Data analysis

Signal detection of spontaneous medical device reports over time

Ty Stanford et al.

Table of contents

1 Set up

1.1 Packages

```
suppressPackageStartupMessages({
  library("readr")
  library("dplyr")
  library("tidyr")
  library("forcats")
  library("purrr")
  library("furrr")
  library("lubridate") # way to handle dates better than default R way
  library("tictoc")    # measure time elapsed in calcs
  library("ggplot2")
  library("ggrepel")
  library("knitr")
  library("gsDesign")
  library("foreach")
  library("arrow") # read/write parquet files
})
```

```
# NOTE: need to run first (only once, assumes devtools installed):
# devtools::install_github("tystan/pharmsignal")
library("pharmsignal") # signal detection algs

# here are the functions written for these analyses
# they will be shown in the *Appendix A*
source("r/_funcs.R")

### NB: packages required that are used in above sourced file
# Sequential
# EmpiricalCalibration
```

1.2 Parallel compution setup

```
# options(future.globals.maxSize = 500 * 1024 ^ 2) # = 500 MiB
options(future.globals.maxSize = 2e3 * 1024 ^ 2) # = 2 GB



# furrr parallel workers/cores setup
# change `workers = 4` based on cores available in processor being used
(thread_to_use <- parallel::detectCores() - 2) # keep a core = 2 threads free
```

```
[1] 22
```

```
plan(multisession, workers = thread_to_use)

### test parallel works
# test code from https://furrr.futureverse.org/
# sequential
tic()
dev_null <- map(c(2, 2, 2), ~Sys.sleep(.x))
toc() # ~6 sec
```

```
6.11 sec elapsed
```

```
# parallel: should be (roughly, plus overheads) a third of the time of sequential
tic()
dev_null <- future_map(c(2, 2, 2), ~Sys.sleep(.x))
toc() # ~2 sec
```

```
2.86 sec elapsed
```

```
# for fun
tic()
dev_null <- future_map(rep(2, thread_to_use), ~Sys.sleep(.x))
toc()
```

```
7.11 sec elapsed
```

```r
# this only applies to the non-parallel (non-"future") operations
set.seed(1234)
# this seed can be set in future_map() etc for reproducible parallel comp seeds
furrr_seed1 <- furrr_options(seed = 5678)
furrr_seed2 <- furrr_options(seed = 9012)
furrr_seed3 <- furrr_options(seed = 3456)
```

1.3 Constants

```r
# arbitrarily, let's go with minimum cell count of 3 (should be discussed!)
arbitrary_cell_min <- 1
```

1.4 Functions

```r
get_sig_tab <- function(nA, nB, nC, nD, alpha = 0.05, n_mcmc = 1e+05) {

  out_cols_of_interest <- c("est_name", "est_scale", "est", "alpha", "ci_lo", "ci_hi")
  sig_tab <- pharmsignal::bcpnn_mcmc_signal(nA, nB, nC, nD, alpha = alpha, n_mcmc = n_mcmc
  sig_tab <- sig_tab[, out_cols_of_interest]
  # sig_tab <- bind_cols(tibble(mnth = mnth), sig_tab)
  return(sig_tab)

}

get_sig_tab_over_time <- function(dat, alpha = 0.05, n_mcmc = 1e+05) {

  n_tp <- nrow(dat)

  sig_tab_over_time <-
    foreach(i = 1:n_tp, .combine = bind_rows, .packages = "dplyr") %do% {
      with(
        dat,
        get_sig_tab(
          # mnth[i],
          nA[i], nB[i], nC[i], nD[i],
          alpha = alpha, n_mcmc = n_mcmc
        )
      )
    }
```

```
  return(sig_tab_over_time)

}

# same as get_sig_tab_over_time(), however, alpha assumed included as column in data
get_sig_tab_over_time_2 <- function(dat, n_mcmc = 1e+05) {

  n_tp <- nrow(dat)

  sig_tab_over_time <-
    foreach(i = 1:n_tp, .combine = bind_rows, .packages = "dplyr") %do% {
      with(
        dat,
        get_sig_tab(
          # mnth[i],
          nA[i], nB[i], nC[i], nD[i],
          alpha = adj_alpha[i],
          n_mcmc = n_mcmc
        )
      )
    }

  return(sig_tab_over_time)

}
```

1.5 Load data

```
### monthly for testing
sra_dat <- read_parquet("dat/sra_dat.parquet")

### want this
cumul_qtrly_dat <- read_parquet("dat/cumul_qtrly_dat.parquet")

(thresholds <- sort(unique(sra_dat$thresh)))
```

```
 [1] "0.010" "0.015" "0.020" "0.025" "0.030" "0.035" "0.040" "0.045" "0.050"
[10] "0.055" "0.060" "0.065" "0.070" "0.075" "0.080" "0.085" "0.090" "0.095"
[19] "0.100"
```

2 Analysis

2.1 BCPNN

```
sra_cum <-
  cumul_qtrly_dat

# make data for each combination of params nested for purrr like processing
sra_cum <-
  sra_cum %>%
  nest(data = c(mnth, nA, nB, nC, nD))

sra_cum2 <-
  sra_dat %>%
  dplyr::filter(dat_type == "cumulative") %>%
  nest(data = c(mnth, nA, nB, nC, nD))


# testing/example
sra_cum$data[[9]] %>% print(., n = nrow(.))
```

```
# A tibble: 18 x 5
   mnth       nA    nB    nC    nD
   <chr>   <dbl> <dbl> <dbl> <dbl>
 1 2013-Q3     4    12     1    10
 2 2013-Q4     6    14     1    10
 3 2014-Q1     6    14     1    11
 4 2014-Q2     7    15     1    14
 5 2014-Q3     9    17     3    21
 6 2014-Q4    26    19     4    27
 7 2015-Q1    27    19     4    28
 8 2015-Q2    27    19     4    28
 9 2015-Q3    27    20     4    28
10 2015-Q4    27    20     6    28
11 2016-Q1    30    21     6    28
12 2016-Q2    34    21     6    28
13 2016-Q3    34    21     7    33
14 2016-Q4    36    23     7    33
15 2017-Q1    45    23     8    34
16 2017-Q2    58    24     8    37
17 2017-Q3    68    24     8    38
```

```
18 2017-Q4    77    25     8    38
```

```
sra_cum2$data[[9]] %>% print(., n = nrow(.))
```

```
# A tibble: 38 x 5
   mnth       nA    nB    nC    nD
   <chr>   <dbl> <dbl> <dbl> <dbl>
 1 2013-09     4    12     1    10
 2 2013-11     6    13     1    10
 3 2013-12     6    14     1    10
 4 2014-03     6    14     1    11
 5 2014-04     6    14     1    12
 6 2014-05     7    15     1    13
 7 2014-06     7    15     1    14
 8 2014-07     7    15     2    15
 9 2014-08     9    17     2    19
10 2014-09     9    17     3    21
11 2014-10    10    18     3    24
12 2014-11    10    18     3    26
13 2014-12    26    19     4    27
14 2015-01    26    19     4    28
15 2015-03    27    19     4    28
16 2015-09    27    20     4    28
17 2015-10    27    20     5    28
18 2015-11    27    20     6    28
19 2016-01    29    20     6    28
20 2016-03    30    21     6    28
21 2016-04    34    21     6    28
22 2016-07    34    21     6    30
23 2016-08    34    21     7    32
24 2016-09    34    21     7    33
25 2016-11    36    22     7    33
26 2016-12    36    23     7    33
27 2017-01    40    23     7    33
28 2017-02    43    23     7    33
29 2017-03    45    23     8    34
30 2017-04    50    24     8    36
31 2017-05    54    24     8    37
32 2017-06    58    24     8    37
33 2017-07    60    24     8    38
34 2017-08    66    24     8    38
```

```
35 2017-09    68    24     8    38
36 2017-10    71    24     8    38
37 2017-11    75    25     8    38
38 2017-12    77    25     8    38
```

```
get_sig_tab_over_time(sra_cum$data[[9]])
```

```
     est_name est_scale       est alpha         ci_lo     ci_hi
1  bcpnn_mcmc      log2 0.3778604  0.05 -4.808200e-01 0.9400987
2  bcpnn_mcmc      log2 0.3738316  0.05 -1.863521e-01 0.8244052
3  bcpnn_mcmc      log2 0.4150070  0.05 -1.636526e-01 0.8790137
4  bcpnn_mcmc      log2 0.5127001  0.05  9.976021e-05 0.9582053
5  bcpnn_mcmc      log2 0.4951774  0.05 -3.518268e-02 0.9394421
6  bcpnn_mcmc      log2 0.5370558  0.05  2.980089e-01 0.8130051
7  bcpnn_mcmc      log2 0.5500767  0.05  3.141467e-01 0.8210319
8  bcpnn_mcmc      log2 0.5500767  0.05  3.134158e-01 0.8247474
9  bcpnn_mcmc      log2 0.5377636  0.05  3.037482e-01 0.8069992
10 bcpnn_mcmc      log2 0.4850381  0.05  2.424461e-01 0.7501966
11 bcpnn_mcmc      log2 0.4647627  0.05  2.433795e-01 0.7129193
12 bcpnn_mcmc      log2 0.4520794  0.05  2.497069e-01 0.6851919
13 bcpnn_mcmc      log2 0.5093946  0.05  2.943064e-01 0.7509517
14 bcpnn_mcmc      log2 0.4823956  0.05  2.776336e-01 0.7121208
15 bcpnn_mcmc      log2 0.4519206  0.05  2.750012e-01 0.6554113
16 bcpnn_mcmc      log2 0.4402744  0.05  2.869655e-01 0.6207770
17 bcpnn_mcmc      log2 0.4208998  0.05  2.823981e-01 0.5835677
18 bcpnn_mcmc      log2 0.3915001  0.05  2.644567e-01 0.5431420
```

```
### for i5-8400/48GB 2133mhz memory
# takes ~ 90 sec for monthly
# takes ~ 40 sec for quarterly
### divide by a fair bit for r9-7900X
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    sig_tab =
      future_map(
        .x = data,
        .f = get_sig_tab_over_time,
        .options = furrr_seed1
      )
```

```
  )
toc()
```

```
13.31 sec elapsed
```

```
# check
sra_cum$sig_tab[[9]]
```

```
     est_name est_scale       est alpha         ci_lo     ci_hi
1  bcpnn_mcmc      log2 0.3778604  0.05 -0.471319571 0.9398523
2  bcpnn_mcmc      log2 0.3738316  0.05 -0.194902215 0.8192987
3  bcpnn_mcmc      log2 0.4150070  0.05 -0.159141619 0.8762078
4  bcpnn_mcmc      log2 0.5127001  0.05  0.005598686 0.9576749
5  bcpnn_mcmc      log2 0.4951774  0.05 -0.035298818 0.9393969
6  bcpnn_mcmc      log2 0.5370558  0.05  0.297340028 0.8106654
7  bcpnn_mcmc      log2 0.5500767  0.05  0.313552382 0.8228209
8  bcpnn_mcmc      log2 0.5500767  0.05  0.314345465 0.8205779
9  bcpnn_mcmc      log2 0.5377636  0.05  0.302385156 0.8047330
10 bcpnn_mcmc      log2 0.4850381  0.05  0.244954289 0.7529505
11 bcpnn_mcmc      log2 0.4647627  0.05  0.241129941 0.7124753
12 bcpnn_mcmc      log2 0.4520794  0.05  0.248667861 0.6845899
13 bcpnn_mcmc      log2 0.5093946  0.05  0.297408421 0.7511102
14 bcpnn_mcmc      log2 0.4823956  0.05  0.279390378 0.7111820
15 bcpnn_mcmc      log2 0.4519206  0.05  0.274249192 0.6566626
16 bcpnn_mcmc      log2 0.4402744  0.05  0.286249818 0.6172668
17 bcpnn_mcmc      log2 0.4208998  0.05  0.280996281 0.5862381
18 bcpnn_mcmc      log2 0.3915001  0.05  0.263734467 0.5423336
```

```
sra_cum_bcpnn <-
  sra_cum %>%
  unnest(cols = c(data, sig_tab)) %>%
  mutate(
    # dte = as_date(paste0(mnth, "-01"))
    dte =
      as_date(paste0(
        substr(mnth, 1, 5),
        sprintf("%02.0f", (as.integer(substr(mnth, 7, 7)) - 1) * 3 + 1),
        "-01"
      ))
```

```
  )

sra_cum_bcpnn
```

```
# A tibble: 1,707 x 15
   grps   dat_type thresh mnth     nA    nB    nC    nD est_name est_scale   est
   <chr>  <chr>    <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <chr>    <chr>     <dbl>
 1 (a) p~ cumulat~ 0.010  2013~     3     7     1     4 bcpnn_m~ log2      0.144
 2 (a) p~ cumulat~ 0.010  2013~     4    10     1     5 bcpnn_m~ log2      0.170
 3 (a) p~ cumulat~ 0.010  2013~     5    11     2     9 bcpnn_m~ log2      0.243
 4 (a) p~ cumulat~ 0.010  2013~     9    11     2     9 bcpnn_m~ log2      0.323
 5 (a) p~ cumulat~ 0.010  2014~     9    11     2    10 bcpnn_m~ log2      0.365
 6 (a) p~ cumulat~ 0.010  2014~    10    12     3    12 bcpnn_m~ log2      0.351
 7 (a) p~ cumulat~ 0.010  2014~    12    14     5    19 bcpnn_m~ log2      0.420
 8 (a) p~ cumulat~ 0.010  2014~    30    15     7    24 bcpnn_m~ log2      0.445
 9 (a) p~ cumulat~ 0.010  2015~    31    15     7    25 bcpnn_m~ log2      0.459
10 (a) p~ cumulat~ 0.010  2015~    31    15     7    25 bcpnn_m~ log2      0.459
# i 1,697 more rows
# i 4 more variables: alpha <dbl>, ci_lo <dbl>, ci_hi <dbl>, dte <date>
```

```
# first signif
bcpnn_signif <-
  sra_cum_bcpnn %>%
  group_by(grps, dat_type, thresh) %>%
  dplyr::filter(ci_lo > 0) %>%
  arrange(dte) %>%
  dplyr::filter(row_number() == 1) %>%
  ungroup() %>%
  rename(dte_reach_sig = dte)


nrow(sra_cum_bcpnn)
```

```
[1] 1707
```

```
sra_cum_bcpnn <-
  left_join(
    sra_cum_bcpnn,
    bcpnn_signif %>% select(grps, dat_type, thresh, dte_reach_sig),
```

```
    c("grps", "dat_type", "thresh")
  )
nrow(sra_cum_bcpnn)
```

```
[1] 1707
```

```
sra_cum_bcpnn
```

```
# A tibble: 1,707 x 16
   grps   dat_type thresh mnth     nA    nB    nC    nD est_name est_scale   est
   <chr>  <chr>    <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <chr>    <chr>     <dbl>
 1 (a) p~ cumulat~ 0.010  2013~     3     7     1     4 bcpnn_m~ log2      0.144
 2 (a) p~ cumulat~ 0.010  2013~     4    10     1     5 bcpnn_m~ log2      0.170
 3 (a) p~ cumulat~ 0.010  2013~     5    11     2     9 bcpnn_m~ log2      0.243
 4 (a) p~ cumulat~ 0.010  2013~     9    11     2     9 bcpnn_m~ log2      0.323
 5 (a) p~ cumulat~ 0.010  2014~     9    11     2    10 bcpnn_m~ log2      0.365
 6 (a) p~ cumulat~ 0.010  2014~    10    12     3    12 bcpnn_m~ log2      0.351
 7 (a) p~ cumulat~ 0.010  2014~    12    14     5    19 bcpnn_m~ log2      0.420
 8 (a) p~ cumulat~ 0.010  2014~    30    15     7    24 bcpnn_m~ log2      0.445
 9 (a) p~ cumulat~ 0.010  2015~    31    15     7    25 bcpnn_m~ log2      0.459
10 (a) p~ cumulat~ 0.010  2015~    31    15     7    25 bcpnn_m~ log2      0.459
# i 1,697 more rows
# i 5 more variables: alpha <dbl>, ci_lo <dbl>, ci_hi <dbl>, dte <date>,
#   dte_reach_sig <date>
```

```
sra_cum_bcpnn <-
  sra_cum_bcpnn %>%
  mutate(
    dte_reach_sig = if_else(is.na(dte_reach_sig), as_date(today()), dte_reach_sig),
    reach_sig = dte >= dte_reach_sig
  )
```

2.2 BCPNN with mult comp adjust

2.3 MaxSPRT

```
# sra_cum <-
#   sra_dat %>%
#   dplyr::filter(dat_type == "cumulative")
sra_cum <-
  cumul_qtrly_dat

cv_tab <-
  sra_cum %>%
  # dplyr::filter(thresh < 0.070) %>%
  group_by(grps, thresh) %>%
  summarise(
    min_dte = min(mnth),
    max_dte = max(mnth),
    rows = n(),
    sum_nA = max(nA),
    sum_nC = max(nC),
    tot_n = sum_nA + sum_nC,
    .groups = "drop"
  ) %>%
  mutate(
    # qtrs = interval(paste0(min_dte, "-01"), paste0(max_dte, "-01")) / months(1) / 4,
    qtrs = rows,
    n_per_qtr = tot_n / qtrs,
    z = sum_nC / sum_nA
  )

cv_tab %>%
  kable(., digits = 1)
```

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	z
(a) pelvic_mesh v hernia_mesh	0.010	2013-Q1	2017-Q4	20	82	12	94	20	4.7	0.1
(a) pelvic_mesh v hernia_mesh	0.015	2013-Q1	2017-Q4	20	82	12	94	20	4.7	0.1
(a) pelvic_mesh v hernia_mesh	0.020	2013-Q3	2017-Q4	18	82	10	92	18	5.1	0.1

grps	thresh	min_dtn	max_dtn	rows	sum_nAm	nCt	nqtrs	n_per_qtr
(a) pelvic_mesh v hernia_mesh	0.025	2013-Q3	2017-Q4	18	82	10	92	18	5.1	0.1
(a) pelvic_mesh v hernia_mesh	0.030	2013-Q3	2017-Q4	18	82	10	92	18	5.1	0.1
(a) pelvic_mesh v hernia_mesh	0.035	2013-Q3	2017-Q4	18	82	10	92	18	5.1	0.1
(a) pelvic_mesh v hernia_mesh	0.040	2013-Q3	2017-Q4	18	81	9	90	18	5.0	0.1
(a) pelvic_mesh v hernia_mesh	0.045	2013-Q3	2017-Q4	18	79	8	87	18	4.8	0.1
(a) pelvic_mesh v hernia_mesh	0.050	2013-Q3	2017-Q4	18	77	8	85	18	4.7	0.1
(a) pelvic_mesh v hernia_mesh	0.055	2013-Q3	2017-Q4	18	77	8	85	18	4.7	0.1
(a) pelvic_mesh v hernia_mesh	0.060	2013-Q3	2017-Q4	18	76	6	82	18	4.6	0.1
(a) pelvic_mesh v hernia_mesh	0.065	2013-Q3	2017-Q4	18	75	5	80	18	4.4	0.1
(a) pelvic_mesh v hernia_mesh	0.070	2014-Q3	2017-Q4	14	73	3	76	14	5.4	0.0
(a) pelvic_mesh v hernia_mesh	0.075	2014-Q3	2017-Q4	14	72	3	75	14	5.4	0.0
(a) pelvic_mesh v hernia_mesh	0.080	2014-Q3	2017-Q4	14	72	3	75	14	5.4	0.0
(a) pelvic_mesh v hernia_mesh	0.085	2014-Q3	2017-Q4	14	71	2	73	14	5.2	0.0
(a) pelvic_mesh v hernia_mesh	0.090	2017-Q1	2017-Q4	4	70	1	71	4	17.8	0.0
(a) pelvic_mesh v hernia_mesh	0.095	2017-Q1	2017-Q4	4	69	1	70	4	17.5	0.0
(a) pelvic_mesh v hernia_mesh	0.100	2017-Q1	2017-Q4	4	69	1	70	4	17.5	0.0
(b) pelvic_mesh v hernia_mesh/other_mesh	0.010	2012-Q4	2017-Q4	21	82	59	141	21	6.7	0.7
(b) pelvic_mesh v hernia_mesh/other_mesh	0.015	2012-Q4	2017-Q4	21	82	59	141	21	6.7	0.7
(b) pelvic_mesh v hernia_mesh/other_mesh	0.020	2012-Q4	2017-Q4	21	82	56	138	21	6.6	0.7
(b) pelvic_mesh v hernia_mesh/other_mesh	0.025	2012-Q4	2017-Q4	21	82	56	138	21	6.6	0.7

grps	thresh	min_dt	max_dt	nrows	sum_n	sum_nd	tot_n	qtrs	n_per_qtr	z
(b) pelvic_mesh v hernia_mesh/other_mesh	0.030	2012-Q4	2017-Q4	21	82	55	137	21	6.5	0.7
(b) pelvic_mesh v hernia_mesh/other_mesh	0.035	2012-Q4	2017-Q4	21	82	54	136	21	6.5	0.7
(b) pelvic_mesh v hernia_mesh/other_mesh	0.040	2012-Q4	2017-Q4	21	81	51	132	21	6.3	0.6
(b) pelvic_mesh v hernia_mesh/other_mesh	0.045	2012-Q4	2017-Q4	21	79	47	126	21	6.0	0.6
(b) pelvic_mesh v hernia_mesh/other_mesh	0.050	2012-Q4	2017-Q4	21	77	45	122	21	5.8	0.6
(b) pelvic_mesh v hernia_mesh/other_mesh	0.055	2013-Q2	2017-Q4	19	77	43	120	19	6.3	0.6
(b) pelvic_mesh v hernia_mesh/other_mesh	0.060	2013-Q2	2017-Q4	19	76	41	117	19	6.2	0.5
(b) pelvic_mesh v hernia_mesh/other_mesh	0.065	2013-Q2	2017-Q4	19	75	38	113	19	5.9	0.5
(b) pelvic_mesh v hernia_mesh/other_mesh	0.070	2013-Q2	2017-Q4	19	73	36	109	19	5.7	0.5
(b) pelvic_mesh v hernia_mesh/other_mesh	0.075	2013-Q2	2017-Q4	19	72	35	107	19	5.6	0.5
(b) pelvic_mesh v hernia_mesh/other_mesh	0.080	2014-Q3	2017-Q4	14	72	33	105	14	7.5	0.5
(b) pelvic_mesh v hernia_mesh/other_mesh	0.085	2014-Q3	2017-Q4	14	71	31	102	14	7.3	0.4
(b) pelvic_mesh v hernia_mesh/other_mesh	0.090	2014-Q4	2017-Q4	13	70	30	100	13	7.7	0.4
(b) pelvic_mesh v hernia_mesh/other_mesh	0.095	2014-Q4	2017-Q4	13	69	30	99	13	7.6	0.4
(b) pelvic_mesh v hernia_mesh/other_mesh	0.100	2014-Q4	2017-Q4	13	69	30	99	13	7.6	0.4
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.010	2012-Q2	2017-Q4	23	82	2017	2099	23	91.3	24.6
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.015	2012-Q2	2017-Q4	23	82	1994	2076	23	90.3	24.3
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.020	2012-Q2	2017-Q4	23	82	1951	2033	23	88.4	23.8
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.025	2012-Q2	2017-Q4	23	82	1910	1992	23	86.6	23.3
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.030	2012-Q2	2017-Q4	23	82	1852	1934	23	84.1	22.6

grps	thresh	min_dtn	max_dtn	nrows	sum_nAm	sum_nCt	nqtrs	n_per_qtr
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.035	2012-Q2	2017-Q4	23	82	1783	1865	23	81.1	21.7
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.040	2012-Q2	2017-Q4	23	81	1715	1796	23	78.1	21.2
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.045	2012-Q2	2017-Q4	23	79	1656	1735	23	75.4	21.0
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.050	2012-Q2	2017-Q4	23	77	1584	1661	23	72.2	20.6
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.055	2012-Q2	2017-Q4	23	77	1510	1587	23	69.0	19.6
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.060	2012-Q2	2017-Q4	23	76	1406	1482	23	64.4	18.5
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.065	2012-Q2	2017-Q4	23	75	1331	1406	23	61.1	17.7
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.070	2012-Q2	2017-Q4	23	73	1258	1331	23	57.9	17.2
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.075	2012-Q2	2017-Q4	23	72	1186	1258	23	54.7	16.5
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.080	2012-Q2	2017-Q4	23	72	1104	1176	23	51.1	15.3
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.085	2012-Q2	2017-Q4	23	71	1024	1095	23	47.6	14.4
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.090	2012-Q4	2017-Q4	21	70	1013	1083	21	51.6	14.5
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.095	2012-Q4	2017-Q4	21	69	940	1009	21	48.0	13.6
(c) pelvic_mesh v hernia_mesh/other_mesh/other_device	0.100	2012-Q4	2017-Q4	21	69	933	1002	21	47.7	13.5
(d) hernia_mesh v other_mesh	0.010	2013-Q1	2017-Q4	20	12	47	59	20	3.0	3.9
(d) hernia_mesh v other_mesh	0.015	2013-Q1	2017-Q4	20	12	47	59	20	3.0	3.9
(d) hernia_mesh v other_mesh	0.020	2013-Q3	2017-Q4	18	10	46	56	18	3.1	4.6
(d) hernia_mesh v other_mesh	0.025	2013-Q3	2017-Q4	18	10	46	56	18	3.1	4.6
(d) hernia_mesh v other_mesh	0.030	2013-Q3	2017-Q4	18	10	45	55	18	3.1	4.5
(d) hernia_mesh v other_mesh	0.035	2013-Q3	2017-Q4	18	10	44	54	18	3.0	4.4

grps	thresh	min_dt	max_dt	nrows	sum_n	sum_n1	nqtrs	n_per_qtr	x
(d) hernia_mesh v other_mesh	0.040	2013-Q3	2017-Q4	18	9	42	51	18	2.8	4.7
(d) hernia_mesh v other_mesh	0.045	2013-Q3	2017-Q4	18	8	39	47	18	2.6	4.9
(d) hernia_mesh v other_mesh	0.050	2013-Q3	2017-Q4	18	8	37	45	18	2.5	4.6
(d) hernia_mesh v other_mesh	0.055	2013-Q3	2017-Q4	18	8	35	43	18	2.4	4.4
(d) hernia_mesh v other_mesh	0.060	2013-Q3	2017-Q4	18	6	35	41	18	2.3	5.8
(d) hernia_mesh v other_mesh	0.065	2013-Q3	2017-Q4	18	5	33	38	18	2.1	6.6
(d) hernia_mesh v other_mesh	0.070	2014-Q3	2017-Q4	14	3	33	36	14	2.6	11.0
(d) hernia_mesh v other_mesh	0.075	2014-Q3	2017-Q4	14	3	32	35	14	2.5	10.7
(d) hernia_mesh v other_mesh	0.080	2014-Q4	2017-Q4	13	3	30	33	13	2.5	10.0
(d) hernia_mesh v other_mesh	0.085	2014-Q4	2017-Q4	13	2	29	31	13	2.4	14.5
(d) hernia_mesh v other_mesh	0.090	2017-Q1	2017-Q4	4	1	29	30	4	7.5	29.0
(d) hernia_mesh v other_mesh	0.095	2017-Q1	2017-Q4	4	1	29	30	4	7.5	29.0
(d) hernia_mesh v other_mesh	0.100	2017-Q1	2017-Q4	4	1	29	30	4	7.5	29.0
(e) hernia_mesh/other_mesh v other_device	0.010	2012-Q4	2017-Q4	21	59	1958	2017	21	96.0	33.2
(e) hernia_mesh/other_mesh v other_device	0.015	2012-Q4	2017-Q4	21	59	1935	1994	21	95.0	32.8
(e) hernia_mesh/other_mesh v other_device	0.020	2012-Q4	2017-Q4	21	56	1895	1951	21	92.9	33.8
(e) hernia_mesh/other_mesh v other_device	0.025	2012-Q4	2017-Q4	21	56	1854	1910	21	91.0	33.1
(e) hernia_mesh/other_mesh v other_device	0.030	2012-Q4	2017-Q4	21	55	1797	1852	21	88.2	32.7
(e) hernia_mesh/other_mesh v other_device	0.035	2012-Q4	2017-Q4	21	54	1729	1783	21	84.9	32.0
(e) hernia_mesh/other_mesh v other_device	0.040	2012-Q4	2017-Q4	21	51	1664	1715	21	81.7	32.6

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	z
(e) hernia_mesh/other_mesh v other_device	0.045	2012-Q4	2017-Q4	21	47	1609	1656	21	78.9	34.2
(e) hernia_mesh/other_mesh v other_device	0.050	2012-Q4	2017-Q4	21	45	1539	1584	21	75.4	34.2
(e) hernia_mesh/other_mesh v other_device	0.055	2013-Q2	2017-Q4	19	43	1467	1510	19	79.5	34.1
(e) hernia_mesh/other_mesh v other_device	0.060	2013-Q2	2017-Q4	19	41	1365	1406	19	74.0	33.3
(e) hernia_mesh/other_mesh v other_device	0.065	2013-Q2	2017-Q4	19	38	1293	1331	19	70.1	34.0
(e) hernia_mesh/other_mesh v other_device	0.070	2013-Q2	2017-Q4	19	36	1222	1258	19	66.2	33.9
(e) hernia_mesh/other_mesh v other_device	0.075	2013-Q2	2017-Q4	19	35	1151	1186	19	62.4	32.9
(e) hernia_mesh/other_mesh v other_device	0.080	2014-Q3	2017-Q4	14	33	1071	1104	14	78.9	32.5
(e) hernia_mesh/other_mesh v other_device	0.085	2014-Q3	2017-Q4	14	31	993	1024	14	73.1	32.0
(e) hernia_mesh/other_mesh v other_device	0.090	2014-Q4	2017-Q4	13	30	983	1013	13	77.9	32.8
(e) hernia_mesh/other_mesh v other_device	0.095	2014-Q4	2017-Q4	13	30	910	940	13	72.3	30.3
(e) hernia_mesh/other_mesh v other_device	0.100	2014-Q4	2017-Q4	13	30	903	933	13	71.8	30.1

```r
# testing/example
row_i <- 1
cv_tab[row_i, ]
```

```
# A tibble: 1 x 11
  grps    thresh min_dte max_dte  rows sum_nA sum_nC tot_n  qtrs n_per_qtr     z
  <chr>   <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>     <dbl> <dbl>
1 (a) pe~ 0.010  2013-Q1 2017-Q4    20     82     12    94    20       4.7 0.146
```

```r
get_maxsprt_cv(cv_tab$tot_n[row_i], floor(cv_tab$n_per_qtr[row_i]), cv_tab$z[row_i])
```

```
[1] 3.27782
```

```r
row_i <- 50
cv_tab[row_i, ]
```

```
# A tibble: 1 x 11
  grps    thresh min_dte max_dte  rows sum_nA sum_nC tot_n  qtrs n_per_qtr     z
  <chr>   <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>     <dbl> <dbl>
1 (c) pe~ 0.065  2012-Q2 2017-Q4    23     75   1331  1406    23      61.1  17.7
```

```r
get_maxsprt_cv(cv_tab$tot_n[row_i], floor(cv_tab$n_per_qtr[row_i]), cv_tab$z[row_i])
```

```
Selected alpha: 0.048 (least conservative value below 0.05)

[1] 2.740269
attr(,"alpha")
[1] 0.048072
```

```r
### takes ~ 70 sec (i5-8400)
# note purrr::possibly() will just catch when model fails and return as.numeric(NA)
get_maxsprt_cv_poss <-
  possibly(get_maxsprt_cv, otherwise = NA_real_, quiet = FALSE)

tic()
cv_tab <-
  cv_tab %>%
  # dplyr::filter(row_number() < 7) %>% ### testing
  mutate(
    cv =
      future_pmap_dbl(
        .l = list(tot_n, floor(n_per_qtr), z),
        .f = ~get_maxsprt_cv_poss(..1, ..2, ..3),
        .options = furrr_seed3
      )
  )
```

```
Error: For this 'N' there is no solution with prob of Type I error smaller than 0.05. Use 'N

Error: For this 'N' there is no solution with prob of Type I error smaller than 0.05. Use 'N
```

```
Error: For this 'N' there is no solution with prob of Type I error smaller than 0.05. Use 'N
Error: For this 'N' there is no solution with prob of Type I error smaller than 0.05. Use 'N

Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)
Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.047 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.045 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.048 (least conservative value below 0.05)

Selected alpha: 0.046 (least conservative value below 0.05)
```

```
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)
Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.045 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.049 (least conservative value below 0.05)

Selected alpha: 0.050 (least conservative value below 0.05)

Selected alpha: 0.048 (least conservative value below 0.05)
```

```
toc()
```

```
29.4 sec elapsed
```

```
cv_tab
```

```
# A tibble: 95 x 12
   grps   thresh min_dte max_dte  rows sum_nA sum_nC tot_n  qtrs n_per_qtr     z
   <chr>  <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>     <dbl> <dbl>
 1 (a) p~ 0.010  2013-Q1 2017-Q4    20     82     12    94    20      4.7  0.146
 2 (a) p~ 0.015  2013-Q1 2017-Q4    20     82     12    94    20      4.7  0.146
 3 (a) p~ 0.020  2013-Q3 2017-Q4    18     82     10    92    18      5.11 0.122
 4 (a) p~ 0.025  2013-Q3 2017-Q4    18     82     10    92    18      5.11 0.122
 5 (a) p~ 0.030  2013-Q3 2017-Q4    18     82     10    92    18      5.11 0.122
 6 (a) p~ 0.035  2013-Q3 2017-Q4    18     82     10    92    18      5.11 0.122
 7 (a) p~ 0.040  2013-Q3 2017-Q4    18     81      9    90    18      5    0.111
 8 (a) p~ 0.045  2013-Q3 2017-Q4    18     79      8    87    18      4.83 0.101
 9 (a) p~ 0.050  2013-Q3 2017-Q4    18     77      8    85    18      4.72 0.104
10 (a) p~ 0.055  2013-Q3 2017-Q4    18     77      8    85    18      4.72 0.104
# i 85 more rows
# i 1 more variable: cv <dbl>
```

```
cv_tab %>% dplyr::filter(is.na(cv))
```

```
# A tibble: 4 x 12
  grps   thresh min_dte max_dte  rows sum_nA sum_nC tot_n  qtrs n_per_qtr      z
  <chr>  <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>     <dbl>  <dbl>
1 (a) p~ 0.085  2014-Q3 2017-Q4    14     71      2    73    14      5.21 0.0282
2 (a) p~ 0.090  2017-Q1 2017-Q4     4     70      1    71     4     17.8  0.0143
3 (a) p~ 0.095  2017-Q1 2017-Q4     4     69      1    70     4     17.5  0.0145
4 (a) p~ 0.100  2017-Q1 2017-Q4     4     69      1    70     4     17.5  0.0145
# i 1 more variable: cv <dbl>
```

```
maxsprt_dat <-
  sra_cum %>%
  mutate(
    maxllr = max_sprt_stat_(c_n = nA, n = nA + nC, z = (nC + nD) / (nA + nB)),
    rre = rr_est_(c_n = nA, n = nA + nC, z = (nC + nD) / (nA + nB))
  )

# maxsprt_dat


maxsprt_dat <-
  maxsprt_dat %>%
  left_join(
    .,
```

```
    cv_tab %>% select(grps, thresh, cv),
    c("grps", "thresh")
  )

maxsprt_dat <-
  maxsprt_dat %>%
  mutate(
    # some cvs don't exist so those llr never reach cv
    reached_cv = if_else(is.na(cv), 0L, as.integer(maxllr > cv)),
    # create date for start of each quarter
    dte =
      as_date(paste0(
        substr(mnth, 1, 5),
        sprintf("%02.0f", (as.integer(substr(mnth, 7, 7)) - 1) * 3 + 1),
        "-01"
      ))
  )

maxsprt_dat %>% dplyr::filter(is.na(cv))
```

```
# A tibble: 26 x 13
   grps         dat_type thresh mnth     nA    nB    nC    nD maxllr   rre    cv
   <chr>        <chr>    <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
 1 (a) pelvic_~ cumulat~ 0.085  2014~     6    20     1    23   1.79  5.54    NA
 2 (a) pelvic_~ cumulat~ 0.085  2014~    23    22     1    30   8.79 15.8     NA
 3 (a) pelvic_~ cumulat~ 0.085  2015~    24    22     1    31   9.37 16.7     NA
 4 (a) pelvic_~ cumulat~ 0.085  2015~    24    22     1    31   9.37 16.7     NA
 5 (a) pelvic_~ cumulat~ 0.085  2015~    24    23     1    31   9.17 16.3     NA
 6 (a) pelvic_~ cumulat~ 0.085  2015~    24    23     1    31   9.17 16.3     NA
 7 (a) pelvic_~ cumulat~ 0.085  2016~    31    24     1    33  11.4  19.2     NA
 8 (a) pelvic_~ cumulat~ 0.085  2016~    31    24     1    33  11.4  19.2     NA
 9 (a) pelvic_~ cumulat~ 0.085  2016~    33    26     1    39  13.5  22.4     NA
10 (a) pelvic_~ cumulat~ 0.085  2016~    33    26     1    39  13.5  22.4     NA
# i 16 more rows
# i 2 more variables: reached_cv <int>, dte <date>
```

```
# have a peak
maxsprt_dat %>%
  select(-dat_type) %>%
  print(., n = 25)
```

```
# A tibble: 1,707 x 12
   grps       thresh mnth     nA    nB    nC    nD maxllr   rre    cv reached_cv
   <chr>      <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>      <int>
 1 (a) pelvi~ 0.010  2013~     3     7     1     4 0.0657  1.5   3.28          0
 2 (a) pelvi~ 0.010  2013~     4    10     1     5 0.129   1.71  3.28          0
 3 (a) pelvi~ 0.010  2013~     5    11     2     9 0.224   1.72  3.28          0
 4 (a) pelvi~ 0.010  2013~     9    11     2     9 0.801   2.48  3.28          0
 5 (a) pelvi~ 0.010  2014~     9    11     2    10 0.976   2.7   3.28          0
 6 (a) pelvi~ 0.010  2014~    10    12     3    12 0.885   2.27  3.28          0
 7 (a) pelvi~ 0.010  2014~    12    14     5    19 1.22    2.22  3.28          0
 8 (a) pelvi~ 0.010  2014~    30    15     7    24 4.05    2.95  3.28          1
 9 (a) pelvi~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28          1
10 (a) pelvi~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28          1
11 (a) pelvi~ 0.010  2015~    31    16     7    25 4.27    3.02  3.28          1
12 (a) pelvi~ 0.010  2015~    31    16     9    25 3.36    2.49  3.28          1
13 (a) pelvi~ 0.010  2016~    35    16     9    25 3.83    2.59  3.28          1
14 (a) pelvi~ 0.010  2016~    39    16     9    25 4.27    2.68  3.28          1
15 (a) pelvi~ 0.010  2016~    39    16    10    30 5.17    2.84  3.28          1
16 (a) pelvi~ 0.010  2016~    41    18    10    30 5.04    2.78  3.28          1
17 (a) pelvi~ 0.010  2017~    50    18    11    31 5.85    2.81  3.28          1
18 (a) pelvi~ 0.010  2017~    63    19    12    33 7.04    2.88  3.28          1
19 (a) pelvi~ 0.010  2017~    73    19    12    34 8.18    3.04  3.28          1
20 (a) pelvi~ 0.010  2017~    82    20    12    34 8.65    3.08  3.28          1
21 (a) pelvi~ 0.015  2013~     3     7     1     4 0.0657  1.5   3.28          0
22 (a) pelvi~ 0.015  2013~     4    10     1     5 0.129   1.71  3.28          0
23 (a) pelvi~ 0.015  2013~     5    11     2     9 0.224   1.72  3.28          0
24 (a) pelvi~ 0.015  2013~     9    11     2     9 0.801   2.48  3.28          0
25 (a) pelvi~ 0.015  2014~     9    11     2    10 0.976   2.7   3.28          0
# i 1,682 more rows
# i 1 more variable: dte <date>
```

3 Session information

```
format(Sys.time(), '%d %b %Y')
```

```
[1] "10 Aug 2023"
```

```
Sys.info()[!(names(Sys.info()) %in% c("login", "nodename"))] %>%
  as.data.frame(.)
```

```
                          .
sysname             Windows
release              10 x64
version         build 17763
machine              x86-64
user                     ty
effective_user           ty
```

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 17763)

Matrix products: default


locale:
[1] LC_COLLATE=English_Australia.1252  LC_CTYPE=English_Australia.1252
[3] LC_MONETARY=English_Australia.1252 LC_NUMERIC=C
[5] LC_TIME=English_Australia.1252

time zone: Australia/Sydney
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
```

```
 [1] pharmsignal_0.1.0 arrow_12.0.1.1    foreach_1.5.2     gsDesign_3.5.0
 [5] knitr_1.43        ggrepel_0.9.3     ggplot2_3.4.2     tictoc_1.2
 [9] lubridate_1.9.2   furrr_0.3.1       future_1.33.0     purrr_1.0.1
[13] forcats_1.0.0     tidyr_1.3.0       dplyr_1.1.2       readr_2.1.4

loaded via a namespace (and not attached):
 [1] Sequential_4.3             gtable_0.3.3
 [3] xfun_0.39                  lattice_0.21-8
 [5] tzdb_0.4.0                 vctrs_0.6.3
 [7] tools_4.3.1                generics_0.1.3
 [9] parallel_4.3.1             tibble_3.2.1
[11] fansi_1.0.4                pkgconfig_2.0.3
[13] Matrix_1.6-0               assertthat_0.2.1
[15] gt_0.9.0                   lifecycle_1.0.3
[17] EmpiricalCalibration_3.1.1 compiler_4.3.1
[19] MatrixModels_0.5-2         mcmc_0.9-7
[21] munsell_0.5.0              codetools_0.2-19
[23] SparseM_1.81               quantreg_5.96
[25] htmltools_0.5.5            yaml_2.3.7
[27] pillar_1.9.0               MASS_7.3-60
[29] iterators_1.0.14           boot_1.3-28.1
[31] parallelly_1.36.0          tidyselect_1.2.0
[33] digest_0.6.33              listenv_0.9.0
[35] splines_4.3.1              fastmap_1.1.1
[37] grid_4.3.1                 colorspace_2.1-0
[39] cli_3.6.1                  magrittr_2.0.3
[41] survival_3.5-5             utf8_1.2.3
[43] withr_2.5.0                scales_1.2.1
[45] bit64_4.0.5                timechange_0.2.0
[47] rmarkdown_2.23             globals_0.16.2
[49] bit_4.0.5                  hms_1.1.3
[51] coda_0.19-4                evaluate_0.21
[53] rlang_1.1.1                MCMCpack_1.6-3
[55] Rcpp_1.0.11                xtable_1.8-4
[57] glue_1.6.2                 xml2_1.3.5
[59] rstudioapi_0.15.0          jsonlite_1.8.7
[61] R6_2.5.1
```