Data analysis

Signal detection of spontaneous medical device reports over time

Ty Stanford et al.

Table of contents

1 Set up

1.1 Packages

```r
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
})
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```
Warning: package 'future' was built under R version 4.2.3
```

```r
# 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")
```

1.2 Parallel compution setup

```
# options(future.globals.maxSize = 500 * 1024 ^ 2) # = 500 MiB
options(future.globals.maxSize = 1e3 * 1024 ^ 2) # = 1 GB



# furrr parallel workers/cores setup
# change `workers = 4` based on cores available in processor being used
plan(multisession, workers = 4)

### 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
```

```
2.94 sec elapsed
```

```
# 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)
```

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", "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)

}
```

1.5 Load data

```r
sra_dat <- read_parquet("dat/sra_dat.parquet")
```

2 Analysis

2.1 BCPNN

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

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

# testing/example
sra_cum$data[[1]]
```

```
# A tibble: 44 x 5
   mnth       nA    nB    nC    nD
   <chr>   <dbl> <dbl> <dbl> <dbl>
 1 2013-01     3     7     1     2
 2 2013-02     3     7     1     4
 3 2013-04     3     7     1     5
 4 2013-05     4    10     1     5
 5 2013-07     4    11     1     7
 6 2013-08     5    11     1     7
 7 2013-09     5    11     2     9
 8 2013-11     8    11     2     9
 9 2013-12     9    11     2     9
10 2014-03     9    11     2    10
# i 34 more rows
```

```
get_sig_tab_over_time(sra_cum$data[[1]])
```

```
     est_name est_scale         est          ci_lo     ci_hi
1   bcpnn_mcmc      log2 -0.03155473 -1.008567292 0.5011816
2   bcpnn_mcmc      log2  0.14423111 -0.881461274 0.7657139
3   bcpnn_mcmc      log2  0.22225892 -0.819459061 0.8710605
4   bcpnn_mcmc      log2  0.16982567 -0.631979264 0.6843789
5   bcpnn_mcmc      log2  0.25877437 -0.558250981 0.7901982
```

```
6  bcpnn_mcmc       log2  0.28945108 -0.359347606 0.7778787
7  bcpnn_mcmc       log2  0.24275055 -0.534420549 0.7826432
8  bcpnn_mcmc       log2  0.31486271 -0.170553281 0.7465905
9  bcpnn_mcmc       log2  0.32253264 -0.103384822 0.7293047
10 bcpnn_mcmc       log2  0.36527691 -0.067632484 0.7879723
11 bcpnn_mcmc       log2  0.40661823 -0.034175482 0.8369273
12 bcpnn_mcmc       log2  0.31431083 -0.117133215 0.7114928
13 bcpnn_mcmc       log2  0.35146856 -0.088210721 0.7597945
14 bcpnn_mcmc       log2  0.32236631 -0.145358061 0.7374164
15 bcpnn_mcmc       log2  0.41872427  0.005768877 0.8060994
16 bcpnn_mcmc       log2  0.42043314 -0.019393480 0.8193463
17 bcpnn_mcmc       log2  0.51134622  0.122251507 0.8922410
18 bcpnn_mcmc       log2  0.48997255  0.090959174 0.8764148
19 bcpnn_mcmc       log2  0.44480211  0.225394906 0.6963962
20 bcpnn_mcmc       log2  0.46323762  0.241125813 0.7192630
21 bcpnn_mcmc       log2  0.45935820  0.241366445 0.7109985
22 bcpnn_mcmc       log2  0.44698469  0.231636145 0.6971812
23 bcpnn_mcmc       log2  0.42807282  0.211373941 0.6762936
24 bcpnn_mcmc       log2  0.40986367  0.191379522 0.6565181
25 bcpnn_mcmc       log2  0.40529856  0.198145687 0.6409685
26 bcpnn_mcmc       log2  0.40021287  0.201086693 0.6289638
27 bcpnn_mcmc       log2  0.38907653  0.206723284 0.6022235
28 bcpnn_mcmc       log2  0.42060041  0.232653673 0.6397847
29 bcpnn_mcmc       log2  0.43735676  0.243979517 0.6600397
30 bcpnn_mcmc       log2  0.45236046  0.256382608 0.6771721
31 bcpnn_mcmc       log2  0.43564014  0.249050440 0.6506842
32 bcpnn_mcmc       log2  0.42578750  0.240710552 0.6406515
33 bcpnn_mcmc       log2  0.41437069  0.240937476 0.6170948
34 bcpnn_mcmc       log2  0.40562061  0.238810116 0.5993361
35 bcpnn_mcmc       log2  0.40236748  0.238325445 0.5936070
36 bcpnn_mcmc       log2  0.40588775  0.250241656 0.5864129
37 bcpnn_mcmc       log2  0.38625000  0.238767885 0.5591967
38 bcpnn_mcmc       log2  0.37617897  0.235058228 0.5426621
39 bcpnn_mcmc       log2  0.38226092  0.243192323 0.5467141
40 bcpnn_mcmc       log2  0.36736813  0.235354142 0.5233362
41 bcpnn_mcmc       log2  0.36257011  0.233573242 0.5155766
42 bcpnn_mcmc       log2  0.35553674  0.229417693 0.5041057
43 bcpnn_mcmc       log2  0.34192103  0.220789503 0.4841618
44 bcpnn_mcmc       log2  0.33766947  0.219170482 0.4782721
```

```
### takes ~ 90 sec
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    sig_tab =
      future_map(
        .x = data,
        .f = get_sig_tab_over_time,
        .options = furrr_seed1
      )
  )
toc()
```

```
91.18 sec elapsed
```

```
# check
sra_cum$sig_tab[[1]]
```

```
     est_name est_scale         est       ci_lo     ci_hi
1  bcpnn_mcmc      log2 -0.03155473 -1.01361518 0.5008338
2  bcpnn_mcmc      log2  0.14423111 -0.87754425 0.7674485
3  bcpnn_mcmc      log2  0.22225892 -0.82925568 0.8768976
4  bcpnn_mcmc      log2  0.16982567 -0.62630109 0.6841108
5  bcpnn_mcmc      log2  0.25877437 -0.56115069 0.7903408
6  bcpnn_mcmc      log2  0.28945108 -0.36195188 0.7791041
7  bcpnn_mcmc      log2  0.24275055 -0.54147551 0.7865138
8  bcpnn_mcmc      log2  0.31486271 -0.16358209 0.7464793
9  bcpnn_mcmc      log2  0.32253264 -0.09929240 0.7367417
10 bcpnn_mcmc      log2  0.36527691 -0.06997405 0.7826290
11 bcpnn_mcmc      log2  0.40661823 -0.03559667 0.8371538
12 bcpnn_mcmc      log2  0.31431083 -0.12150818 0.7158149
13 bcpnn_mcmc      log2  0.35146856 -0.08877886 0.7628745
14 bcpnn_mcmc      log2  0.32236631 -0.15065287 0.7402075
15 bcpnn_mcmc      log2  0.41872427  0.00681964 0.8113143
16 bcpnn_mcmc      log2  0.42043314 -0.01475558 0.8214752
17 bcpnn_mcmc      log2  0.51134622  0.12474650 0.8952461
18 bcpnn_mcmc      log2  0.48997255  0.09205215 0.8748550
19 bcpnn_mcmc      log2  0.44480211  0.22642533 0.6997523
20 bcpnn_mcmc      log2  0.46323762  0.24023405 0.7193275
```

```
21 bcpnn_mcmc      log2  0.45935820  0.24200791 0.7127425
22 bcpnn_mcmc      log2  0.44698469  0.23167170 0.6944536
23 bcpnn_mcmc      log2  0.42807282  0.21226736 0.6772533
24 bcpnn_mcmc      log2  0.40986367  0.19249103 0.6557647
25 bcpnn_mcmc      log2  0.40529856  0.19777767 0.6411809
26 bcpnn_mcmc      log2  0.40021287  0.20072109 0.6295022
27 bcpnn_mcmc      log2  0.38907653  0.20484737 0.6038758
28 bcpnn_mcmc      log2  0.42060041  0.23274148 0.6415478
29 bcpnn_mcmc      log2  0.43735676  0.24612622 0.6588602
30 bcpnn_mcmc      log2  0.45236046  0.25818115 0.6760420
31 bcpnn_mcmc      log2  0.43564014  0.24895062 0.6536131
32 bcpnn_mcmc      log2  0.42578750  0.23965009 0.6400406
33 bcpnn_mcmc      log2  0.41437069  0.23951271 0.6165066
34 bcpnn_mcmc      log2  0.40562061  0.23923085 0.5994463
35 bcpnn_mcmc      log2  0.40236748  0.23938575 0.5932197
36 bcpnn_mcmc      log2  0.40588775  0.24969119 0.5872826
37 bcpnn_mcmc      log2  0.38625000  0.23831821 0.5599170
38 bcpnn_mcmc      log2  0.37617897  0.23431857 0.5426523
39 bcpnn_mcmc      log2  0.38226092  0.24172534 0.5480638
40 bcpnn_mcmc      log2  0.36736813  0.23617735 0.5236761
41 bcpnn_mcmc      log2  0.36257011  0.23406734 0.5142182
42 bcpnn_mcmc      log2  0.35553674  0.22958272 0.5048607
43 bcpnn_mcmc      log2  0.34192103  0.22109856 0.4837613
44 bcpnn_mcmc      log2  0.33766947  0.22005543 0.4760598
```

```r
sra_cum_bcpnn <-
  sra_cum %>%
  unnest(cols = c(data, sig_tab)) %>%
  mutate(dte = as_date(paste0(mnth, "-01")))

sra_cum_bcpnn
```

```
# A tibble: 4,523 x 14
   grps         dat_type thresh mnth     nA    nB    nC    nD est_name est_scale
   <chr>        <chr>     <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <chr>    <chr>
 1 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     2 bcpnn_m~ log2
 2 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     4 bcpnn_m~ log2
 3 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     5 bcpnn_m~ log2
 4 pelvic_mesh~ cumulat~   0.01 2013~     4    10     1     5 bcpnn_m~ log2
 5 pelvic_mesh~ cumulat~   0.01 2013~     4    11     1     7 bcpnn_m~ log2
 6 pelvic_mesh~ cumulat~   0.01 2013~     5    11     1     7 bcpnn_m~ log2
 7 pelvic_mesh~ cumulat~   0.01 2013~     5    11     2     9 bcpnn_m~ log2
 8 pelvic_mesh~ cumulat~   0.01 2013~     8    11     2     9 bcpnn_m~ log2
 9 pelvic_mesh~ cumulat~   0.01 2013~     9    11     2     9 bcpnn_m~ log2
10 pelvic_mesh~ cumulat~   0.01 2014~     9    11     2    10 bcpnn_m~ log2
# i 4,513 more rows
# i 4 more variables: est <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] 4523
```

```
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] 4523
```

```
sra_cum_bcpnn
```

```
# A tibble: 4,523 x 15
   grps          dat_type thresh mnth      nA    nB    nC    nD est_name est_scale
   <chr>         <chr>     <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <chr>    <chr>
```

```
 1 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     2 bcpnn_m~ log2
 2 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     4 bcpnn_m~ log2
 3 pelvic_mesh~ cumulat~   0.01 2013~     3     7     1     5 bcpnn_m~ log2
 4 pelvic_mesh~ cumulat~   0.01 2013~     4    10     1     5 bcpnn_m~ log2
 5 pelvic_mesh~ cumulat~   0.01 2013~     4    11     1     7 bcpnn_m~ log2
 6 pelvic_mesh~ cumulat~   0.01 2013~     5    11     1     7 bcpnn_m~ log2
 7 pelvic_mesh~ cumulat~   0.01 2013~     5    11     2     9 bcpnn_m~ log2
 8 pelvic_mesh~ cumulat~   0.01 2013~     8    11     2     9 bcpnn_m~ log2
 9 pelvic_mesh~ cumulat~   0.01 2013~     9    11     2     9 bcpnn_m~ log2
10 pelvic_mesh~ cumulat~   0.01 2014~     9    11     2    10 bcpnn_m~ log2
# i 4,513 more rows
# i 5 more variables: est <dbl>, ci_lo <dbl>, ci_hi <dbl>, dte <date>,
#   dte_reach_sig <date>
```

```r
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

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

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,
    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
hernia_mesh v other_mesh	0.0	2013-01	2017-12	49	12	47	59	14.8	4.0	3.9
hernia_mesh v other_mesh	0.0	2013-01	2017-12	49	12	47	59	14.8	4.0	3.9
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	10	46	56	12.8	4.4	4.6
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	10	46	56	12.8	4.4	4.6
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	10	45	55	12.8	4.3	4.5
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	10	44	54	12.8	4.2	4.4
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	9	42	51	12.8	4.0	4.7
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	8	39	47	12.8	3.7	4.9
hernia_mesh v other_mesh	0.0	2013-09	2017-12	42	8	37	45	12.8	3.5	4.6
hernia_mesh v other_mesh	0.1	2013-09	2017-12	42	8	35	43	12.8	3.4	4.4
hernia_mesh v other_mesh	0.1	2013-09	2017-12	42	6	35	41	12.8	3.2	5.8
hernia_mesh v other_mesh	0.1	2013-09	2017-12	42	5	33	38	12.8	3.0	6.6
hernia_mesh v other_mesh	0.1	2014-07	2017-12	34	3	33	36	10.2	3.5	11.0
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	59	1958	2017	15.2	132.3	33.2

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	nqtrs	n_per_qtr	z
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	59	1935	1994	15.2	130.8	32.8
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	56	1895	1951	15.2	127.9	33.8
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	56	1854	1910	15.2	125.2	33.1
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	55	1797	1852	15.2	121.4	32.7
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	54	1729	1783	15.2	116.9	32.0
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	51	1664	1715	15.2	112.5	32.6
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	47	1609	1656	15.2	108.6	34.2
hernia_mesh/other_mesh v other_device	0.0	2012-11	2017-12	62	45	1539	1584	15.2	103.9	34.2
hernia_mesh/other_mesh v other_device	0.1	2013-04	2017-12	57	43	1467	1510	14.0	107.9	34.1
hernia_mesh/other_mesh v other_device	0.1	2013-04	2017-12	57	41	1365	1406	14.0	100.4	33.3
hernia_mesh/other_mesh v other_device	0.1	2013-04	2017-12	57	38	1293	1331	14.0	95.1	34.0
hernia_mesh/other_mesh v other_device	0.1	2013-04	2017-12	57	36	1222	1258	14.0	89.9	33.9
pelvic_mesh v hernia_mesh	0.0	2013-01	2017-12	44	82	12	94	14.8	6.4	0.1
pelvic_mesh v hernia_mesh	0.0	2013-01	2017-12	44	82	12	94	14.8	6.4	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	82	10	92	12.8	7.2	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	82	10	92	12.8	7.2	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	82	10	92	12.8	7.2	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	82	10	92	12.8	7.2	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	81	9	90	12.8	7.1	0.1
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	79	8	87	12.8	6.8	0.1

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	z
pelvic_mesh v hernia_mesh	0.0	2013-09	2017-12	38	77	8	85	12.8	6.7	0.1
pelvic_mesh v hernia_mesh	0.1	2013-09	2017-12	38	77	8	85	12.8	6.7	0.1
pelvic_mesh v hernia_mesh	0.1	2013-09	2017-12	38	76	6	82	12.8	6.4	0.1
pelvic_mesh v hernia_mesh	0.1	2013-09	2017-12	38	75	5	80	12.8	6.3	0.1
pelvic_mesh v hernia_mesh	0.1	2014-07	2017-12	31	73	3	76	10.2	7.4	0.0
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	59	141	15.2	9.2	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	59	141	15.2	9.2	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	56	138	15.2	9.0	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	56	138	15.2	9.0	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	55	137	15.2	9.0	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	82	54	136	15.2	8.9	0.7
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	81	51	132	15.2	8.7	0.6
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	79	47	126	15.2	8.3	0.6
pelvic_mesh v hernia_mesh/other_mesh	0.0	2012-11	2017-12	55	77	45	122	15.2	8.0	0.6
pelvic_mesh v hernia_mesh/other_mesh	0.1	2013-04	2017-12	51	77	43	120	14.0	8.6	0.6
pelvic_mesh v hernia_mesh/other_mesh	0.1	2013-04	2017-12	51	76	41	117	14.0	8.4	0.5
pelvic_mesh v hernia_mesh/other_mesh	0.1	2013-04	2017-12	51	75	38	113	14.0	8.1	0.5
pelvic_mesh v hernia_mesh/other_mesh	0.1	2013-04	2017-12	51	73	36	109	14.0	7.8	0.5
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	2017	2099	16.5	127.2	24.6
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	1994	2076	16.5	125.8	24.3

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	z
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	1951	2033	16.5	123.2	23.8
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	1910	1992	16.5	120.7	23.3
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	1852	1934	16.5	117.2	22.6
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	82	1783	1865	16.5	113.0	21.7
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	81	1715	1796	16.5	108.8	21.2
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	79	1656	1735	16.5	105.2	21.0
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.0	2012-06	2017-12	67	77	1584	1661	16.5	100.7	20.6
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.1	2012-06	2017-12	67	77	1510	1587	16.5	96.2	19.6
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.1	2012-06	2017-12	67	76	1406	1482	16.5	89.8	18.5
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.1	2012-06	2017-12	67	75	1331	1406	16.5	85.2	17.7
pelvic_mesh v hernia_mesh/other_mesh/other_device	0.1	2012-06	2017-12	67	73	1258	1331	16.5	80.7	17.2

```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>     <dbl> <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl>
1 hernia~    0.01 2013-01 2017-12    49     12     47    59  14.8         4  3.92
```

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

```
[1] 2.75595
```

```
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>    <dbl> <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl>
1 pelvic~   0.06 2013-04 2017-12    51     76     41   117    14      8.36 0.539
```

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

```
[1] 3.45153
```

```
### takes ~ 1 min
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(..1, ..2, ..3),
        .options = furrr_seed2
      )
  )
```

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

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

Selected alpha: 0.048 (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: 0.047 (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: 0.050 (least conservative value below 0.05)

Selected alpha: 0.047 (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: 0.048 (least conservative value below 0.05)

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

```
toc()
```

```
65.4 sec elapsed
```

```r
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 %>%
  inner_join(
    .,
    cv_tab %>% select(grps, thresh, cv),
    c("grps", "thresh")
  ) %>%
  mutate(reached_cv = as.integer(maxllr > cv))


maxsprt_dat %>%
  select(-dat_type) %>%
  print(., n = 25)
```

```
# A tibble: 3,407 x 11
   grps        thresh mnth     nA    nB    nC    nD maxllr   rre    cv reached_cv
   <chr>        <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>      <int>
 1 pelvic_me~    0.01 2013~     3     7     1     2 0       0.9   3.27          0
 2 pelvic_me~    0.01 2013~     3     7     1     4 0.0657  1.5   3.27          0
 3 pelvic_me~    0.01 2013~     3     7     1     5 0.141   1.8   3.27          0
 4 pelvic_me~    0.01 2013~     4    10     1     5 0.129   1.71  3.27          0
 5 pelvic_me~    0.01 2013~     4    11     1     7 0.264   2.13  3.27          0
 6 pelvic_me~    0.01 2013~     5    11     1     7 0.423   2.5   3.27          0
 7 pelvic_me~    0.01 2013~     5    11     2     9 0.224   1.72  3.27          0
 8 pelvic_me~    0.01 2013~     8    11     2     9 0.657   2.32  3.27          0
 9 pelvic_me~    0.01 2013~     9    11     2     9 0.801   2.48  3.27          0
10 pelvic_me~    0.01 2014~     9    11     2    10 0.976   2.7   3.27          0
11 pelvic_me~    0.01 2014~     9    11     2    11 1.15    2.92  3.27          0
12 pelvic_me~    0.01 2014~    10    12     3    11 0.735   2.12  3.27          0
13 pelvic_me~    0.01 2014~    10    12     3    12 0.885   2.27  3.27          0
14 pelvic_me~    0.01 2014~    10    12     4    13 0.671   1.93  3.27          0
15 pelvic_me~    0.01 2014~    12    14     4    17 1.33    2.42  3.27          0
```

```
16 pelvic_me~   0.01 2014~    12    14      5     19 1.22     2.22  3.27           0
17 pelvic_me~   0.01 2014~    14    14      5     22 2.06     2.7   3.27           0
18 pelvic_me~   0.01 2014~    14    14      6     23 1.79     2.42  3.27           0
19 pelvic_me~   0.01 2014~    30    15      7     24 4.05     2.95  3.27           1
20 pelvic_me~   0.01 2015~    30    15      7     25 4.31     3.05  3.27           1
21 pelvic_me~   0.01 2015~    31    15      7     25 4.45     3.08  3.27           1
22 pelvic_me~   0.01 2015~    31    16      7     25 4.27     3.02  3.27           1
23 pelvic_me~   0.01 2015~    31    16      8     25 3.78     2.72  3.27           1
24 pelvic_me~   0.01 2015~    31    16      9     25 3.36     2.49  3.27           1
25 pelvic_me~   0.01 2016~    33    16      9     25 3.60     2.54  3.27           1
# i 3,382 more rows
```

3 Session information

```
sra_cum_bcpnn %>%
  write_parquet(., sink = "out/sra_cum_bcpnn.parquet")


## close multisession workers by switching plan
plan(sequential)
```

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

```
[1] "17 Jul 2023"
```

```
Sys.info() %>% as.data.frame(.)
```

```
                             .
sysname                Windows
release                 10 x64
version            build 19044
nodename       DESKTOP-R5P5N23
machine                 x86-64
login                       ty
user                        ty
effective_user              ty
```

```
sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default

locale:
[1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8
[3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C
[5] LC_TIME=English_Australia.utf8
```

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

other attached packages:
 [1] pharmsignal_0.1.0 arrow_11.0.0.2    foreach_1.5.2     gsDesign_3.4.0
 [5] knitr_1.42        ggrepel_0.9.3     ggplot2_3.4.1     tictoc_1.1
 [9] lubridate_1.9.2   furrr_0.3.1       future_1.32.0     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] Rcpp_1.0.10        lattice_0.20-45    listenv_0.9.0      assertthat_0.2.1
 [5] digest_0.6.31      utf8_1.2.3         parallelly_1.34.0  R6_2.5.1
 [9] MatrixModels_0.5-1 evaluate_0.20      coda_0.19-4        pillar_1.9.0
[13] rlang_1.1.1        rstudioapi_0.14    SparseM_1.81       Matrix_1.5-3
[17] rmarkdown_2.20     splines_4.2.2      bit_4.0.5          munsell_0.5.0
[21] compiler_4.2.2     xfun_0.37          pkgconfig_2.0.3    globals_0.16.2
[25] mcmc_0.9-7         htmltools_0.5.4    tidyselect_1.2.0   tibble_3.2.1
[29] codetools_0.2-18   fansi_1.0.4        tzdb_0.3.0         withr_2.5.0
[33] MASS_7.3-58.1      grid_4.2.2         jsonlite_1.8.4     xtable_1.8-4
[37] gtable_0.3.1       lifecycle_1.0.3    magrittr_2.0.3     scales_1.2.1
[41] cli_3.6.0          ellipsis_0.3.2     generics_0.1.3     vctrs_0.6.3
[45] boot_1.3-28        iterators_1.0.14   tools_4.2.2        bit64_4.0.5
[49] glue_1.6.2         hms_1.1.2          parallel_4.2.2     fastmap_1.1.0
[53] survival_3.4-0     yaml_2.3.7         timechange_0.2.0   colorspace_2.1-0
[57] Sequential_4.3     quantreg_5.94      MCMCpack_1.6-3
```