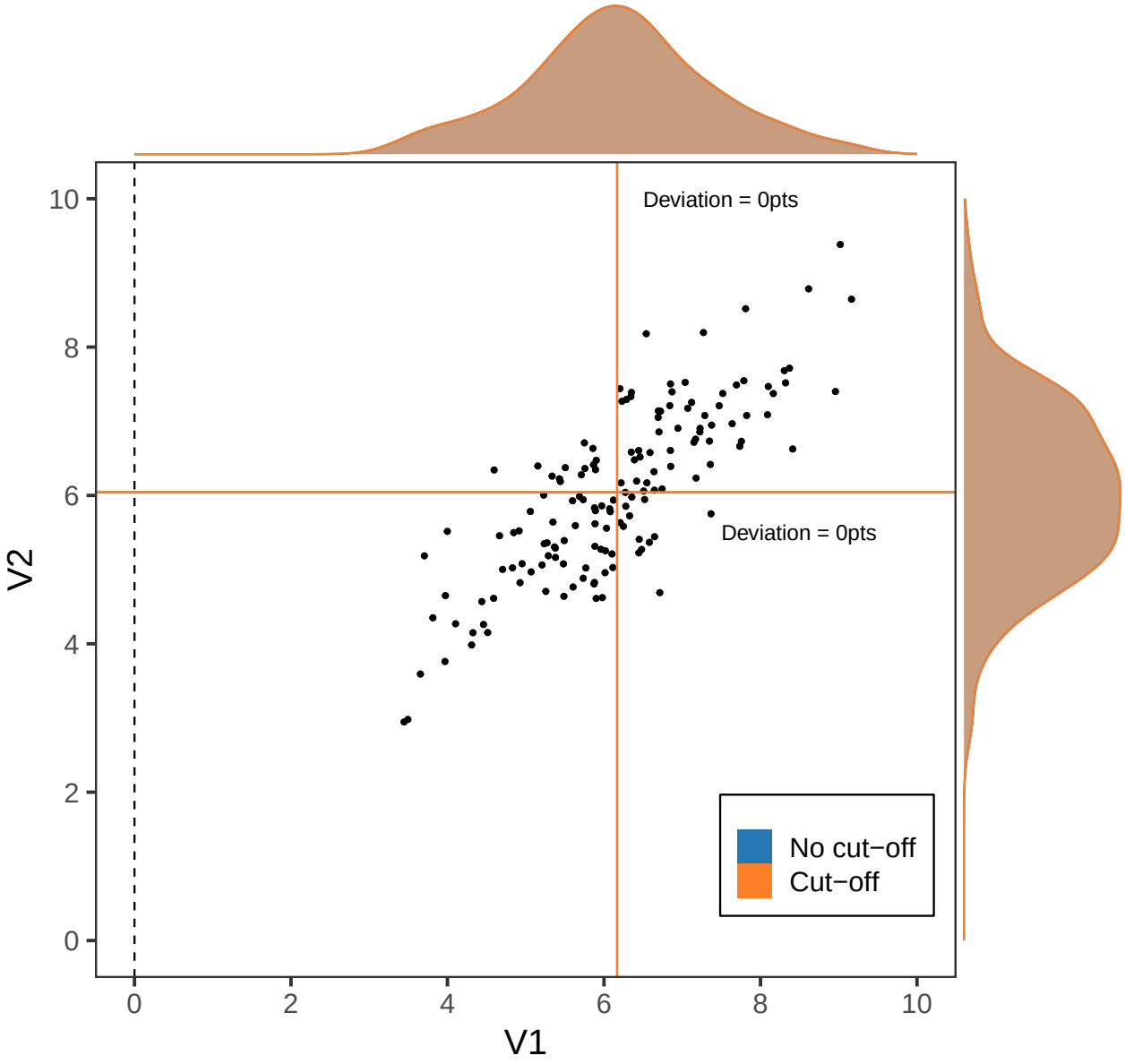
threshold: 0

```
# Process data
placebo_1.0 <- six_1 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

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

```
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	150	0.125	0.95	0.00306	0.24

```
# Plot the data
ggMarginal(placebo_1.0[, 1:3] %>%
               bind_rows(six_1) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_1.0$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(six_1$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_1.0$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 0') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
             groupColour = TRUE,
             groupFill = TRUE)
```



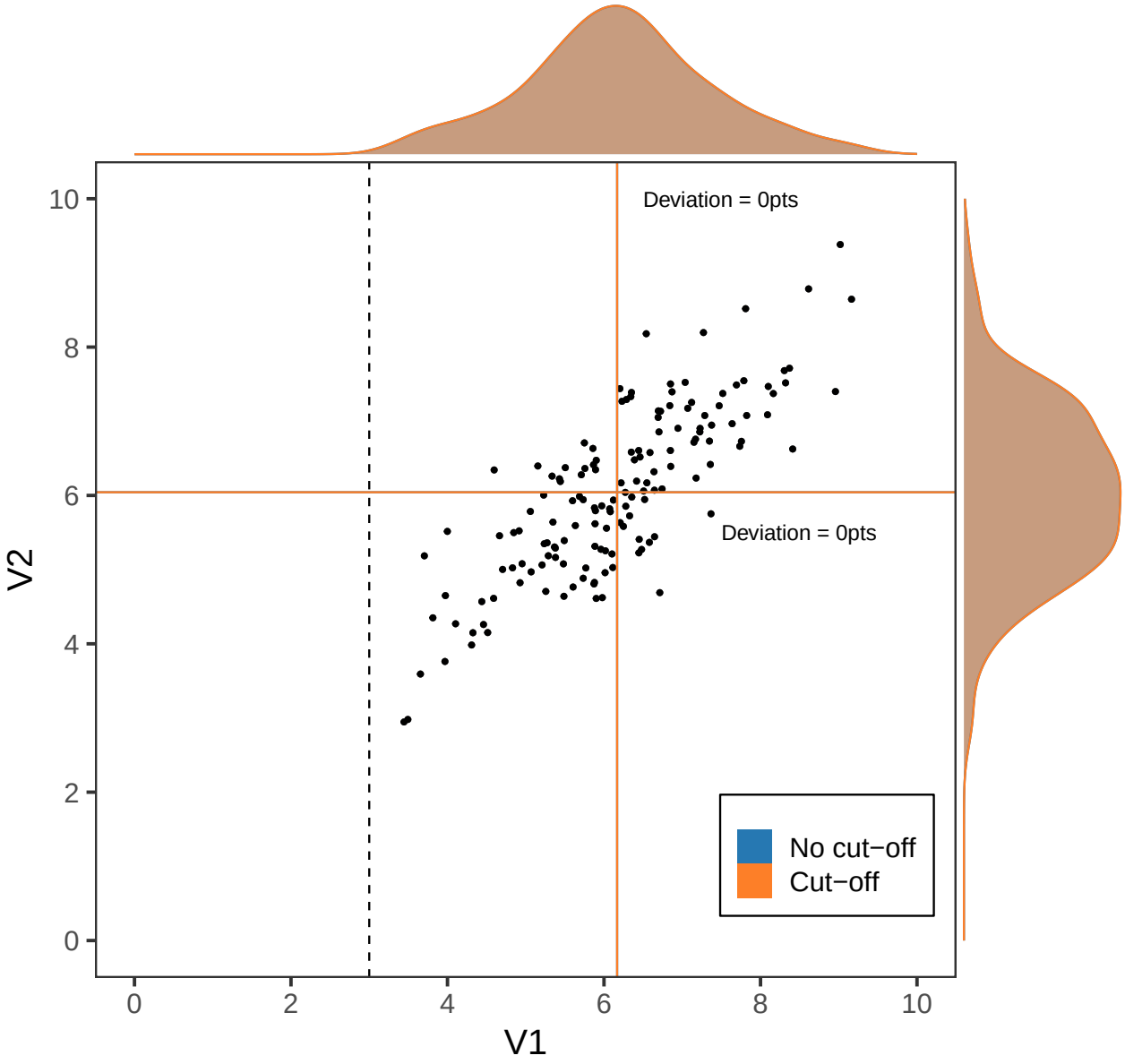
threshold: 3

```
# Process data
placebo_1.3 <- six_1 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_1.3,
              R = 5000,
              traditional = FALSE,
```

```
            bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	150	0.125	0.95	0.0144	0.236

```
# Plot the data
ggMarginal(placebo_1.3[, 1:3] %>%
               bind_rows(six_1) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_1.3$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(six_1$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_1.3$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 3') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
           groupColour = TRUE,
           groupFill = TRUE)
```



threshold: 4

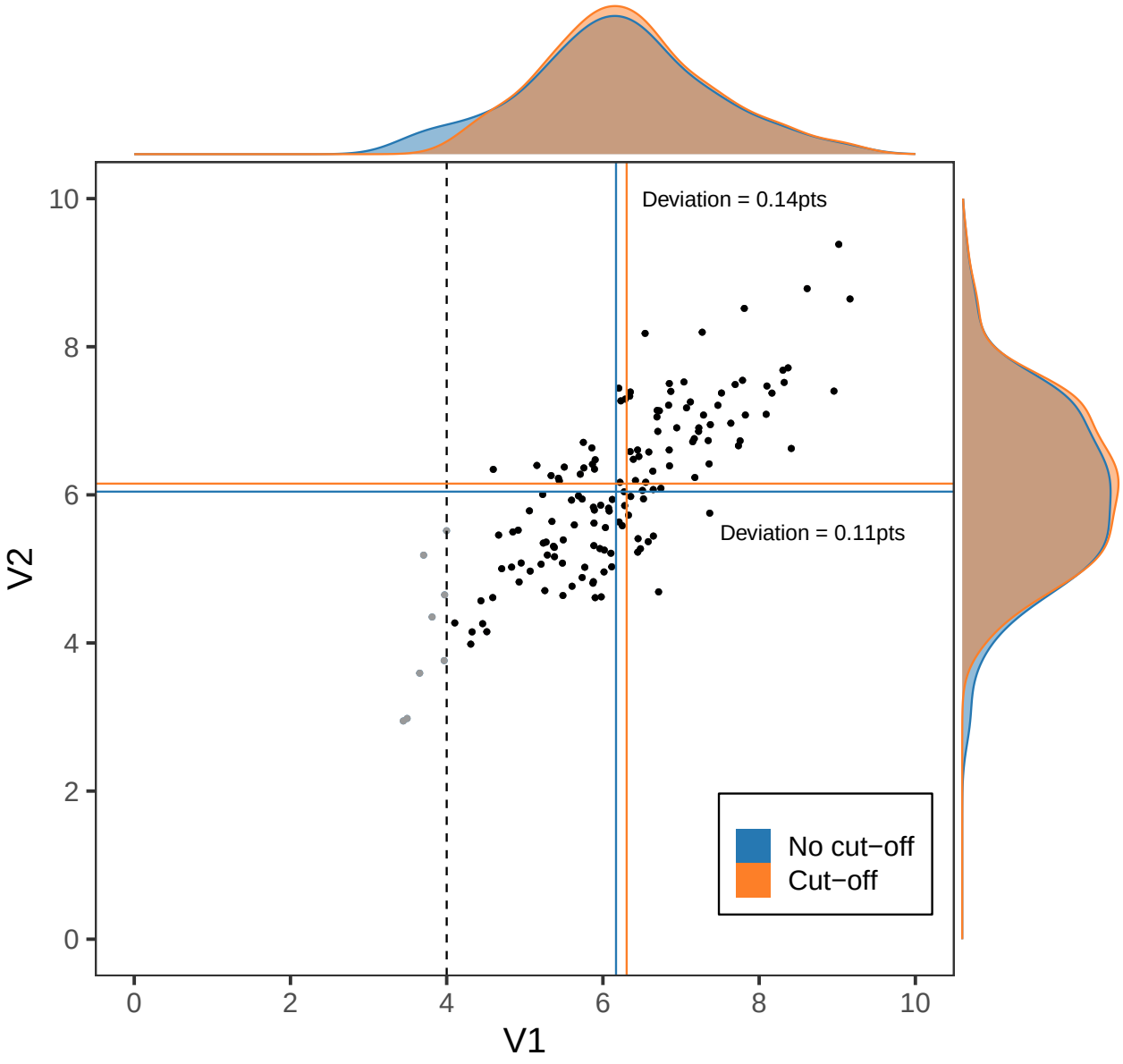
```
# Process that data
placebo_1.4 <- six_1 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
```

```
            data = placebo_1.4,
            R = 5000,
            traditional = FALSE,
            bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	142	0.152	0.95	0.0361	0.268

```
# Plot the data
ggMarginal(placebo_1.4[, 1:3] %>%
             bind_rows(six_1) %>%
             mutate(group = factor(group,
                                   levels = c('No cut-off', 'Cut-off'),
                                   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$V1),
                        colour = pal[1]) +
             geom_vline(xintercept = mean(placebo_1.4$V1),
                        colour = pal[2]) +
             geom_vline(xintercept = 4, linetype = 2) +
             geom_hline(yintercept = mean(six_1$V2),
                        colour = pal[1]) +
             geom_hline(yintercept = mean(placebo_1.4$V2),
                        colour = pal[2]) +
             annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                      label = str_glue("Deviation = {round(six_1V1.df$deviation[5], 2)}pts")) +
             annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                      label = str_glue("Deviation = {round(six_1V2.df$deviation[5], 2)}pts")) +
             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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 4') +
             theme(legend.title = element_blank(),
                   legend.position = c(0.85, 0.15),
                   legend.background = element_rect(colour = '#000000',
                                                    size = 0.5),
                   panel.grid = element_blank(),
                   plot.subtitle = element_text(size = 16)),
           groupColour = TRUE,
```

```
    groupFill = TRUE)
```



threshold: 5

```
# Process that data
placebo_1.5 <- six_1 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)
```

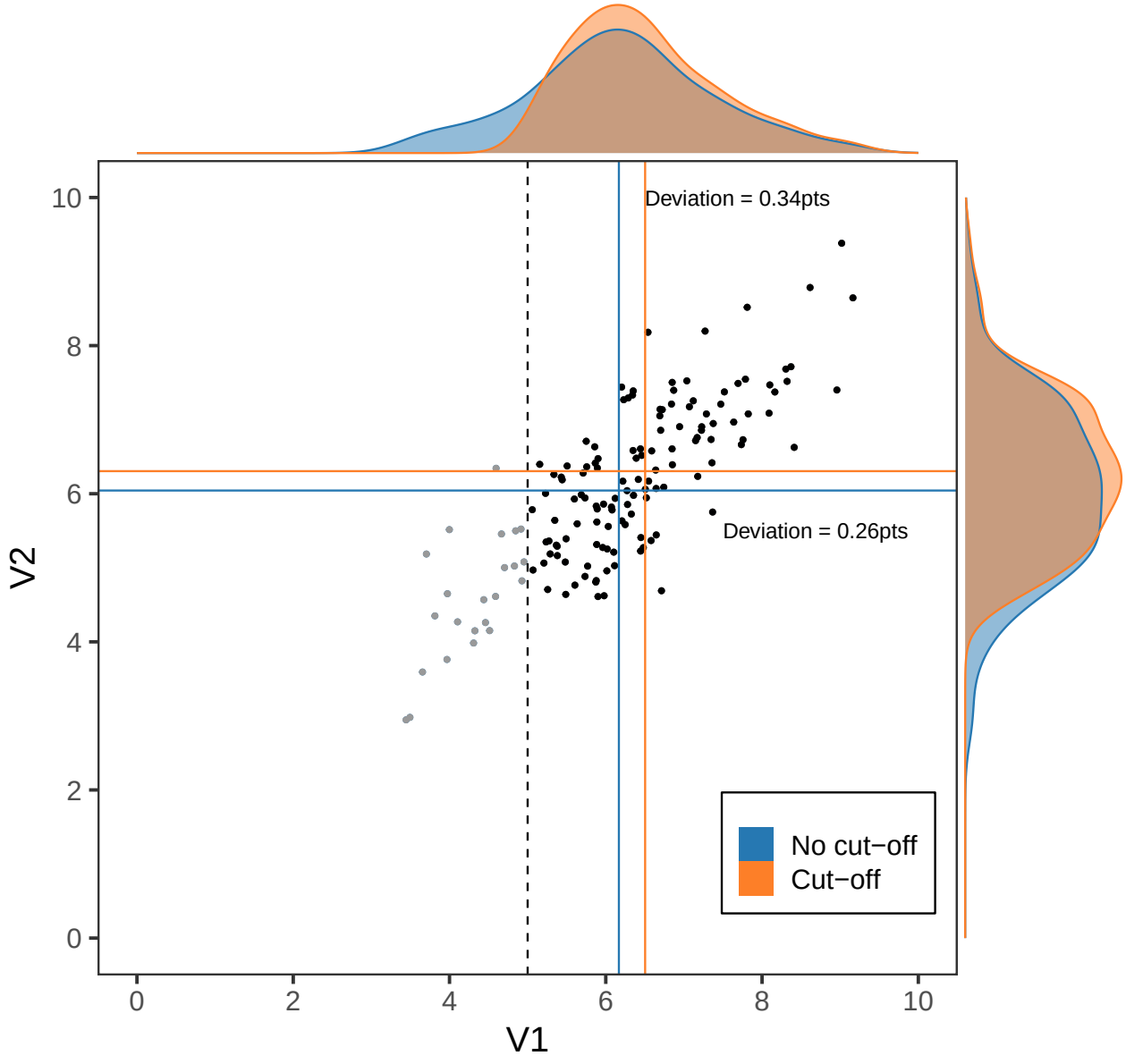
```r
# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_1.5,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	127	0.199	0.95	0.0767	0.32

```r
# Plot the data
ggMarginal(placebo_1.5[, 1:3] %>%
               bind_rows(six_1) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_1.5$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(six_1$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_1.5$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V1.df$deviation[6], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_1V2.df$deviation[6], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
```

```
            plot.subtitle = element_text(size = 16)),
    groupColour = TRUE,
    groupFill = TRUE)
```

Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut−off: 5

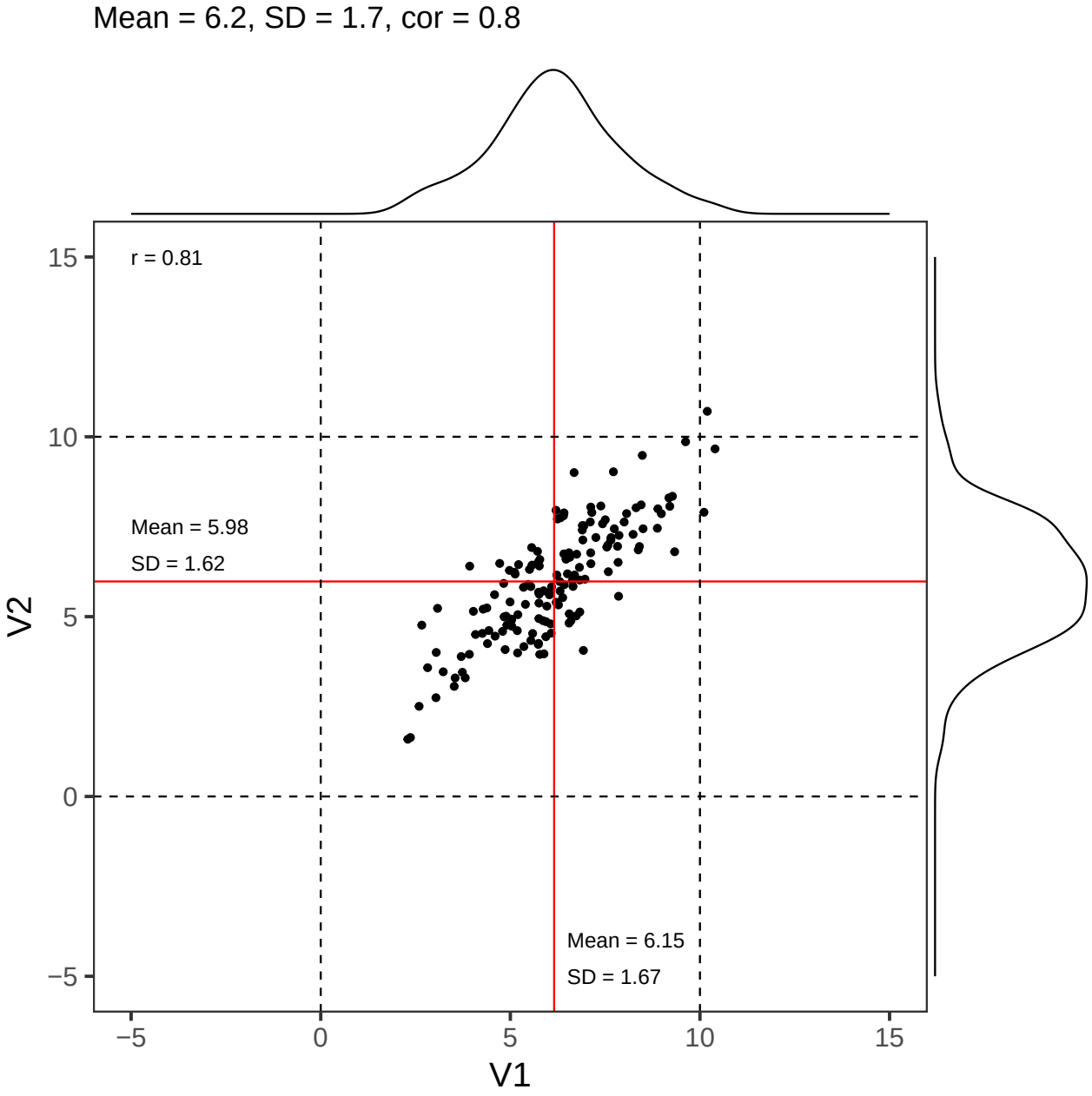


Mean = 6.2, SD = 1.7, cor = 0.8

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 = 150, 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$V2), colour = 'red') +
               geom_vline(xintercept = mean(six_2.base$V1), colour = 'red') +
               geom_hline(yintercept = 0, linetype = 2) +
               geom_hline(yintercept = 10, linetype = 2) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_vline(xintercept = 10, linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0,
                        label = str_glue("r = {round(cor(six_2.base$V1, six_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_2.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 6.5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_2.base$V2),2)}")) +
               annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_2.base$V1), 2)}")) +
               annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_2.base$V1), 2)}")) +
               labs(subtitle = 'Mean = 6.2, SD = 1.7, cor = 0.8') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)))
```



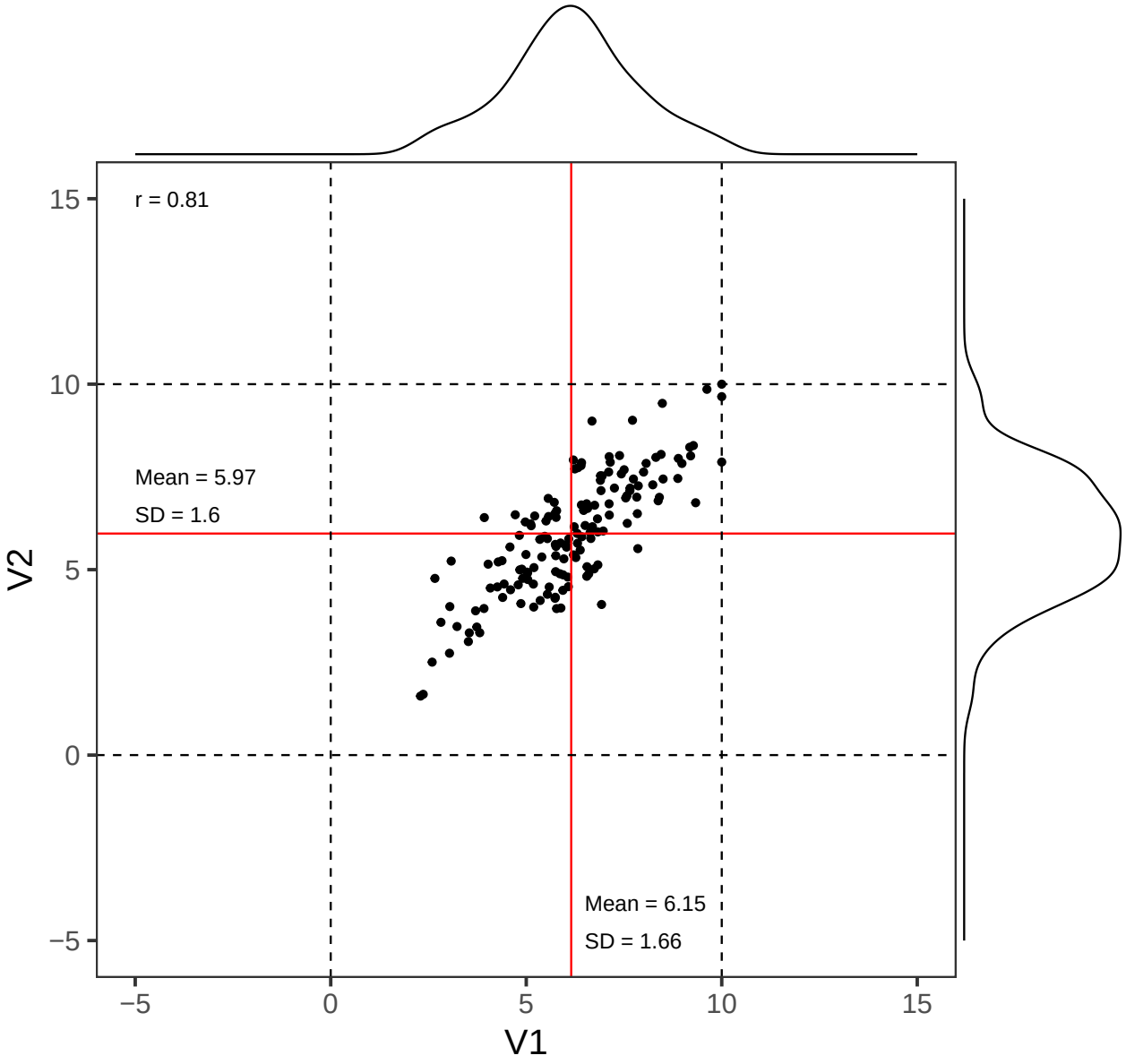

```
# Linear regression
summary(lm(V2 ~ V1, data = six_2.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_2.base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52349 -0.62731 -0.08872  0.64155  2.61321
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.14653    0.29581   3.876 0.000159 ***
## V1           0.78499    0.04639  16.920  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9466 on 148 degrees of freedom
## Multiple R-squared:  0.6592, Adjusted R-squared:  0.6569
## F-statistic: 286.3 on 1 and 148 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 cut-off')

# Plot processed data
ggMarginal(ggplot(data = six_2) +
                aes(x = V1, y = V2) +
                geom_point() +
                geom_hline(yintercept = mean(six_2$V2), colour = 'red') +
                geom_vline(xintercept = mean(six_2$V1), colour = 'red') +
                geom_hline(yintercept = 0, linetype = 2) +
                geom_hline(yintercept = 10, linetype = 2) +
                geom_vline(xintercept = 0, linetype = 2) +
                geom_vline(xintercept = 10, linetype = 2) +
                annotate(geom = 'text', x = -5, y = 15, hjust = 0,
                         label = str_glue("r = {round(cor(six_2$V1, six_2$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
                         label = str_glue("Mean = {round(mean(six_2$V2), 2)}")) +
                annotate(geom = 'text', x = -5, y = 6.5, hjust = 0,
                         label = str_glue("SD = {round(sd(six_2$V2),2)}")) +
                annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
                         label = str_glue("Mean = {round(mean(six_2$V1), 2)}")) +
                annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
                         label = str_glue("SD = {round(sd(six_2$V1), 2)}")) +
                labs(subtitle = 'Mean = 6.2, SD = 1.7 (constrained), cor = 0.8') +
                scale_y_continuous(limits = c(-5, 15)) +
                scale_x_continuous(limits = c(-5, 15)) +
                theme(panel.grid = element_blank(),
                      plot.subtitle = element_text(size = 16)))
```




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

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52066 -0.62987 -0.08419  0.65892  2.61548
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.15941    0.29601   3.917 0.000137 ***
## V1           0.78273    0.04648  16.841  < 2e-16 ***
```

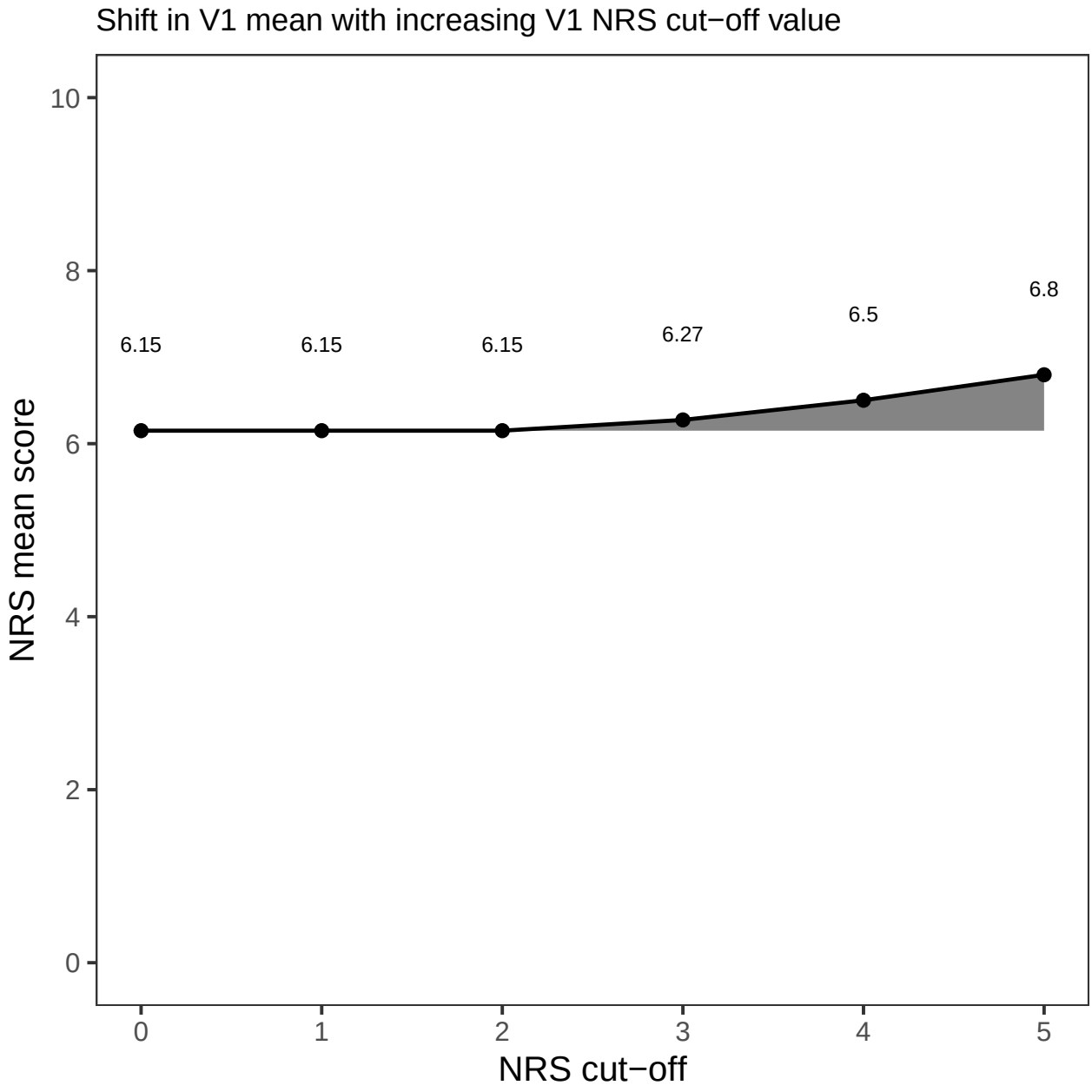
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9419 on 148 degrees of freedom
## Multiple R-squared:  0.6571, Adjusted R-squared:  0.6548
## F-statistic: 283.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

Model mean of V1 with increasing V1 cut-offs from 0 to 5

```
# Extract visit 1 data
six_2V1 <- six_2$V1

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

# Generate a vector of V1 means at each V1 cut-off
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 6.150009 0.0000000
## 2      1 6.150009 0.0000000
## 3      2 6.150009 0.0000000
## 4      3 6.274218 0.1242094
## 5      4 6.501513 0.3515039
## 6      5 6.796144 0.6461349
```

```
# 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 = 3) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)), nudge_y = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V1 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



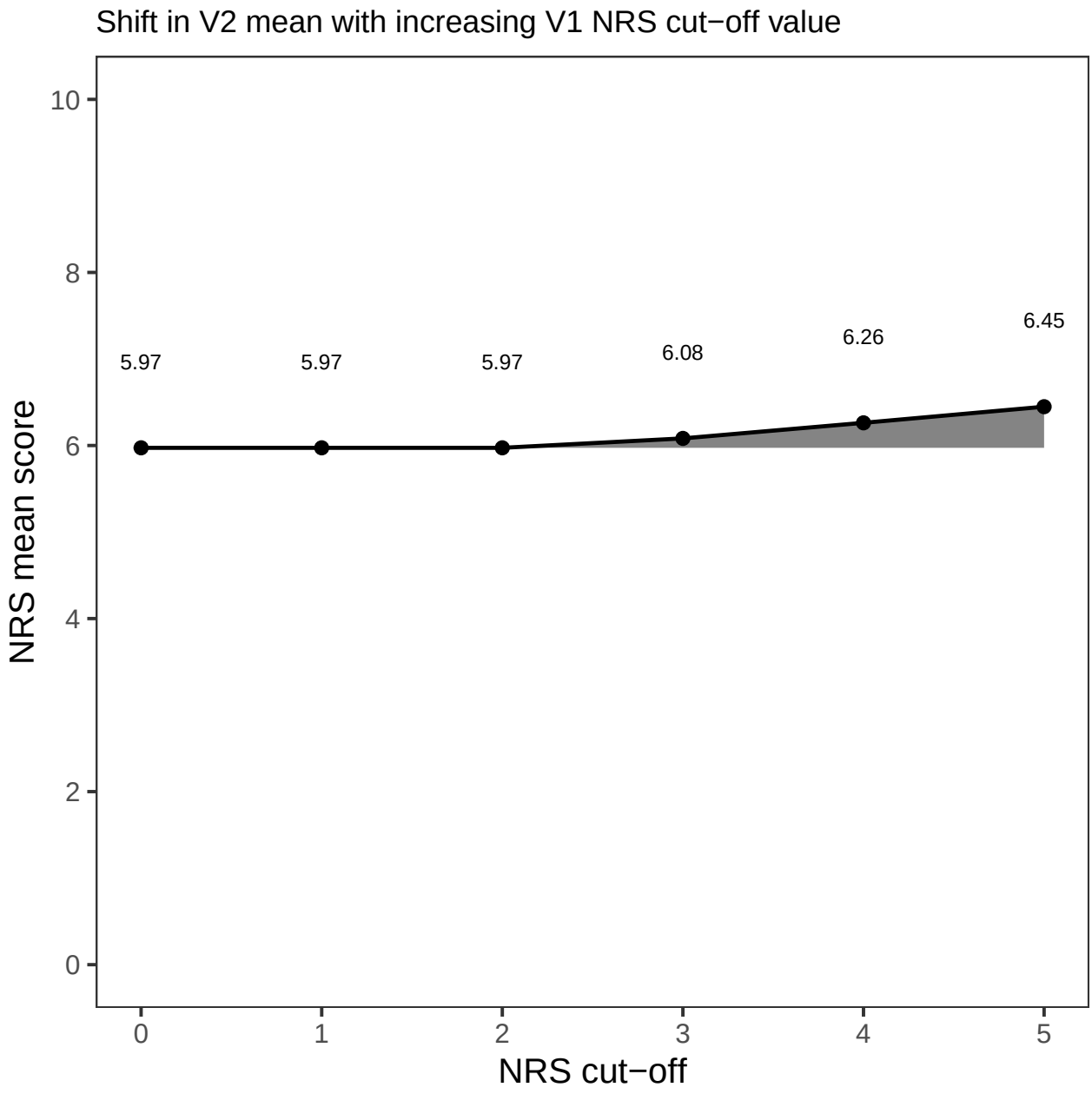
Model mean of V2 with increasing V1 cut-offs from 0 to 5

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

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

# Generate a vector of V2 means at each V1 cut-off
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 5.973181 0.0000000
## 2      1 5.973181 0.0000000
## 3      2 5.973181 0.0000000
## 4      3 6.082052 0.1088711
## 5      4 6.261946 0.2887654
## 6      5 6.448240 0.4750591
```

```
# 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 = 3) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)), nudge_y = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V2 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



Placebo response

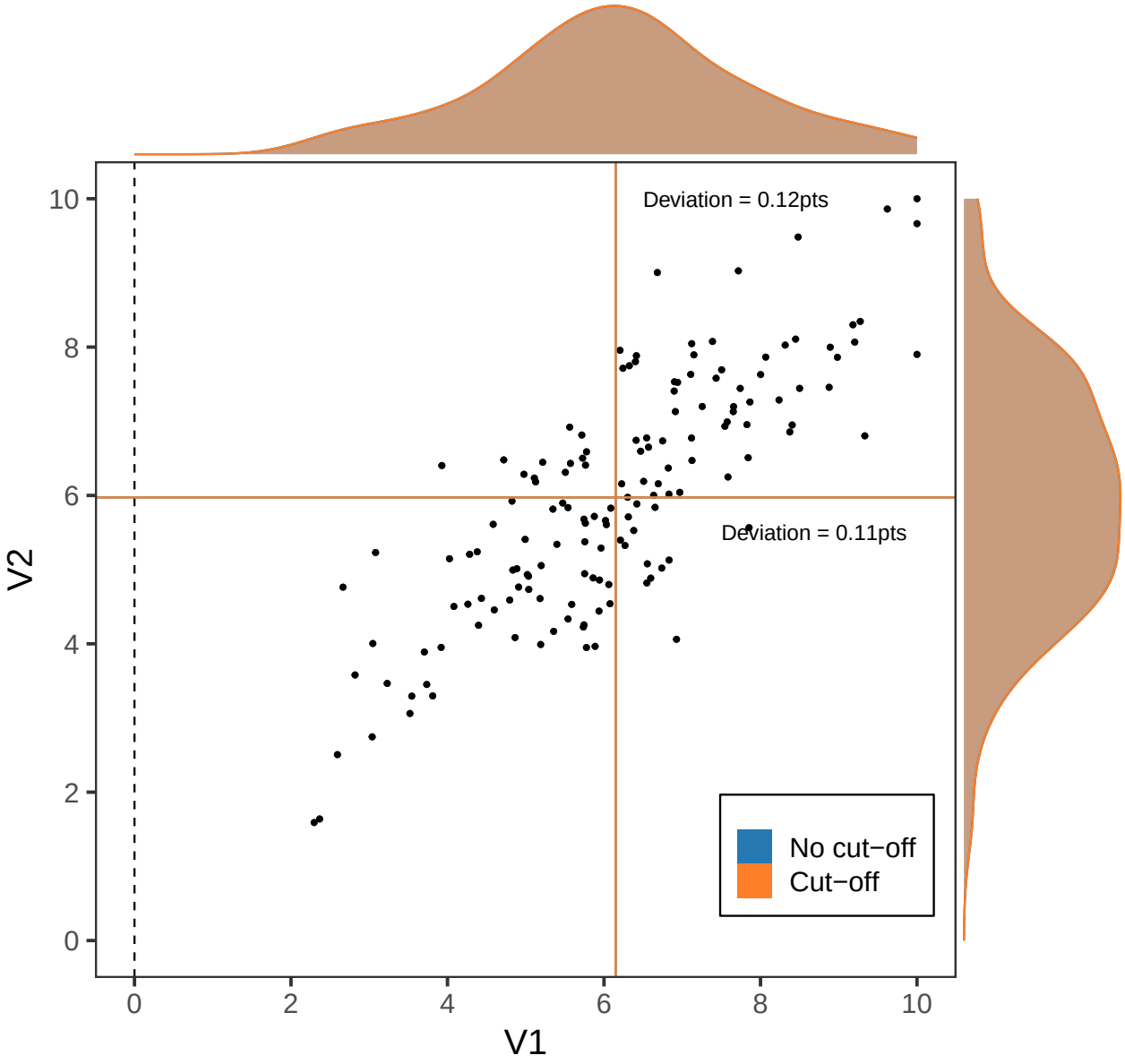
threshold: 0

```
# Process data
placebo_2.0 <- six_2 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

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

```
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	150	0.177	0.95	0.00956	0.341

```
# Plot the data
ggMarginal(placebo_2.0[, 1:3] %>%
               bind_rows(six_2) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_2.0$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(six_2$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_2.0$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 0') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
             groupColour = TRUE,
             groupFill = TRUE)
```



threshold: 3

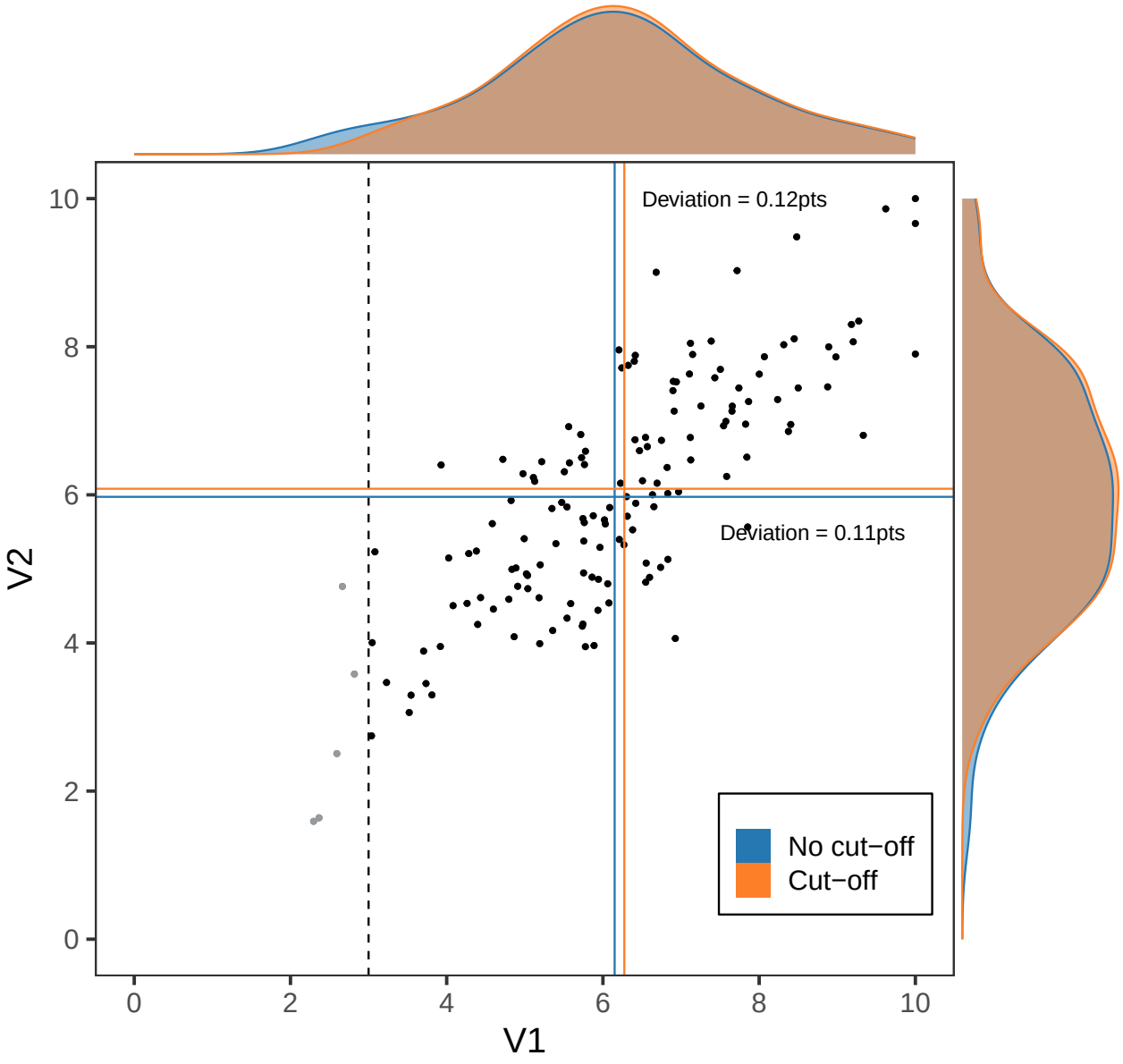
```
# Process data
placebo_2.3 <- six_2 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
```

```
              data = placebo_2.3,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	145	0.192	0.95	0.0321	0.347

```
# Plot the data
ggMarginal(placebo_2.3[, 1:3] %>%
               bind_rows(six_2) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_2.3$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(six_2$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_2.3$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.7, cor = 0.8; Cut-off: 3') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
           groupColour = TRUE,
```

```
    groupFill = TRUE)
```



threshold: 4

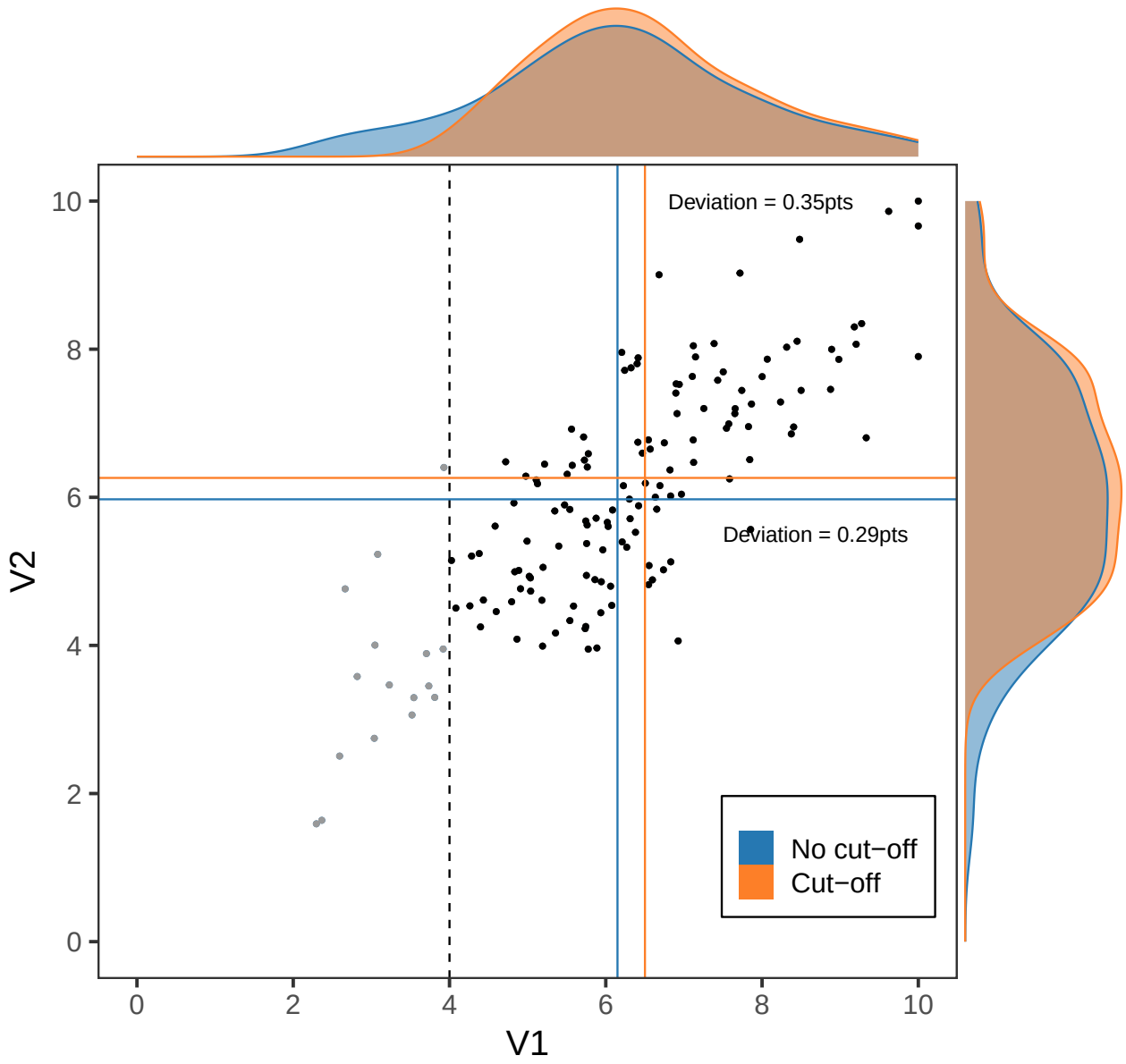
```
# Process that data
placebo_2.4 <- six_2 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)
```

```r
# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_2.4,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	134	0.24	0.95	0.0783	0.404

```r
# Plot the data
ggMarginal(placebo_2.4[, 1:3] %>%
               bind_rows(six_2) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_2.4$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(six_2$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_2.4$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.8, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V1.df$deviation[5], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V2.df$deviation[5], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.7, cor = 0.8; Cut-off: 4') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
```

```
            plot.subtitle = element_text(size = 16)),
    groupColour = TRUE,
    groupFill = TRUE)
```



threshold: 5

```
# Process that data
placebo_2.5 <- six_2 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
```

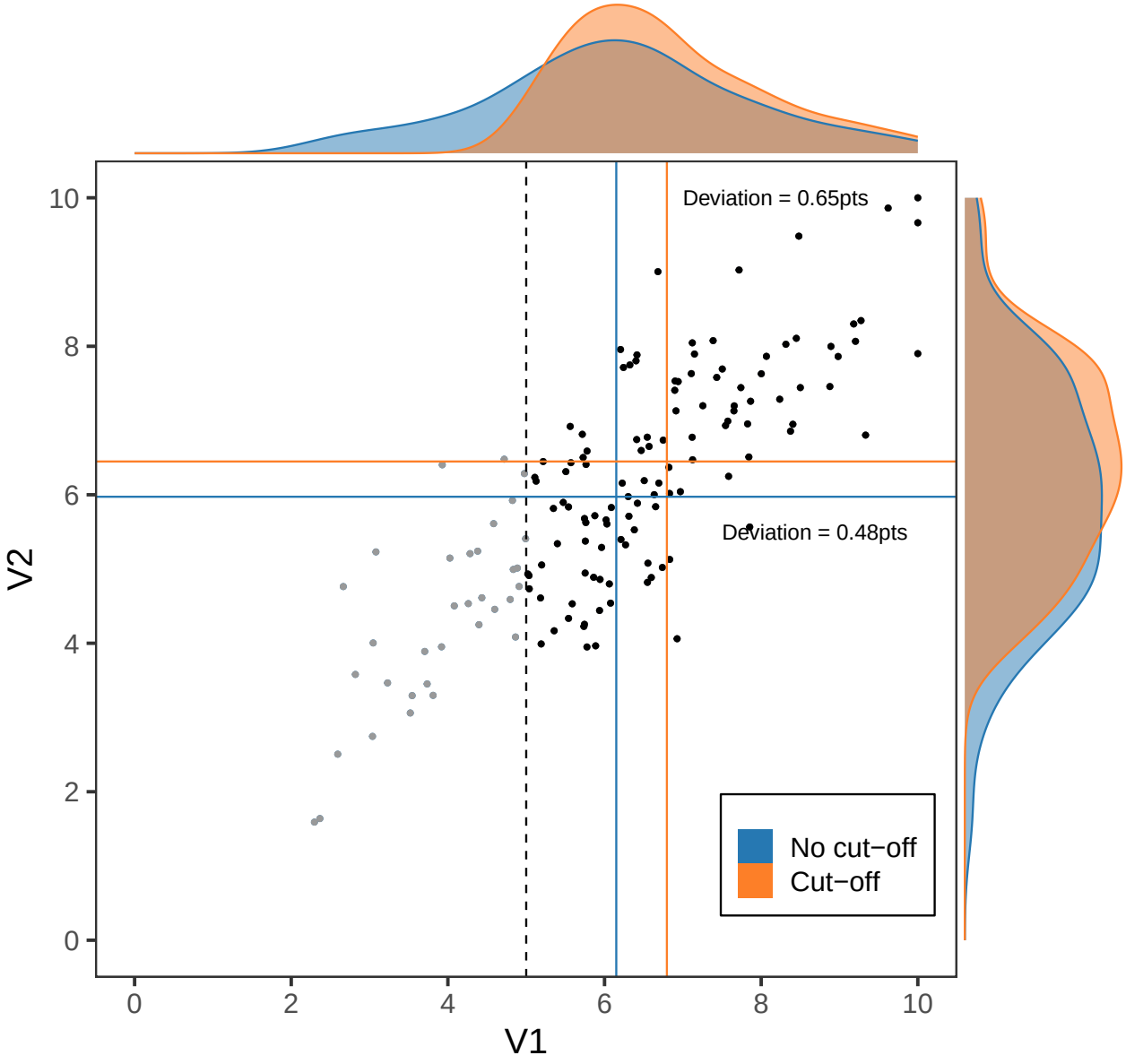
```r
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_2.5,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	116	0.348	0.95	0.175	0.514

```r
# Plot the data
ggMarginal(placebo_2.5[, 1:3] %>%
               bind_rows(six_2) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_2.5$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(six_2$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_2.5$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 7, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V1.df$deviation[6], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_2V2.df$deviation[6], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.7, cor = 0.8; Cut-off: 5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
```

```
                                                size = 0.5),
          panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16)),
groupColour = TRUE,
groupFill = TRUE)
```

Base model: Mean = 6.2, SD = 1.7, cor = 0.8; Cut−off: 5

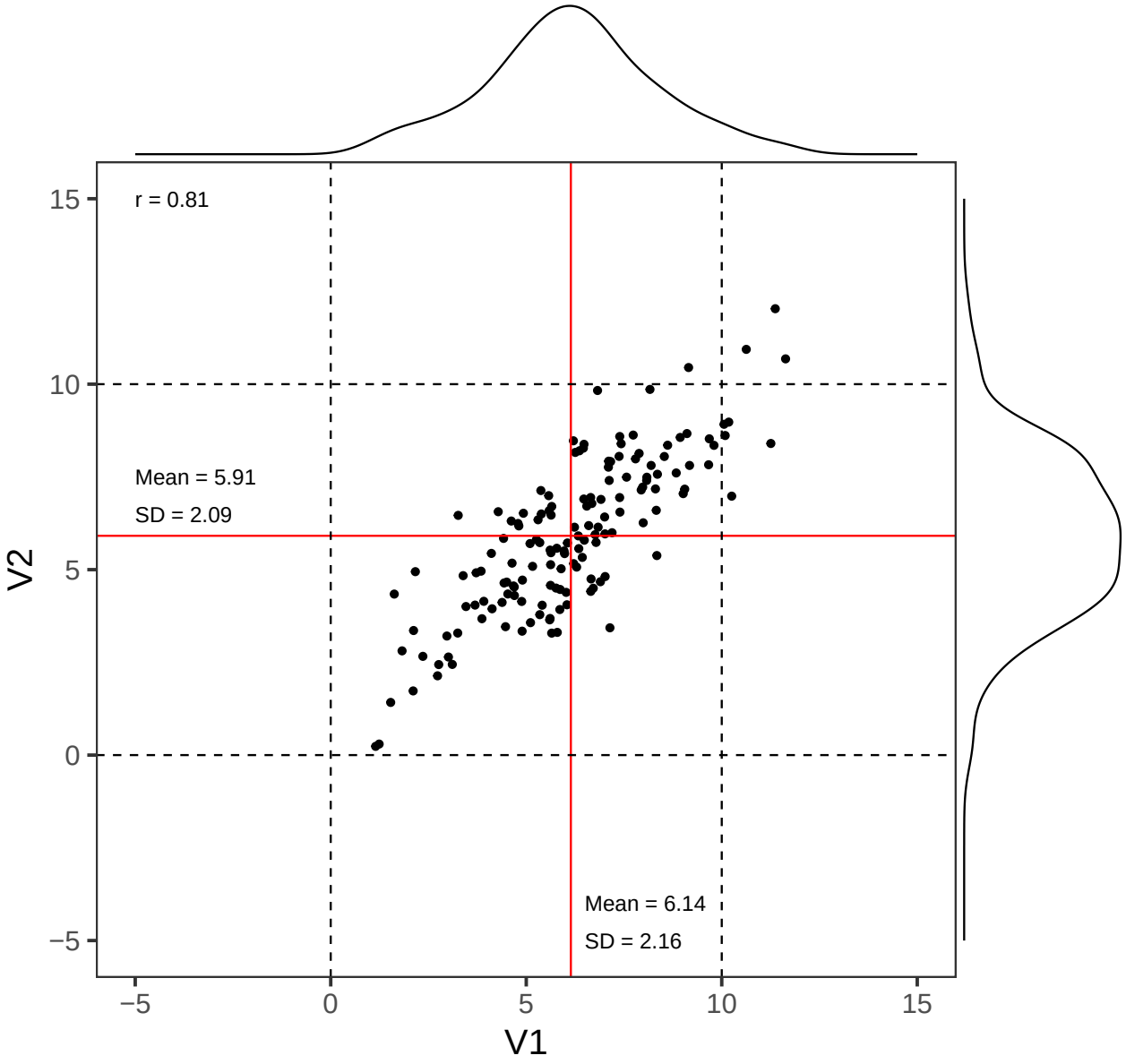


Mean = 6.2, SD = 2.2, cor = 0.8

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 = 150, 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$V2), colour = 'red') +
               geom_vline(xintercept = mean(six_3.base$V1), colour = 'red') +
               geom_hline(yintercept = 0, linetype = 2) +
               geom_hline(yintercept = 10, linetype = 2) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_vline(xintercept = 10, linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0,
                        label = str_glue("r = {round(cor(six_3.base$V1, six_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_3.base$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 6.5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_3.base$V2),2)}")) +
               annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_3.base$V1), 2)}")) +
               annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_3.base$V1), 2)}")) +
               labs(subtitle = 'Mean = 6.2, SD = 2.2, cor = 0.8') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)))
```



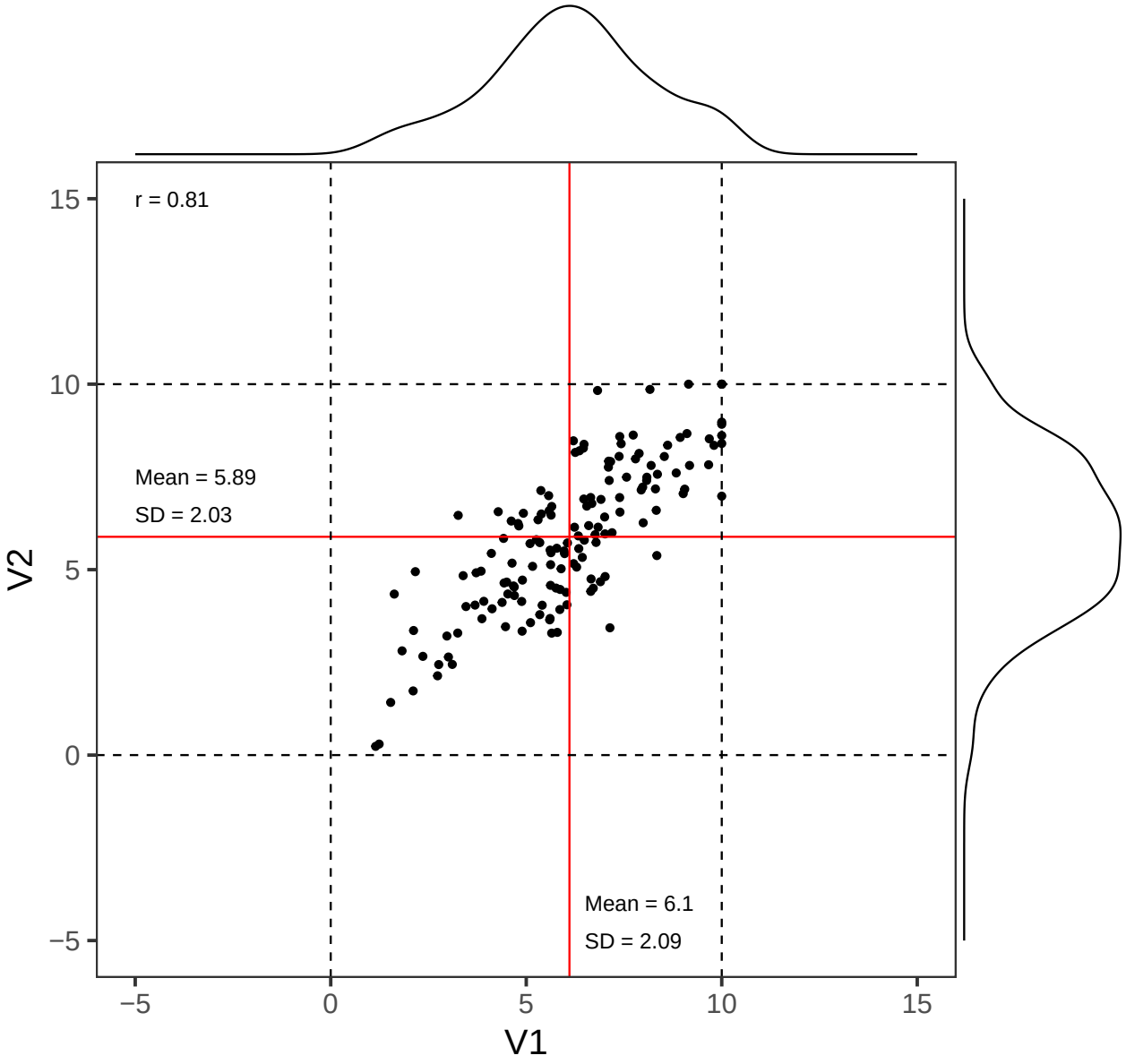

```
# Linear regression
summary(lm(V2 ~ V1, data = six_3.base))
```

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_3.base)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2657 -0.8118 -0.1148  0.8302  3.3818
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.09167    0.30196   3.615 0.000411 ***
## V1           0.78499    0.04639  16.920  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.225 on 148 degrees of freedom
## Multiple R-squared:  0.6592, Adjusted R-squared:  0.6569
## F-statistic: 286.3 on 1 and 148 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 cut-off')

# Plot processed data
ggMarginal(ggplot(data = six_3) +
               aes(x = V1, y = V2) +
               geom_point() +
               geom_hline(yintercept = mean(six_3$V2), colour = 'red') +
               geom_vline(xintercept = mean(six_3$V1), colour = 'red') +
               geom_hline(yintercept = 0, linetype = 2) +
               geom_hline(yintercept = 10, linetype = 2) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_vline(xintercept = 10, linetype = 2) +
               annotate(geom = 'text', x = -5, y = 15, hjust = 0,
                        label = str_glue("r = {round(cor(six_3$V1, six_3$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 7.5, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_3$V2), 2)}")) +
               annotate(geom = 'text', x = -5, y = 6.5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_3$V2),2)}")) +
               annotate(geom = 'text', x = 6.5, y = -4, hjust = 0,
                        label = str_glue("Mean = {round(mean(six_3$V1), 2)}")) +
               annotate(geom = 'text', x = 6.5, y = -5, hjust = 0,
                        label = str_glue("SD = {round(sd(six_3$V1), 2)}")) +
               labs(subtitle = 'Mean = 6.2, SD = 2.2 (constrained), cor = 0.8') +
               scale_y_continuous(limits = c(-5, 15)) +
               scale_x_continuous(limits = c(-5, 15)) +
               theme(panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)))
```




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

```
##
## Call:
## lm(formula = V2 ~ V1, data = six_3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2656 -0.8118 -0.1049  0.8494  3.3815
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.10066    0.30475   3.612 0.000416 ***
## V1           0.78372    0.04726  16.585  < 2e-16 ***
```

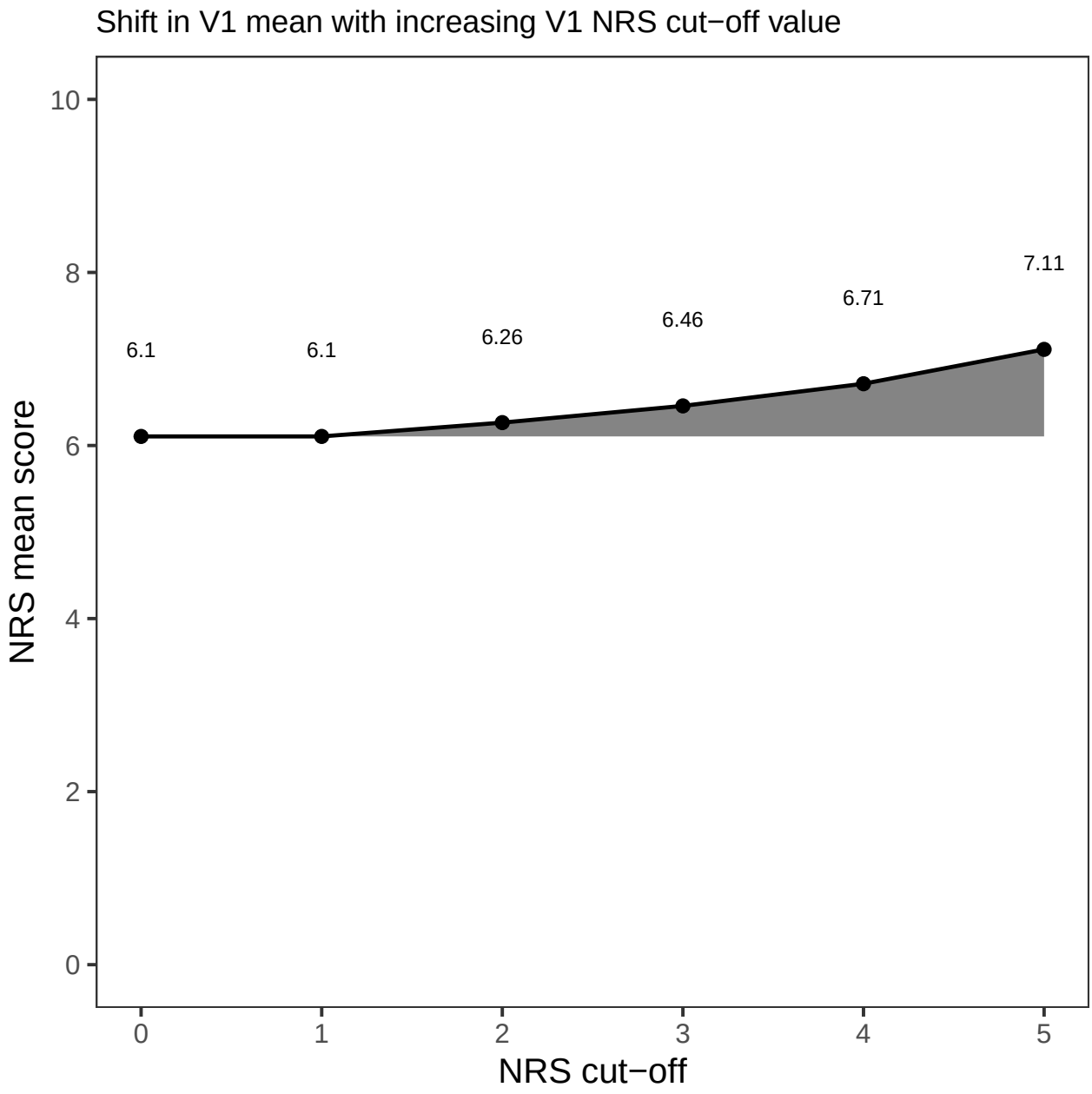
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.203 on 148 degrees of freedom
## Multiple R-squared:  0.6502, Adjusted R-squared:  0.6478
## F-statistic: 275.1 on 1 and 148 DF,  p-value: < 2.2e-16
```

Model mean of V1 with increasing V1 cut-offs from 0 to 5

```
# Extract visit 1 data
six_3V1 <- six_3$V1

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

# Generate a vector of V1 means at each V1 cut-off
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 6.104899 0.0000000
## 2      1 6.104899 0.0000000
## 3      2 6.264592 0.1596930
## 4      3 6.457630 0.3527316
## 5      4 6.713726 0.6088268
## 6      5 7.110804 1.0059052
```

```
# 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 = 3) +
    geom_line(size = 1) +
    geom_text(aes(label = round(mean, 2)), nudge_y = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V1 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



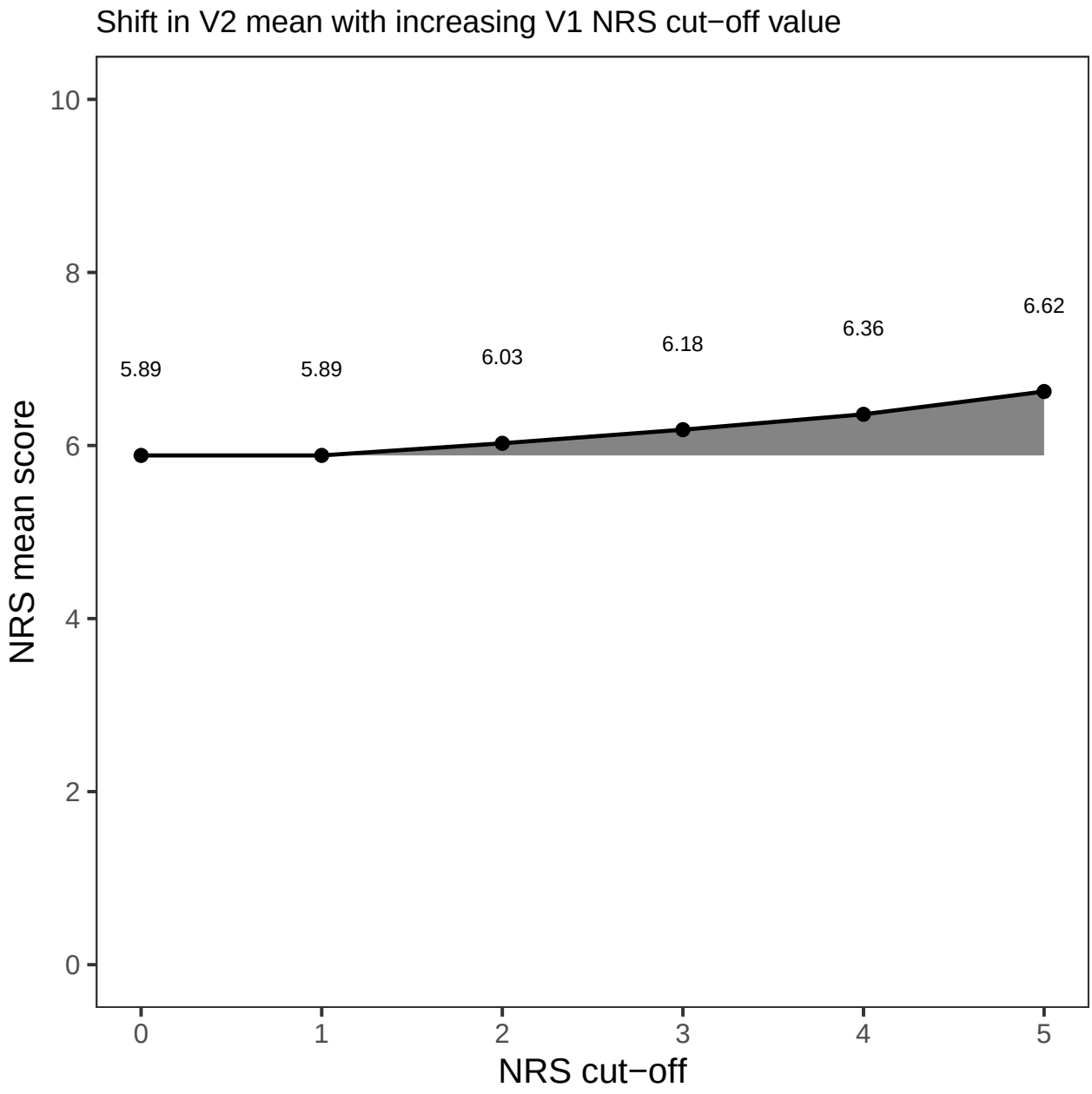
Model mean of V2 with increasing V1 cut-offs from 0 to 5

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

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

# Generate a vector of V2 means at each V1 cut-off
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 5.885219 0.0000000
## 2      1 5.885219 0.0000000
## 3      2 6.025378 0.1401593
## 4      3 6.182584 0.2973654
## 5      4 6.360414 0.4751955
## 6      5 6.623728 0.7385090
```

```
# 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 = 1) +
    scale_y_continuous(limits = c(0, 10),
                       breaks = c(0, 2, 4, 6, 8, 10)) +
    labs(subtitle = 'Shift in V2 mean with increasing V1 NRS cut-off value',
         x = 'NRS cut-off',
         y = 'NRS mean score') +
    theme(panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16))
```



Placebo response

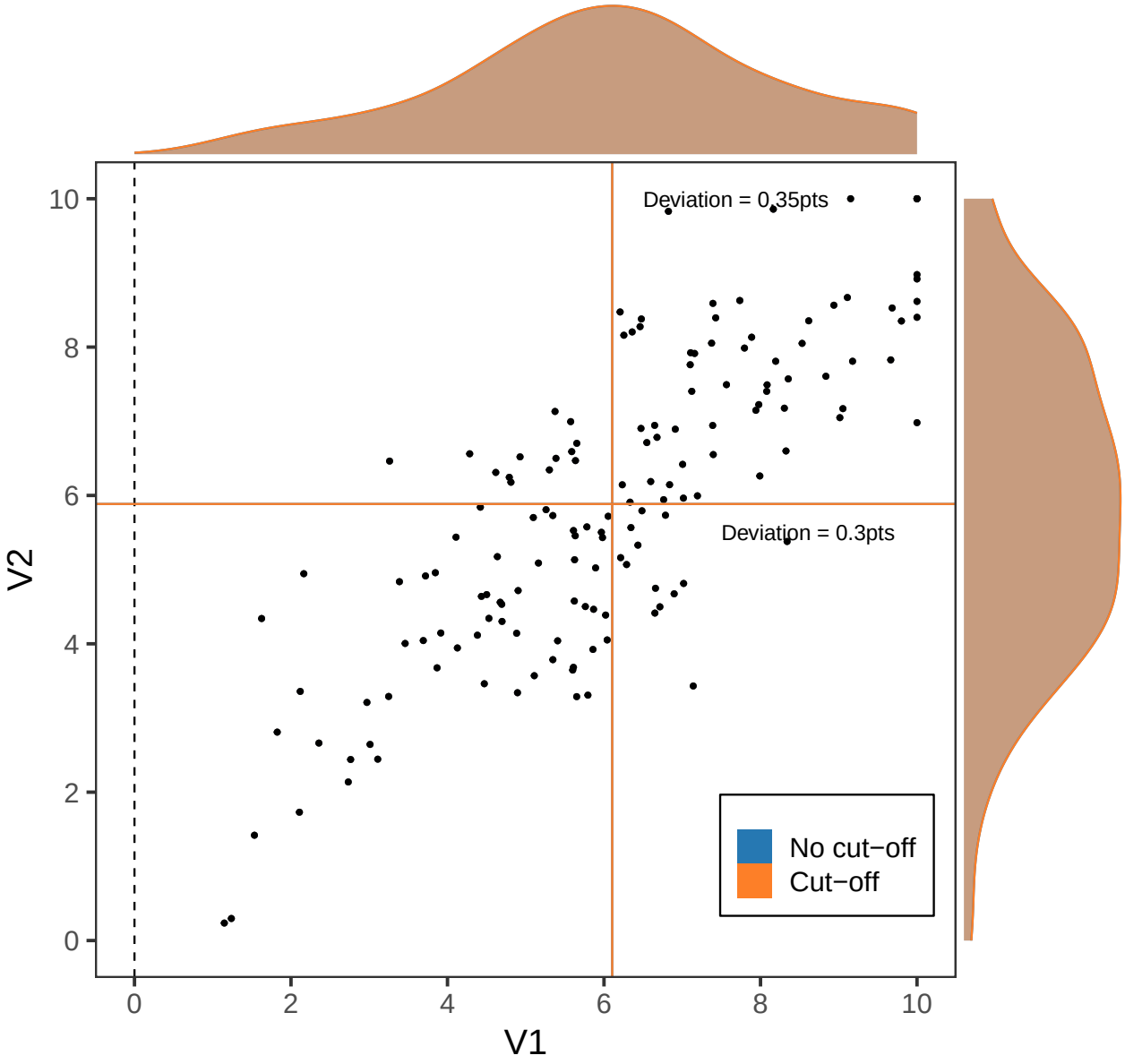
threshold: 0

```
# Process data
placebo_3.0 <- six_3 %>%
    filter(V1 >= 0) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

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

```
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	150	0.22	0.95	0.00458	0.428

```
# Plot the data
ggMarginal(placebo_3.0[, 1:3] %>%
               bind_rows(six_3) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_3.0$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 0, linetype = 2) +
               geom_hline(yintercept = mean(six_3$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_3.0$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 1.2, cor = 0.8; Cut-off: 0') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
           groupColour = TRUE,
           groupFill = TRUE)
```



threshold: 3

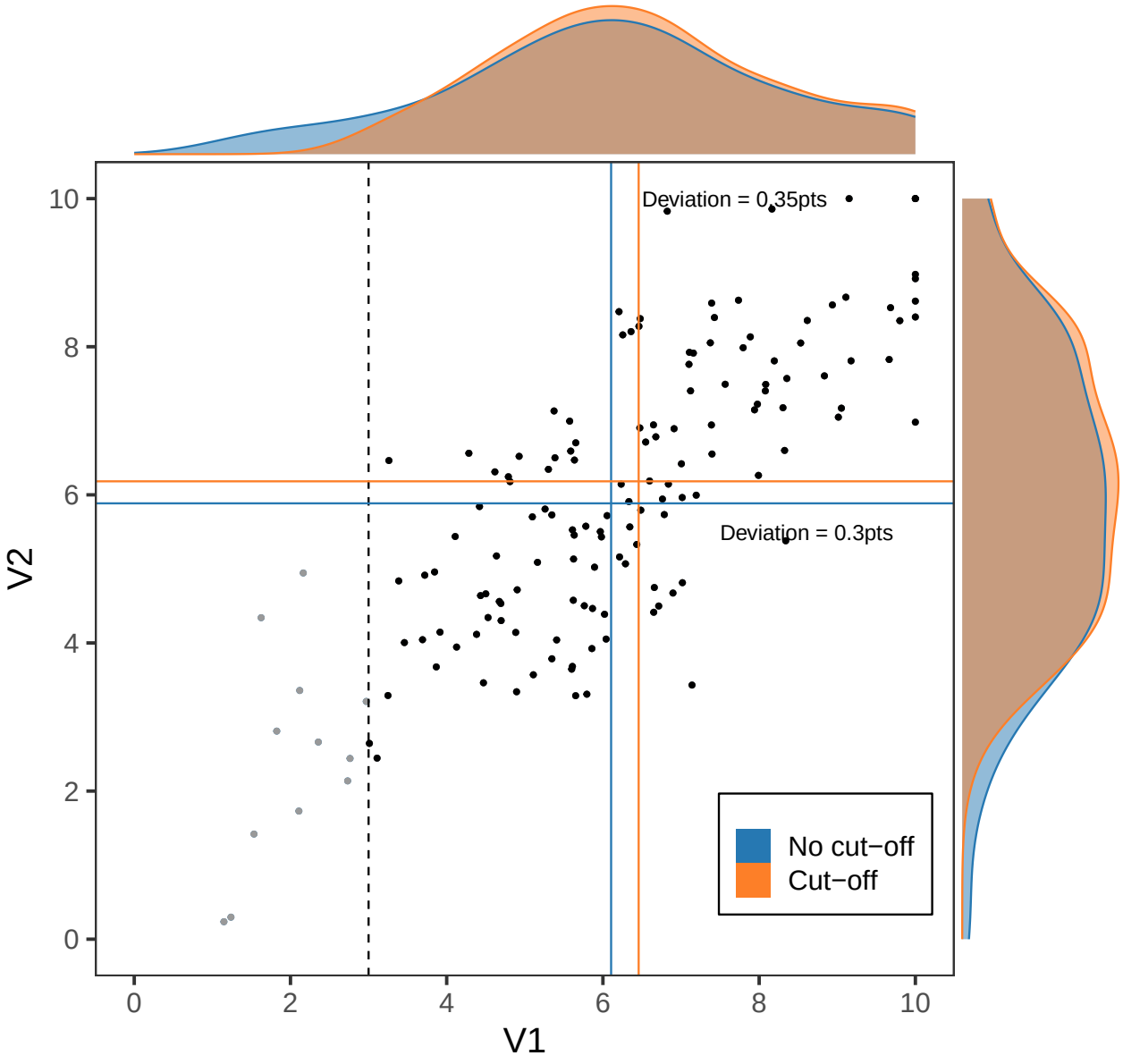
```
# Process data
placebo_3.3 <- six_3 %>%
    filter(V1 >= 3) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
```

```
              data = placebo_3.3,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	138	0.275	0.95	0.0533	0.485

```
# Plot the data
ggMarginal(placebo_3.3[, 1:3] %>%
               bind_rows(six_3) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_3.3$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 3, linetype = 2) +
               geom_hline(yintercept = mean(six_3$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_3.3$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.5, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V1.df$deviation[4], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V2.df$deviation[4], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 2.2, cor = 0.8; Cut-off: 3') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
                     plot.subtitle = element_text(size = 16)),
           groupColour = TRUE,
```

```
    groupFill = TRUE)
```



threshold: 4

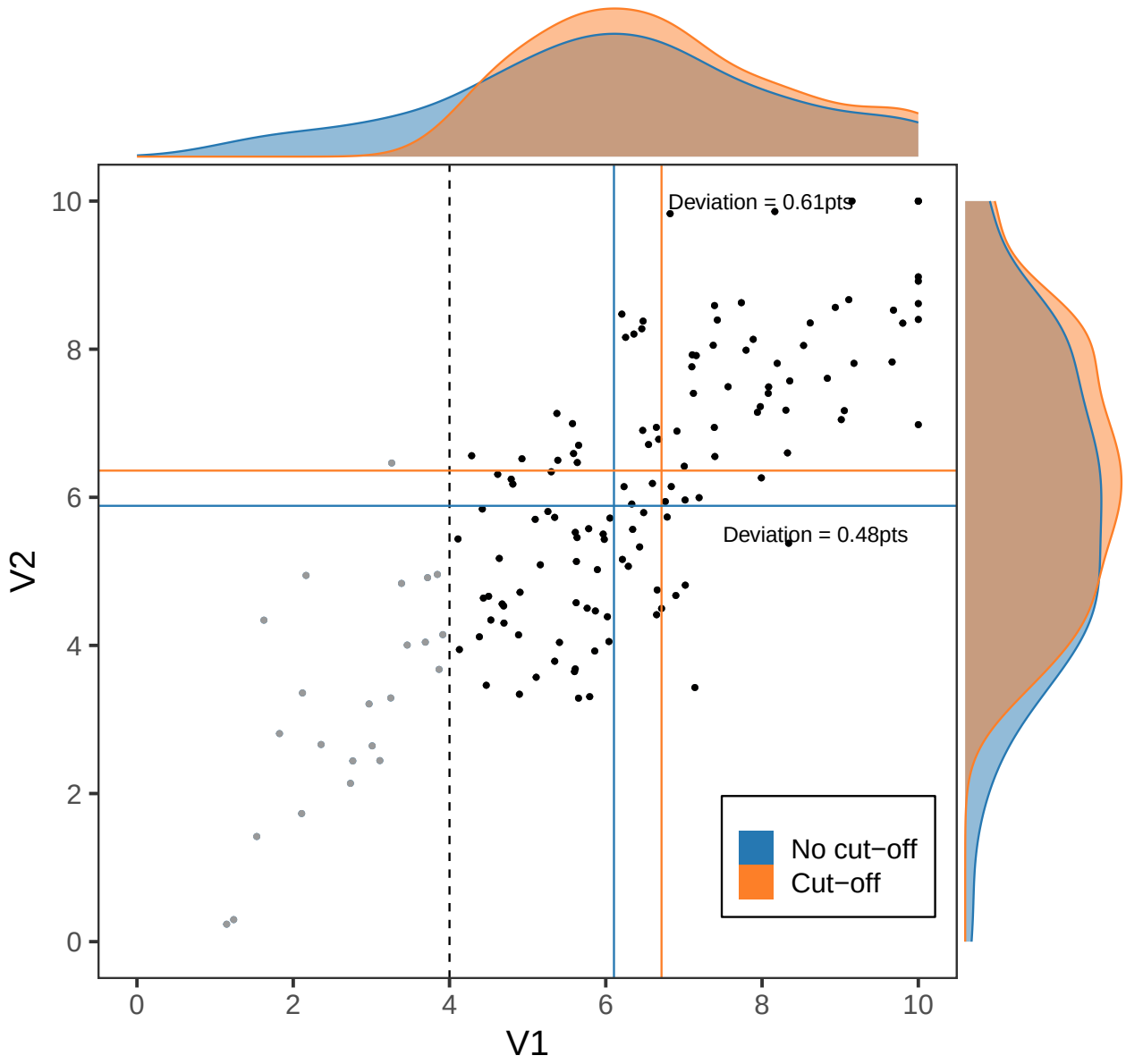
```
# Process that data
placebo_3.4 <- six_3 %>%
    filter(V1 >= 4) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
set.seed(2019)
```

```r
# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_3.4,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	127	0.353	0.95	0.132	0.572

```r
# Plot the data
ggMarginal(placebo_3.4[, 1:3] %>%
               bind_rows(six_3) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_3.4$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 4, linetype = 2) +
               geom_hline(yintercept = mean(six_3$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_3.4$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 6.8, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V1.df$deviation[5], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V2.df$deviation[5], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 2.2, cor = 0.8; Cut-off: 4') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
                                                      size = 0.5),
                     panel.grid = element_blank(),
```

```
            plot.subtitle = element_text(size = 16)),
    groupColour = TRUE,
    groupFill = TRUE)
```



threshold: 5

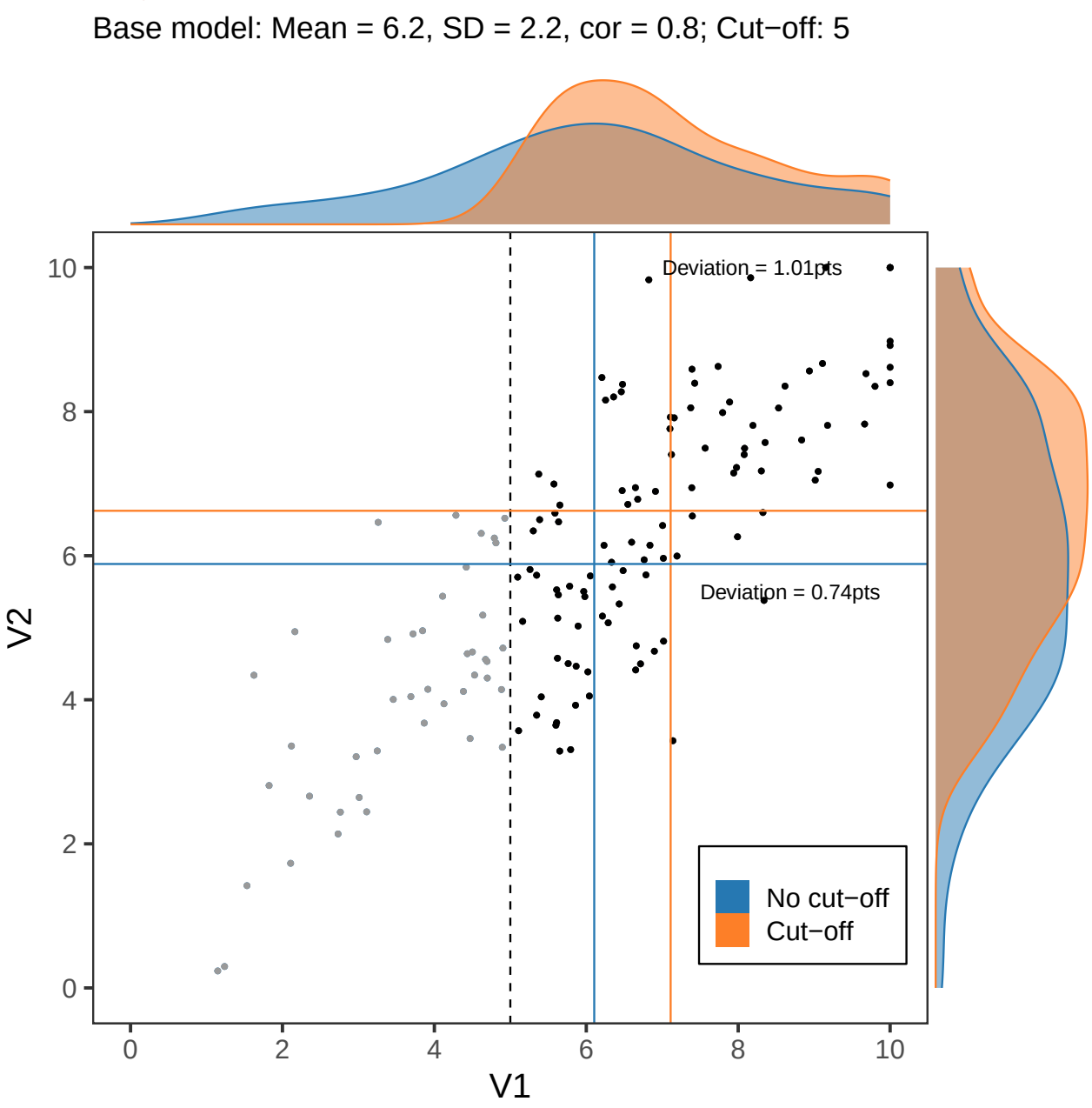
```
# Process that data
placebo_3.5 <- six_3 %>%
    filter(V1 >= 5) %>%
    mutate(difference = V1 - V2) %>%
    mutate(group = 'Cut-off')

# Set seed
```

```r
set.seed(2019)

# Calculate the mean (95%CI) difference between V1 and V2
groupwiseMean(difference ~ 1,
              data = placebo_3.5,
              R = 5000,
              traditional = FALSE,
              bca = TRUE) %>%
  kable(.)
```

.id	n	Mean	Conf.level	Bca.lower	Bca.upper
NA	107	0.487	0.95	0.237	0.726

```r
# Plot the data
ggMarginal(placebo_3.5[, 1:3] %>%
               bind_rows(six_3) %>%
               mutate(group = factor(group,
                                     levels = c('No cut-off', 'Cut-off'),
                                     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$V1),
                          colour = pal[1]) +
               geom_vline(xintercept = mean(placebo_3.5$V1),
                          colour = pal[2]) +
               geom_vline(xintercept = 5, linetype = 2) +
               geom_hline(yintercept = mean(six_3$V2),
                          colour = pal[1]) +
               geom_hline(yintercept = mean(placebo_3.5$V2),
                          colour = pal[2]) +
               annotate(geom = 'text', x = 7, y = 10, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V1.df$deviation[6], 2)}pts")) +
               annotate(geom = 'text', x = 7.5, y = 5.5, hjust = 0,
                        label = str_glue("Deviation = {round(six_3V2.df$deviation[6], 2)}pts")) +
               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(subtitle = 'Base model: Mean = 6.2, SD = 2.2, cor = 0.8; Cut-off: 5') +
               theme(legend.title = element_blank(),
                     legend.position = c(0.85, 0.15),
                     legend.background = element_rect(colour = '#000000',
```

```
                                                    size = 0.5),
          panel.grid = element_blank(),
          plot.subtitle = element_text(size = 16)),
groupColour = TRUE,
groupFill = TRUE)
```



Session information

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
```

```
## Running under: macOS Mojave 10.14.5
##
## 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] knitr_1.23       MBESS_4.6.0      ggExtra_0.8      rcompanion_2.2.2
##  [5] MASS_7.3-51.4    magrittr_1.5     forcats_0.4.0    stringr_1.4.0
##  [9] dplyr_0.8.3      purrr_0.3.2      readr_1.3.1      tidyr_0.8.3.9000
## [13] tibble_2.1.3     ggplot2_3.2.0    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-140       matrixStats_0.54.0 lubridate_1.7.4
##  [4] httr_1.4.0         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_1.2.1         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_0.8-71     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_0.1.4       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
```