Baseline calculations

Peter Kamerman

Last knitted: 10 August 2019

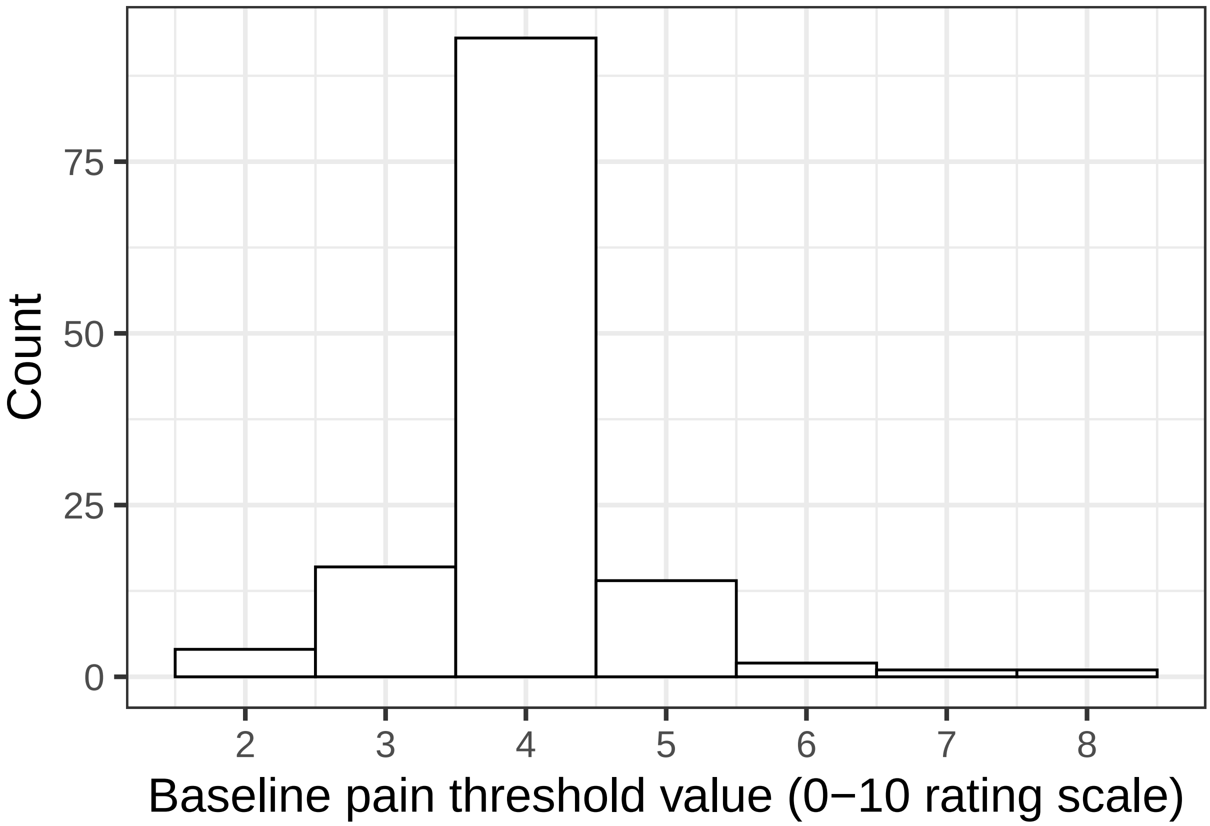
Contents

Pain intensity threshold values

- Data source: Finnerup et al 2015, Appendix 4.
- All numeric values used (after removing duplicates).
- All values converted to a values out of 10.

```
cut_off <- c(5, 4, 4, 2, 4, 3, 4, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
             4, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
             4, 4, 4, 4, 4, 4, 4, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 3, 4,
             4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 4, 4, 4, 3, 5, 4, 4, 4, 5, 5,
             4, 4, 5, 3, 4, 4, 4, 5, 4, 3, 3, 3, 3, 3, 3, 3, 4, 7, 5, 4, 3, 2,
             3, 4, 4, 2.5, 2.5, 4, 4, 4, 4, 5, 4, 4, 8, 4, 4, 4, 5, 3, 4, 4, 4)

ggplot(data = data.frame(cutoff = cut_off)) +
    aes(cutoff) +
    geom_histogram(binwidth = 1,
                   fill = '#FFFFFF',
                   colour = '#000000') +
    labs(x = 'Baseline pain threshold value (0-10 rating scale)',
         y = 'Count') +
    scale_x_continuous(breaks = 2:8) +
    theme_bw(base_size = 18)
```



```
summary(cut_off)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   4.000   4.015   4.000   8.000
```

```
length(cut_off)
```

```
## [1] 131
```

```
data.frame(cutoff = cut_off) %>%
    group_by(cutoff) %>%
    summarise(count = n(),
              percent = round(100 * (count/length(cut_off)), 1)) %>%
    knitr::kable(., caption = 'Summary of threshold values')
```

Table 1: Summary of threshold values

cutoff	count	percent
2.0	2	1.5
2.5	2	1.5
3.0	16	12.2
4.0	93	71.0
5.0	14	10.7
6.0	2	1.5
7.0	1	0.8
8.0	1	0.8

Explore baseline pain intensity data

Import data

- Data source: Finnerup et al 2015, Appendix 4,

```
data <- read_csv('data/baseline-data.csv')
```

```
## Parsed with column specification:
## cols(
##   ID = col_double(),
##   authors = col_character(),
##   pain_requirement = col_character(),
##   scale = col_character(),
##   included_in_analysis = col_character(),
##   sample_size = col_double(),
##   mean_or_median = col_character(),
##   value = col_double(),
##   sd = col_double(),
##   additional_notes = col_character()
## )
```

Explore data

Median

```
data %>%
    filter(mean_or_median == 'median') %>%
    summarise(max = max(value),
              min = min(value)) %>%
    knitr::kable(., caption = 'Range of median values')
```

Table 2: Range of median values

max	min
8.4	4.7

Mean

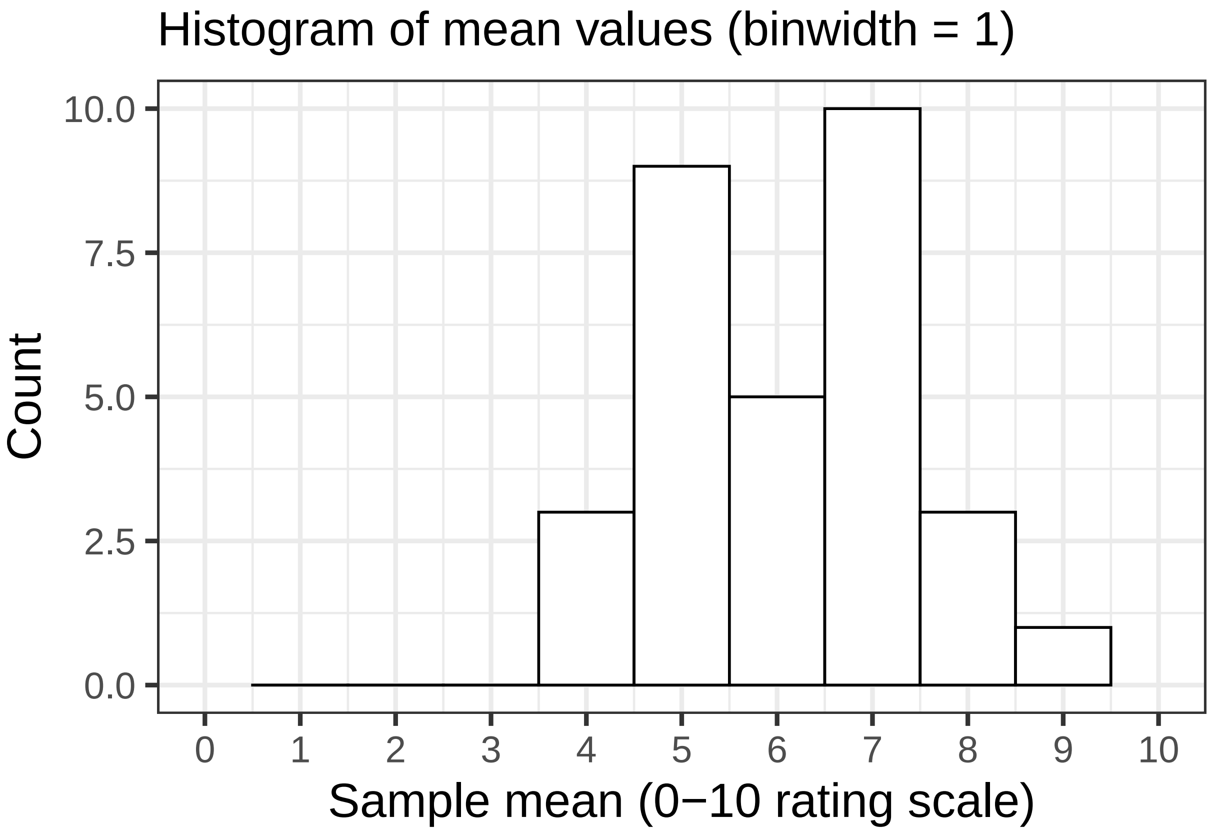
```
data %>%
    filter(mean_or_median == 'mean') %>%
    summarise(max = max(value),
              min = min(value)) %>%
    knitr::kable(., caption = 'Range of mean values')
```

Table 3: Range of mean values

max	min
8.7	3.7

```
data %>%
    filter(mean_or_median == 'mean') %>%
    ggplot(data = .) +
    aes(value) +
    geom_histogram(binwidth = 1,
                   fill = '#FFFFFF',
                   colour = '#000000') +
    labs(subtitle = 'Histogram of mean values (binwidth = 1)',
         x = 'Sample mean (0-10 rating scale)',
         y = 'Count') +
    scale_x_continuous(limits = c(0, 10),
                       breaks = 0:10) +
    theme_bw(base_size = 18)
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



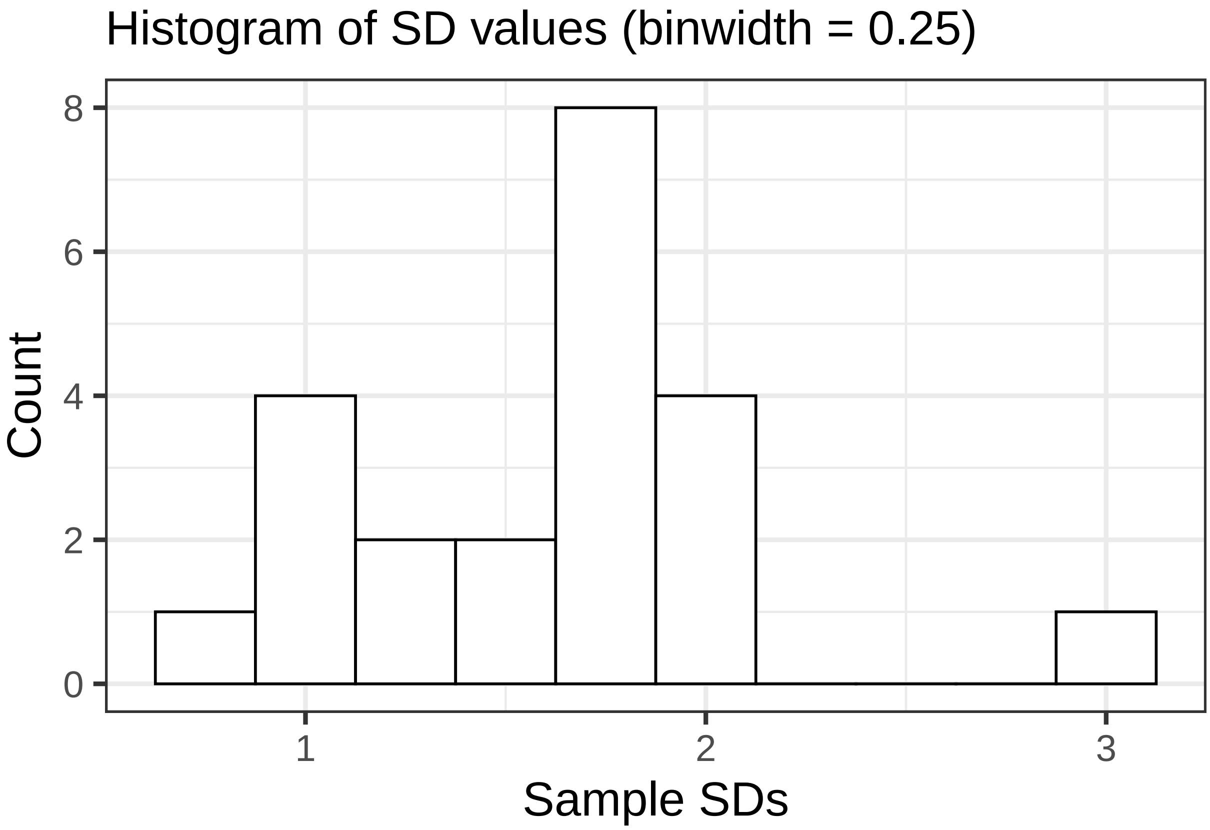
SD

```
data %>%
    filter(!is.na(sd)) %>%
    summarise(max = max(sd),
              min = min(sd)) %>%
    knitr::kable(., caption = 'Range of SD values')
```

Table 4: Range of SD values

max	min
2.9	0.81

```
data %>%
    filter(!is.na(sd)) %>%
    ggplot(data = .) +
    aes(sd) +
    geom_histogram(binwidth = 0.25,
                   fill = '#FFFFFF',
                   colour = '#000000') +
    labs(subtitle = 'Histogram of SD values (binwidth = 0.25)',
         x = 'Sample SDs',
         y = 'Count') +
    theme_bw(base_size = 18)
```



Weighted medians

- Three studies, four median values.
- Hahn_2004(active), Hahn_2004(placebo), Yuen_2002, Low_1995

```
# Process data
median <- data %>%
    filter(included_in_analysis == 'yes') %>%
    filter(mean_or_median == 'median') %>%
    # Calculate total sample size
    mutate(total_sample = sum(sample_size)) %>%
    # Calculate weights
    mutate(weights = sample_size / total_sample)

## Weighted median
weighted.median(x = median$value,
                w = median$weights)
```

```
## [1] 5.07
```

Pooled mean

- Formula sourced from: https://www.ncbi.nlm.nih.gov/books/NBK56512/

```
# Process data
mean <- data %>%
    filter(included_in_analysis == 'yes') %>%
    filter(mean_or_median == 'mean') %>%
    # Multiply mean by the sample size
    mutate(mean_by_n = value * sample_size)

# Pooled mean
sum(mean$mean_by_n) / sum(mean$sample_size)
```

```
## [1] 6.243982
```

Pooled SD

- Formula sourced from: https://www.ncbi.nlm.nih.gov/books/NBK56512/

```
# Process data
sd <- data %>%
    filter(included_in_analysis == 'yes') %>%
    filter(mean_or_median == 'mean') %>%
    filter(!is.na(sd)) %>%
    # Square the SD
    mutate(SD_squared = sd^2) %>%
    # Calculate sample size -1
    mutate(n_minus_1 = sample_size - 1) %>%
    # Get the number of groups
    mutate(k = length(unique(.$authors))) %>%
    # Calculate (n-1)SD^2
    mutate(numerator = SD_squared * n_minus_1)

# Pooled SD
sqrt(sum(sd$numerator) / (sum(sd$sample_size) - sd$k[[1]]))
```

```
## [1] 1.665597
```

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] spatstat_1.60-1     rpart_4.1-15        nlme_3.1-141
##  [4] spatstat.data_1.4-0 forcats_0.4.0       stringr_1.4.0
##  [7] dplyr_0.8.3         purrr_0.3.2         readr_1.3.1
## [10] tidyr_0.8.3.9000    tibble_2.1.3        ggplot2_3.2.0
## [13] tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5      xfun_0.8              splines_3.6.0
##  [4] haven_2.1.1           lattice_0.20-38       colorspace_1.4-1
##  [7] vctrs_0.2.0           generics_0.0.2        spatstat.utils_1.13-0
## [10] htmltools_0.3.6       mgcv_1.8-28           yaml_2.2.0
## [13] rlang_0.4.0           pillar_1.4.2          glue_1.3.1
## [16] withr_2.1.2.9000      modelr_0.1.5          readxl_1.3.1
## [19] munsell_0.5.0         gtable_0.3.0          cellranger_1.1.0
## [22] rvest_0.3.4           evaluate_0.14         labeling_0.3
## [25] knitr_1.24            highr_0.8             broom_0.5.2
## [28] Rcpp_1.0.2            tensor_1.5            scales_1.0.0
## [31] backports_1.1.4       jsonlite_1.6          abind_1.4-5
## [34] deldir_0.1-23         hms_0.5.0             digest_0.6.20
## [37] stringi_1.4.3         polyclip_1.10-0       grid_3.6.0
## [40] cli_1.1.0             tools_3.6.0           goftest_1.1-1
## [43] magrittr_1.5          lazyeval_0.2.2        crayon_1.3.4
## [46] pkgconfig_2.0.2       zeallot_0.1.0         Matrix_1.2-17
## [49] xml2_1.2.2            lubridate_1.7.4       assertthat_0.2.1
## [52] rmarkdown_1.14        httr_1.4.1            rstudioapi_0.10
## [55] R6_2.4.0              compiler_3.6.0
```