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 'furrr' was built under R version 4.2.3
```

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

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

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

```
Warning: package 'foreach' 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")
```

```
### 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] 6
```

```
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
```

```
6.14 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 + overhead
```

```
4.14 sec elapsed
```

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

```
4.28 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)
furrr_seed3 <- furrr_options(seed = 3456)
```

1.3 Constants

```
# arbitrarily, let's go with minimum cell count of 1 (will change based on context/applica
arbitrary_cell_min <- 1
```

1.4 Functions

```
# do 90% CI only with lower == one sided 0.05
get_sig_tab <- function(nA, nB, nC, nD, alpha = 0.10, method = "bcpnn", n_mcmc = 1e+05) {

  out_cols_of_interest <- c("est_name", "est_scale", "est", "alpha", "ci_lo") # "ci_hi" (c
  sig_tab <- NULL # initialise in scope
  if (method == "bcpnn") {
    sig_tab <- pharmsignal::bcpnn_mcmc_signal(nA, nB, nC, nD, alpha = alpha, n_mcmc = n_mc
  } else if (method == "prr") {
    sig_tab <- pharmsignal::prr_signal(nA, nB, nC, nD, alpha = alpha)
  } else {
    stop("method for calcaultions unknown")
  }
  sig_tab <- sig_tab[, out_cols_of_interest]
  # sig_tab <- bind_cols(tibble(mnth = mnth), sig_tab)
  return(sig_tab)

}
```

```r
get_sig_tab_over_time <- function(dat, alpha = 0.10, method = "bcpnn", 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, method = method, n_mcmc = n_mcmc
        )
      )
    }

  return(sig_tab_over_time)

}



# if it's multiple comparisons central need to sparing use alpha
get_mult_compare_adj_alpha <- function(dat, alpha = 0.1) {

  n_reports <- nrow(dat)

  information_fracs <- (1:n_reports) / n_reports

  ### alternatives:
  # spend_obj <- sfLDPocock(alpha = alpha, t = information_fracs, param = NULL)
  # spend_obj <- sfLDOF(alpha = alpha, t = information_fracs, param = NULL)
  spend_obj <- sfExponential(alpha = alpha, t = information_fracs, param = 0.5)

  # plot(1:n_reports, spend_obj$spend, main = "alpha spending func", xlab = "look")

  return(bind_cols(dat, adj_alpha = spend_obj$spend))

}
```

```
# same as get_sig_tab_over_time(), however, alpha assumed included as column in data
get_sig_tab_over_time_2 <- function(dat, method = "bcpnn", 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],
          method = method,
          n_mcmc = n_mcmc
        )
      )
    }

  return(sig_tab_over_time)

}

# test
data.frame(nA = 30, nB = 5512, nC = 41, nD = 17445, adj_alpha = 0.1) %>%
  get_sig_tab_over_time_2(.)
```

```
    est_name est_scale       est alpha     ci_lo
1 bcpnn_mcmc      log2 0.7942907   0.1 0.4317622
```

```
data.frame(nA = 30, nB = 5512, nC = 41, nD = 17445, adj_alpha = 0.1) %>%
  get_sig_tab_over_time_2(., method = "prr")
```

```
  est_name  est_scale      est alpha    ci_lo
1      prr orig scale 2.308667   0.1 1.556277
```

```
2 ^ c(0.432304, 0.7942907) # similar to prr on ratio scale
```

```
[1] 1.349387 1.734225
```

```r
log2(c(1.556277, 2.308667)) # similar to bcpnn on log2 scale
```

```
[1] 0.6380989 1.2070601
```

1.5 Load data

```r
### 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
1  bcpnn_mcmc      log2 0.3778604   0.1 -0.27151962
2  bcpnn_mcmc      log2 0.3738316   0.1 -0.06911637
3  bcpnn_mcmc      log2 0.4150070   0.1 -0.03108974
4  bcpnn_mcmc      log2 0.5127001   0.1  0.11104715
5  bcpnn_mcmc      log2 0.4951774   0.1  0.06535270
6  bcpnn_mcmc      log2 0.5370558   0.1  0.33465530
7  bcpnn_mcmc      log2 0.5500767   0.1  0.35025304
8  bcpnn_mcmc      log2 0.5500767   0.1  0.35086730
9  bcpnn_mcmc      log2 0.5377636   0.1  0.33836792
10 bcpnn_mcmc      log2 0.4850381   0.1  0.28365055
11 bcpnn_mcmc      log2 0.4647627   0.1  0.27878002
12 bcpnn_mcmc      log2 0.4520794   0.1  0.27929577
13 bcpnn_mcmc      log2 0.5093946   0.1  0.32733841
14 bcpnn_mcmc      log2 0.4823956   0.1  0.30910135
15 bcpnn_mcmc      log2 0.4519206   0.1  0.30245415
16 bcpnn_mcmc      log2 0.4402744   0.1  0.30985519
17 bcpnn_mcmc      log2 0.4208998   0.1  0.30189089
18 bcpnn_mcmc      log2 0.3915001   0.1  0.28170266
```

```
### 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()
```

```
81.43 sec elapsed
```

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

```
     est_name est_scale       est alpha       ci_lo
1  bcpnn_mcmc      log2 0.3778604   0.1 -0.27857272
2  bcpnn_mcmc      log2 0.3738316   0.1 -0.06530536
3  bcpnn_mcmc      log2 0.4150070   0.1 -0.03003815
4  bcpnn_mcmc      log2 0.5127001   0.1  0.11613920
5  bcpnn_mcmc      log2 0.4951774   0.1  0.06855644
6  bcpnn_mcmc      log2 0.5370558   0.1  0.33396400
7  bcpnn_mcmc      log2 0.5500767   0.1  0.34967589
8  bcpnn_mcmc      log2 0.5500767   0.1  0.35032689
9  bcpnn_mcmc      log2 0.5377636   0.1  0.33806521
10 bcpnn_mcmc      log2 0.4850381   0.1  0.28283536
11 bcpnn_mcmc      log2 0.4647627   0.1  0.27663489
12 bcpnn_mcmc      log2 0.4520794   0.1  0.28009528
13 bcpnn_mcmc      log2 0.5093946   0.1  0.33020338
14 bcpnn_mcmc      log2 0.4823956   0.1  0.31108265
15 bcpnn_mcmc      log2 0.4519206   0.1  0.30188642
16 bcpnn_mcmc      log2 0.4402744   0.1  0.30898281
17 bcpnn_mcmc      log2 0.4208998   0.1  0.30167104
18 bcpnn_mcmc      log2 0.3915001   0.1  0.28215524
```

```
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 14
   grps       dat_t~1 thresh mnth      nA    nB    nC    nD est_n~2 est_s~3   est
   <chr>      <chr>   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <chr>   <chr>   <dbl>
 1 (a) pelvi~ cumula~ 0.010  2013~      3     7     1     4 bcpnn_~ log2    0.144
 2 (a) pelvi~ cumula~ 0.010  2013~      4    10     1     5 bcpnn_~ log2    0.170
 3 (a) pelvi~ cumula~ 0.010  2013~      5    11     2     9 bcpnn_~ log2    0.243
 4 (a) pelvi~ cumula~ 0.010  2013~      9    11     2     9 bcpnn_~ log2    0.323
 5 (a) pelvi~ cumula~ 0.010  2014~      9    11     2    10 bcpnn_~ log2    0.365
 6 (a) pelvi~ cumula~ 0.010  2014~     10    12     3    12 bcpnn_~ log2    0.351
 7 (a) pelvi~ cumula~ 0.010  2014~     12    14     5    19 bcpnn_~ log2    0.420
 8 (a) pelvi~ cumula~ 0.010  2014~     30    15     7    24 bcpnn_~ log2    0.445
 9 (a) pelvi~ cumula~ 0.010  2015~     31    15     7    25 bcpnn_~ log2    0.459
10 (a) pelvi~ cumula~ 0.010  2015~     31    15     7    25 bcpnn_~ log2    0.459
# ... with 1,697 more rows, 3 more variables: alpha <dbl>, ci_lo <dbl>,
#   dte <date>, and abbreviated variable names 1: dat_type, 2: est_name,
#   3: est_scale
```

```
# 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,
```

```r
    bcpnn_signif %>% select(grps, dat_type, thresh, dte_reach_sig),
    c("grps", "dat_type", "thresh")
  )
nrow(sra_cum_bcpnn)
```

```
[1] 1707
```

```r
sra_cum_bcpnn
```

```
# A tibble: 1,707 x 15
   grps       dat_t~1 thresh mnth     nA    nB    nC    nD est_n~2 est_s~3   est
   <chr>      <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <chr>   <chr>   <dbl>
 1 (a) pelvi~ cumula~ 0.010  2013~     3     7     1     4 bcpnn_~ log2    0.144
 2 (a) pelvi~ cumula~ 0.010  2013~     4    10     1     5 bcpnn_~ log2    0.170
 3 (a) pelvi~ cumula~ 0.010  2013~     5    11     2     9 bcpnn_~ log2    0.243
 4 (a) pelvi~ cumula~ 0.010  2013~     9    11     2     9 bcpnn_~ log2    0.323
 5 (a) pelvi~ cumula~ 0.010  2014~     9    11     2    10 bcpnn_~ log2    0.365
 6 (a) pelvi~ cumula~ 0.010  2014~    10    12     3    12 bcpnn_~ log2    0.351
 7 (a) pelvi~ cumula~ 0.010  2014~    12    14     5    19 bcpnn_~ log2    0.420
 8 (a) pelvi~ cumula~ 0.010  2014~    30    15     7    24 bcpnn_~ log2    0.445
 9 (a) pelvi~ cumula~ 0.010  2015~    31    15     7    25 bcpnn_~ log2    0.459
10 (a) pelvi~ cumula~ 0.010  2015~    31    15     7    25 bcpnn_~ log2    0.459
# ... with 1,697 more rows, 4 more variables: alpha <dbl>, ci_lo <dbl>,
#   dte <date>, dte_reach_sig <date>, and abbreviated variable names
#   1: dat_type, 2: est_name, 3: est_scale
```

```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
  )
```

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

2.2 BCPNN with mult comp adjust

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


sra_cum <-
  sra_cum %>%
  nest(data = c(mnth, nA, nB, nC, nD))



# test get_mult_compare_adj_alpha()
get_mult_compare_adj_alpha(sra_cum$data[[11]])
```

```
# A tibble: 18 x 6
   mnth       nA    nB    nC    nD adj_alpha
   <chr>   <dbl> <dbl> <dbl> <dbl>     <dbl>
 1 2013-Q3     4    12     1    10 0.0000572
 2 2013-Q4     5    15     1    10 0.001
 3 2014-Q1     5    15     1    11 0.00355
 4 2014-Q2     6    16     1    14 0.00756
 5 2014-Q3     8    18     3    21 0.0127
 6 2014-Q4    25    20     4    27 0.0185
 7 2015-Q1    26    20     4    28 0.0249
 8 2015-Q2    26    20     4    28 0.0316
 9 2015-Q3    26    21     4    28 0.0385
10 2015-Q4    26    21     4    30 0.0455
11 2016-Q1    29    22     4    30 0.0526
12 2016-Q2    33    22     4    30 0.0596
13 2016-Q3    33    22     5    35 0.0666
14 2016-Q4    35    24     5    35 0.0735
15 2017-Q1    44    24     6    36 0.0803
16 2017-Q2    57    25     6    39 0.0870
17 2017-Q3    67    25     6    40 0.0935
18 2017-Q4    76    26     6    40 0.1
```

```r
get_sig_tab_over_time_2(get_mult_compare_adj_alpha(sra_cum$data[[11]]))
```

```
     est_name est_scale       est        alpha      ci_lo
1  bcpnn_mcmc      log2 0.3778604 5.719516e-05 -2.7775539
2  bcpnn_mcmc      log2 0.3314299 1.000000e-03 -1.3130928
3  bcpnn_mcmc      log2 0.3719247 3.552305e-03 -0.9735553
4  bcpnn_mcmc      log2 0.4794164 7.562748e-03 -0.4878578
5  bcpnn_mcmc      log2 0.4505463 1.266582e-02 -0.3737733
6  bcpnn_mcmc      log2 0.5291089 1.853315e-02  0.2334080
7  bcpnn_mcmc      log2 0.5426635 2.491337e-02  0.2676446
8  bcpnn_mcmc      log2 0.5426635 3.162278e-02  0.2787698
9  bcpnn_mcmc      log2 0.5303613 3.852888e-02  0.2786349
10 bcpnn_mcmc      log2 0.5654356 4.553645e-02  0.3168146
11 bcpnn_mcmc      log2 0.5392350 5.257699e-02  0.3160579
12 bcpnn_mcmc      log2 0.5198015 5.960122e-02  0.3204447
13 bcpnn_mcmc      log2 0.5742360 6.657378e-02  0.3693870
14 bcpnn_mcmc      log2 0.5445855 7.346941e-02  0.3548644
15 bcpnn_mcmc      log2 0.5026231 8.027030e-02  0.3392577
16 bcpnn_mcmc      log2 0.4817174 8.696406e-02  0.3429426
17 bcpnn_mcmc      log2 0.4572070 9.354240e-02  0.3327737
18 bcpnn_mcmc      log2 0.4241559 1.000000e-01  0.3113718
```

```r
get_sig_tab_over_time(sra_cum$data[[11]])
```

```
     est_name est_scale       est alpha       ci_lo
1  bcpnn_mcmc      log2 0.3778604   0.1 -0.27276457
2  bcpnn_mcmc      log2 0.3314299   0.1 -0.18935042
3  bcpnn_mcmc      log2 0.3719247   0.1 -0.15817042
4  bcpnn_mcmc      log2 0.4794164   0.1  0.01891085
5  bcpnn_mcmc      log2 0.4505463   0.1 -0.02705216
6  bcpnn_mcmc      log2 0.5291089   0.1  0.32291402
7  bcpnn_mcmc      log2 0.5426635   0.1  0.33973052
8  bcpnn_mcmc      log2 0.5426635   0.1  0.33969117
9  bcpnn_mcmc      log2 0.5303613   0.1  0.32927472
10 bcpnn_mcmc      log2 0.5654356   0.1  0.36001100
11 bcpnn_mcmc      log2 0.5392350   0.1  0.34945147
12 bcpnn_mcmc      log2 0.5198015   0.1  0.34340898
13 bcpnn_mcmc      log2 0.5742360   0.1  0.38955969
14 bcpnn_mcmc      log2 0.5445855   0.1  0.36988520
15 bcpnn_mcmc      log2 0.5026231   0.1  0.34910679
```

```
16 bcpnn_mcmc      log2 0.4817174   0.1  0.34742783
17 bcpnn_mcmc      log2 0.4572070   0.1  0.33516223
18 bcpnn_mcmc      log2 0.4241559   0.1  0.31176932
```

```r
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    data =
      map(
        .x = data,
        .f = get_mult_compare_adj_alpha
      )
  )
toc()
```

```
0.17 sec elapsed
```

```r
# test
sra_cum$data[[11]] # check adj_alpha added as column in data
```

```
# A tibble: 18 x 6
   mnth       nA    nB    nC    nD adj_alpha
   <chr>   <dbl> <dbl> <dbl> <dbl>     <dbl>
 1 2013-Q3     4    12     1    10 0.0000572
 2 2013-Q4     5    15     1    10 0.001
 3 2014-Q1     5    15     1    11 0.00355
 4 2014-Q2     6    16     1    14 0.00756
 5 2014-Q3     8    18     3    21 0.0127
 6 2014-Q4    25    20     4    27 0.0185
 7 2015-Q1    26    20     4    28 0.0249
 8 2015-Q2    26    20     4    28 0.0316
 9 2015-Q3    26    21     4    28 0.0385
10 2015-Q4    26    21     4    30 0.0455
11 2016-Q1    29    22     4    30 0.0526
12 2016-Q2    33    22     4    30 0.0596
13 2016-Q3    33    22     5    35 0.0666
14 2016-Q4    35    24     5    35 0.0735
15 2017-Q1    44    24     6    36 0.0803
16 2017-Q2    57    25     6    39 0.0870
```

```
17 2017-Q3    67    25     6    40 0.0935
18 2017-Q4    76    26     6    40 0.1
```

```
### takes ~ 100 sec (i5-8400 6c/6t)
### takes ~55 sec on laptop lols (i5 8th gen 4c/8t)
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    sig_tab =
      future_map(
        .x = data,
        .f = get_sig_tab_over_time_2, # the alpha in data version
        .options = furrr_seed1
      )
  )
toc()
```

```
83.12 sec elapsed
```

```
# check
sra_cum$sig_tab[[11]]
```

```
     est_name est_scale       est        alpha      ci_lo
1  bcpnn_mcmc      log2 0.3778604 5.719516e-05 -2.4252741
2  bcpnn_mcmc      log2 0.3314299 1.000000e-03 -1.2692260
3  bcpnn_mcmc      log2 0.3719247 3.552305e-03 -0.9503514
4  bcpnn_mcmc      log2 0.4794164 7.562748e-03 -0.4816096
5  bcpnn_mcmc      log2 0.4505463 1.266582e-02 -0.3786932
6  bcpnn_mcmc      log2 0.5291089 1.853315e-02  0.2341781
7  bcpnn_mcmc      log2 0.5426635 2.491337e-02  0.2692897
8  bcpnn_mcmc      log2 0.5426635 3.162278e-02  0.2796775
9  bcpnn_mcmc      log2 0.5303613 3.852888e-02  0.2788266
10 bcpnn_mcmc      log2 0.5654356 4.553645e-02  0.3196894
11 bcpnn_mcmc      log2 0.5392350 5.257699e-02  0.3167301
12 bcpnn_mcmc      log2 0.5198015 5.960122e-02  0.3204761
13 bcpnn_mcmc      log2 0.5742360 6.657378e-02  0.3686319
14 bcpnn_mcmc      log2 0.5445855 7.346941e-02  0.3541094
15 bcpnn_mcmc      log2 0.5026231 8.027030e-02  0.3392261
16 bcpnn_mcmc      log2 0.4817174 8.696406e-02  0.3425696
```

```
17 bcpnn_mcmc      log2 0.4572070 9.354240e-02  0.3331946
18 bcpnn_mcmc      log2 0.4241559 1.000000e-01  0.3126421
```

```
sra_cum_bcpnn_mc_adj <-
  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_mc_adj
```

```
# A tibble: 1,707 x 15
   grps     dat_t~1 thresh mnth     nA    nB    nC    nD adj_a~2 est_n~3 est_s~4
   <chr>    <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>   <chr>
 1 (a) pel~ cumula~ 0.010  2013~     3     7     1     4 3.37e-5 bcpnn_~ log2
 2 (a) pel~ cumula~ 0.010  2013~     4    10     1     5 6.88e-4 bcpnn_~ log2
 3 (a) pel~ cumula~ 0.010  2013~     5    11     2     9 2.62e-3 bcpnn_~ log2
 4 (a) pel~ cumula~ 0.010  2013~     9    11     2     9 5.81e-3 bcpnn_~ log2
 5 (a) pel~ cumula~ 0.010  2014~     9    11     2    10 1   e-2 bcpnn_~ log2
 6 (a) pel~ cumula~ 0.010  2014~    10    12     3    12 1.49e-2 bcpnn_~ log2
 7 (a) pel~ cumula~ 0.010  2014~    12    14     5    19 2.04e-2 bcpnn_~ log2
 8 (a) pel~ cumula~ 0.010  2014~    30    15     7    24 2.62e-2 bcpnn_~ log2
 9 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.23e-2 bcpnn_~ log2
10 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.85e-2 bcpnn_~ log2
# ... with 1,697 more rows, 4 more variables: est <dbl>, alpha <dbl>,
#   ci_lo <dbl>, dte <date>, and abbreviated variable names 1: dat_type,
#   2: adj_alpha, 3: est_name, 4: est_scale
```

```
# first signif
bcpnn_mc_adj_signif <-
  sra_cum_bcpnn_mc_adj %>%
  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_mc_adj)
```

```
[1] 1707
```

```
sra_cum_bcpnn_mc_adj <-
  left_join(
    sra_cum_bcpnn_mc_adj,
    bcpnn_mc_adj_signif %>% select(grps, dat_type, thresh, dte_reach_sig),
    c("grps", "dat_type", "thresh")
  )
nrow(sra_cum_bcpnn_mc_adj)
```

```
[1] 1707
```

```
sra_cum_bcpnn_mc_adj
```

```
# A tibble: 1,707 x 16
   grps     dat_t~1 thresh mnth      nA    nB    nC    nD adj_a~2 est_n~3 est_s~4
   <chr>    <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>   <chr>
 1 (a) pel~ cumula~ 0.010  2013~     3     7     1     4 3.37e-5 bcpnn_~ log2
 2 (a) pel~ cumula~ 0.010  2013~     4    10     1     5 6.88e-4 bcpnn_~ log2
 3 (a) pel~ cumula~ 0.010  2013~     5    11     2     9 2.62e-3 bcpnn_~ log2
 4 (a) pel~ cumula~ 0.010  2013~     9    11     2     9 5.81e-3 bcpnn_~ log2
 5 (a) pel~ cumula~ 0.010  2014~     9    11     2    10 1   e-2 bcpnn_~ log2
 6 (a) pel~ cumula~ 0.010  2014~    10    12     3    12 1.49e-2 bcpnn_~ log2
 7 (a) pel~ cumula~ 0.010  2014~    12    14     5    19 2.04e-2 bcpnn_~ log2
 8 (a) pel~ cumula~ 0.010  2014~    30    15     7    24 2.62e-2 bcpnn_~ log2
 9 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.23e-2 bcpnn_~ log2
10 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.85e-2 bcpnn_~ log2
# ... with 1,697 more rows, 5 more variables: est <dbl>, alpha <dbl>,
#   ci_lo <dbl>, dte <date>, dte_reach_sig <date>, and abbreviated variable
#   names 1: dat_type, 2: adj_alpha, 3: est_name, 4: est_scale
```

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

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

2.3 PRR with mult comp adjust

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


sra_cum <-
  sra_cum %>%
  nest(data = c(mnth, nA, nB, nC, nD))



# test
get_mult_compare_adj_alpha(sra_cum$data[[11]])
```

```
# A tibble: 18 x 6
   mnth       nA    nB    nC    nD adj_alpha
   <chr>   <dbl> <dbl> <dbl> <dbl>     <dbl>
 1 2013-Q3     4    12     1    10 0.0000572
 2 2013-Q4     5    15     1    10 0.001
 3 2014-Q1     5    15     1    11 0.00355
 4 2014-Q2     6    16     1    14 0.00756
 5 2014-Q3     8    18     3    21 0.0127
 6 2014-Q4    25    20     4    27 0.0185
 7 2015-Q1    26    20     4    28 0.0249
 8 2015-Q2    26    20     4    28 0.0316
 9 2015-Q3    26    21     4    28 0.0385
10 2015-Q4    26    21     4    30 0.0455
11 2016-Q1    29    22     4    30 0.0526
12 2016-Q2    33    22     4    30 0.0596
13 2016-Q3    33    22     5    35 0.0666
14 2016-Q4    35    24     5    35 0.0735
15 2017-Q1    44    24     6    36 0.0803
16 2017-Q2    57    25     6    39 0.0870
17 2017-Q3    67    25     6    40 0.0935
18 2017-Q4    76    26     6    40 0.1
```

```r
get_sig_tab_over_time_2(get_mult_compare_adj_alpha(sra_cum$data[[11]]))
```

```
     est_name est_scale       est        alpha      ci_lo
1  bcpnn_mcmc      log2 0.3778604 5.719516e-05 -2.3823226
2  bcpnn_mcmc      log2 0.3314299 1.000000e-03 -1.3454188
3  bcpnn_mcmc      log2 0.3719247 3.552305e-03 -0.9331261
4  bcpnn_mcmc      log2 0.4794164 7.562748e-03 -0.4892387
5  bcpnn_mcmc      log2 0.4505463 1.266582e-02 -0.3800232
6  bcpnn_mcmc      log2 0.5291089 1.853315e-02  0.2342780
7  bcpnn_mcmc      log2 0.5426635 2.491337e-02  0.2666768
8  bcpnn_mcmc      log2 0.5426635 3.162278e-02  0.2782165
9  bcpnn_mcmc      log2 0.5303613 3.852888e-02  0.2785088
10 bcpnn_mcmc      log2 0.5654356 4.553645e-02  0.3178754
11 bcpnn_mcmc      log2 0.5392350 5.257699e-02  0.3159453
12 bcpnn_mcmc      log2 0.5198015 5.960122e-02  0.3207910
13 bcpnn_mcmc      log2 0.5742360 6.657378e-02  0.3682926
14 bcpnn_mcmc      log2 0.5445855 7.346941e-02  0.3535223
15 bcpnn_mcmc      log2 0.5026231 8.027030e-02  0.3395410
16 bcpnn_mcmc      log2 0.4817174 8.696406e-02  0.3430154
17 bcpnn_mcmc      log2 0.4572070 9.354240e-02  0.3327655
18 bcpnn_mcmc      log2 0.4241559 1.000000e-01  0.3124256
```

```r
get_sig_tab_over_time_2(get_mult_compare_adj_alpha(sra_cum$data[[11]]), method = "prr")
```

```
   est_name  est_scale      est        alpha      ci_lo
1       prr orig scale 2.750000 5.719516e-05 0.04066616
2       prr orig scale 2.750000 1.000000e-03 0.09303769
3       prr orig scale 3.000000 3.552305e-03 0.14772511
4       prr orig scale 4.090909 7.562748e-03 0.26339441
5       prr orig scale 2.461538 1.266582e-02 0.53132229
6       prr orig scale 4.305556 1.853315e-02 1.37316395
7       prr orig scale 4.521739 2.491337e-02 1.52286634
8       prr orig scale 4.521739 3.162278e-02 1.59362698
9       prr orig scale 4.425532 3.852888e-02 1.61983333
10      prr orig scale 4.702128 4.553645e-02 1.77347101
11      prr orig scale 4.833333 5.257699e-02 1.88692159
12      prr orig scale 5.100000 5.960122e-02 2.05551981
13      prr orig scale 4.800000 6.657378e-02 2.17067746
14      prr orig scale 4.745763 7.346941e-02 2.19025371
15      prr orig scale 4.529412 8.027030e-02 2.29602631
```

```
16      prr orig scale 5.213415 8.696406e-02 2.68795062
17      prr orig scale 5.583333 9.354240e-02 2.92265162
18      prr orig scale 5.712418 1.000000e-01 3.03209290
```

```
get_sig_tab_over_time(sra_cum$data[[11]], method = "prr")
```

```
   est_name  est_scale      est alpha     ci_lo
1       prr orig scale 2.750000   0.1 0.4912212
2       prr orig scale 2.750000   0.1 0.5060331
3       prr orig scale 3.000000   0.1 0.5487100
4       prr orig scale 4.090909   0.1 0.7555189
5       prr orig scale 2.461538   0.1 0.8951429
6       prr orig scale 4.305556   0.1 1.9379924
7       prr orig scale 4.521739   0.1 2.0354629
8       prr orig scale 4.521739   0.1 2.0354629
9       prr orig scale 4.425532   0.1 1.9905948
10      prr orig scale 4.702128   0.1 2.1084471
11      prr orig scale 4.833333   0.1 2.1757613
12      prr orig scale 5.100000   0.1 2.3065749
13      prr orig scale 4.800000   0.1 2.3563014
14      prr orig scale 4.745763   0.1 2.3318890
15      prr orig scale 4.529412   0.1 2.3909246
16      prr orig scale 5.213415   0.1 2.7583297
17      prr orig scale 5.583333   0.1 2.9591422
18      prr orig scale 5.712418   0.1 3.0320929
```

```
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    data =
      map(
        .x = data,
        .f = get_mult_compare_adj_alpha
      )
  )
toc()
```

```
0.17 sec elapsed
```

```
# test
sra_cum$data[[11]] # check adj_alpha added as column in data
```

```
# A tibble: 18 x 6
   mnth       nA    nB    nC    nD adj_alpha
   <chr>   <dbl> <dbl> <dbl> <dbl>     <dbl>
 1 2013-Q3     4    12     1    10 0.0000572
 2 2013-Q4     5    15     1    10 0.001
 3 2014-Q1     5    15     1    11 0.00355
 4 2014-Q2     6    16     1    14 0.00756
 5 2014-Q3     8    18     3    21 0.0127
 6 2014-Q4    25    20     4    27 0.0185
 7 2015-Q1    26    20     4    28 0.0249
 8 2015-Q2    26    20     4    28 0.0316
 9 2015-Q3    26    21     4    28 0.0385
10 2015-Q4    26    21     4    30 0.0455
11 2016-Q1    29    22     4    30 0.0526
12 2016-Q2    33    22     4    30 0.0596
13 2016-Q3    33    22     5    35 0.0666
14 2016-Q4    35    24     5    35 0.0735
15 2017-Q1    44    24     6    36 0.0803
16 2017-Q2    57    25     6    39 0.0870
17 2017-Q3    67    25     6    40 0.0935
18 2017-Q4    76    26     6    40 0.1
```

```
get_sig_tab_over_time_2_prr <- function(dat) {
  get_sig_tab_over_time_2(dat, method = "prr")
}


### takes ~2 sec on laptop (i5 8th gen 4c/8t)
tic()
sra_cum <-
  sra_cum %>%
  mutate(
    sig_tab =
      future_map(
        .x = data,
        .f = get_sig_tab_over_time_2_prr, # the alpha in data version
        .options = furrr_seed1
      )
  )
toc()
```

```
3.37 sec elapsed
```

```
# check
sra_cum$sig_tab[[11]]
```

```
   est_name  est_scale      est         alpha      ci_lo
1       prr orig scale 2.750000 5.719516e-05 0.04066616
2       prr orig scale 2.750000 1.000000e-03 0.09303769
3       prr orig scale 3.000000 3.552305e-03 0.14772511
4       prr orig scale 4.090909 7.562748e-03 0.26339441
5       prr orig scale 2.461538 1.266582e-02 0.53132229
6       prr orig scale 4.305556 1.853315e-02 1.37316395
7       prr orig scale 4.521739 2.491337e-02 1.52286634
8       prr orig scale 4.521739 3.162278e-02 1.59362698
9       prr orig scale 4.425532 3.852888e-02 1.61983333
10      prr orig scale 4.702128 4.553645e-02 1.77347101
11      prr orig scale 4.833333 5.257699e-02 1.88692159
12      prr orig scale 5.100000 5.960122e-02 2.05551981
13      prr orig scale 4.800000 6.657378e-02 2.17067746
14      prr orig scale 4.745763 7.346941e-02 2.19025371
15      prr orig scale 4.529412 8.027030e-02 2.29602631
16      prr orig scale 5.213415 8.696406e-02 2.68795062
17      prr orig scale 5.583333 9.354240e-02 2.92265162
18      prr orig scale 5.712418 1.000000e-01 3.03209290
```

```
sra_cum_prr_mc_adj <-
  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_prr_mc_adj
```

```
# A tibble: 1,707 x 15
   grps     dat_t~1 thresh mnth     nA    nB    nC    nD adj_a~2 est_n~3 est_s~4
   <chr>    <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>   <chr>
 1 (a) pel~ cumula~ 0.010  2013~     3     7     1     4 3.37e-5 prr     orig s~
 2 (a) pel~ cumula~ 0.010  2013~     4    10     1     5 6.88e-4 prr     orig s~
 3 (a) pel~ cumula~ 0.010  2013~     5    11     2     9 2.62e-3 prr     orig s~
 4 (a) pel~ cumula~ 0.010  2013~     9    11     2     9 5.81e-3 prr     orig s~
 5 (a) pel~ cumula~ 0.010  2014~     9    11     2    10 1   e-2 prr     orig s~
 6 (a) pel~ cumula~ 0.010  2014~    10    12     3    12 1.49e-2 prr     orig s~
 7 (a) pel~ cumula~ 0.010  2014~    12    14     5    19 2.04e-2 prr     orig s~
 8 (a) pel~ cumula~ 0.010  2014~    30    15     7    24 2.62e-2 prr     orig s~
 9 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.23e-2 prr     orig s~
10 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.85e-2 prr     orig s~
# ... with 1,697 more rows, 4 more variables: est <dbl>, alpha <dbl>,
#   ci_lo <dbl>, dte <date>, and abbreviated variable names 1: dat_type,
#   2: adj_alpha, 3: est_name, 4: est_scale
```

```
# first signif
prr_mc_adj_signif <-
  sra_cum_prr_mc_adj %>%
  group_by(grps, dat_type, thresh) %>%
  dplyr::filter(ci_lo > 1) %>% # 1 is the critical value on ratio scale
  arrange(dte) %>%
  dplyr::filter(row_number() == 1) %>%
  ungroup() %>%
  rename(dte_reach_sig = dte)


nrow(sra_cum_prr_mc_adj)
```

```
[1] 1707
```

```
sra_cum_prr_mc_adj <-
  left_join(
    sra_cum_prr_mc_adj,
    prr_mc_adj_signif %>% select(grps, dat_type, thresh, dte_reach_sig),
    c("grps", "dat_type", "thresh")
  )
nrow(sra_cum_prr_mc_adj)
```

```
[1] 1707
```

```
sra_cum_prr_mc_adj
```

```
# A tibble: 1,707 x 16
   grps     dat_t~1 thresh mnth     nA    nB    nC    nD adj_a~2 est_n~3 est_s~4
   <chr>    <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>   <chr>
 1 (a) pel~ cumula~ 0.010  2013~     3     7     1     4 3.37e-5 prr     orig s~
 2 (a) pel~ cumula~ 0.010  2013~     4    10     1     5 6.88e-4 prr     orig s~
 3 (a) pel~ cumula~ 0.010  2013~     5    11     2     9 2.62e-3 prr     orig s~
 4 (a) pel~ cumula~ 0.010  2013~     9    11     2     9 5.81e-3 prr     orig s~
 5 (a) pel~ cumula~ 0.010  2014~     9    11     2    10 1    e-2 prr     orig s~
 6 (a) pel~ cumula~ 0.010  2014~    10    12     3    12 1.49e-2 prr     orig s~
 7 (a) pel~ cumula~ 0.010  2014~    12    14     5    19 2.04e-2 prr     orig s~
 8 (a) pel~ cumula~ 0.010  2014~    30    15     7    24 2.62e-2 prr     orig s~
 9 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.23e-2 prr     orig s~
10 (a) pel~ cumula~ 0.010  2015~    31    15     7    25 3.85e-2 prr     orig s~
# ... with 1,697 more rows, 5 more variables: est <dbl>, alpha <dbl>,
#   ci_lo <dbl>, dte <date>, dte_reach_sig <date>, and abbreviated variable
#   names 1: dat_type, 2: adj_alpha, 3: est_name, 4: est_scale
```

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

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

2.4 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
(a) pelvic_mesh v hernia_mesh	0.025	2013-Q3	2017-Q4	18	82	10	92	18	5.1	0.1

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	r
(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
(b) pelvic_mesh v hernia_mesh/other_mesh	0.030	2012-Q4	2017-Q4	21	82	55	137	21	6.5	0.7

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	r
(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
(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

grps	thresh	min_dte	max_dte	nrows	sum_nA	sum_nC	tot_n	nqtrs	n_per_qtr	z
(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
(d) hernia_mesh v other_mesh	0.040	2013-Q3	2017-Q4	18	9	42	51	18	2.8	4.7

grps	thresh	min_dte	max_dte	rows	sum_nA	sum_nC	tot_n	qtrs	n_per_qtr	r
(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
(e) hernia_mesh/other_mesh v other_device	0.045	2012-Q4	2017-Q4	21	47	1609	1656	21	78.9	34.2

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	0.050	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~1     z
  <chr>     <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>   <dbl> <dbl>
1 (a) pelv~ 0.010  2013-Q1 2017-Q4    20     82     12    94    20     4.7 0.146
# ... with abbreviated variable name 1: n_per_qtr
```

```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
```

```
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~1     z
  <chr>     <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>   <dbl> <dbl>
1 (c) pelv~ 0.065  2012-Q2 2017-Q4    23     75   1331  1406    23    61.1  17.7
# ... with abbreviated variable name 1: n_per_qtr
```

```
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] 0.048193
```

```
### 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()
```

```
176.14 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~1     z
   <chr>     <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>   <dbl> <dbl>
 1 (a) pel~ 0.010  2013-Q1 2017-Q4    20     82     12    94    20    4.7  0.146
 2 (a) pel~ 0.015  2013-Q1 2017-Q4    20     82     12    94    20    4.7  0.146
 3 (a) pel~ 0.020  2013-Q3 2017-Q4    18     82     10    92    18    5.11 0.122
 4 (a) pel~ 0.025  2013-Q3 2017-Q4    18     82     10    92    18    5.11 0.122
 5 (a) pel~ 0.030  2013-Q3 2017-Q4    18     82     10    92    18    5.11 0.122
 6 (a) pel~ 0.035  2013-Q3 2017-Q4    18     82     10    92    18    5.11 0.122
 7 (a) pel~ 0.040  2013-Q3 2017-Q4    18     81      9    90    18    5    0.111
 8 (a) pel~ 0.045  2013-Q3 2017-Q4    18     79      8    87    18    4.83 0.101
 9 (a) pel~ 0.050  2013-Q3 2017-Q4    18     77      8    85    18    4.72 0.104
10 (a) pel~ 0.055  2013-Q3 2017-Q4    18     77      8    85    18    4.72 0.104
# ... with 85 more rows, 1 more variable: cv <dbl>, and abbreviated variable
#   name 1: n_per_qtr
```

```
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~1      z
  <chr>     <chr>  <chr>   <chr>   <int>  <dbl>  <dbl> <dbl> <int>   <dbl>  <dbl>
1 (a) pel~ 0.085  2014-Q3 2017-Q4    14     71      2    73    14    5.21 0.0282
2 (a) pel~ 0.090  2017-Q1 2017-Q4     4     70      1    71     4   17.8  0.0143
3 (a) pel~ 0.095  2017-Q1 2017-Q4     4     69      1    70     4   17.5  0.0145
4 (a) pel~ 0.100  2017-Q1 2017-Q4     4     69      1    70     4   17.5  0.0145
# ... with 1 more variable: cv <dbl>, and abbreviated variable name
#   1: n_per_qtr
```

```
# remove analyses where thresholds don't allow enough events (extreme threshold values)
# cv_tab <- cv_tab %>% dplyr::filter(!is.na(cv))

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
```

```r
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_t~1 thresh mnth     nA    nB    nC    nD maxllr   rre    cv reach~2
   <chr> <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>   <int>
 1 (a) ~ cumula~ 0.085  2014~     6    20     1    23   1.79  5.54    NA       0
 2 (a) ~ cumula~ 0.085  2014~    23    22     1    30   8.79 15.8     NA       0
 3 (a) ~ cumula~ 0.085  2015~    24    22     1    31   9.37 16.7     NA       0
 4 (a) ~ cumula~ 0.085  2015~    24    22     1    31   9.37 16.7     NA       0
 5 (a) ~ cumula~ 0.085  2015~    24    23     1    31   9.17 16.3     NA       0
 6 (a) ~ cumula~ 0.085  2015~    24    23     1    31   9.17 16.3     NA       0
 7 (a) ~ cumula~ 0.085  2016~    31    24     1    33  11.4  19.2     NA       0
 8 (a) ~ cumula~ 0.085  2016~    31    24     1    33  11.4  19.2     NA       0
 9 (a) ~ cumula~ 0.085  2016~    33    26     1    39  13.5  22.4     NA       0
10 (a) ~ cumula~ 0.085  2016~    33    26     1    39  13.5  22.4     NA       0
# ... with 16 more rows, 1 more variable: dte <date>, and abbreviated variable
#   names 1: dat_type, 2: reached_cv
# 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 reach~1
   <chr>         <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>   <int>
 1 (a) pelvic_m~ 0.010  2013~     3     7     1     4 0.0657  1.5   3.28       0
 2 (a) pelvic_m~ 0.010  2013~     4    10     1     5 0.129   1.71  3.28       0
 3 (a) pelvic_m~ 0.010  2013~     5    11     2     9 0.224   1.72  3.28       0
 4 (a) pelvic_m~ 0.010  2013~     9    11     2     9 0.801   2.48  3.28       0
 5 (a) pelvic_m~ 0.010  2014~     9    11     2    10 0.976   2.7   3.28       0
 6 (a) pelvic_m~ 0.010  2014~    10    12     3    12 0.885   2.27  3.28       0
 7 (a) pelvic_m~ 0.010  2014~    12    14     5    19 1.22    2.22  3.28       0
 8 (a) pelvic_m~ 0.010  2014~    30    15     7    24 4.05    2.95  3.28       1
 9 (a) pelvic_m~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28       1
10 (a) pelvic_m~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28       1
11 (a) pelvic_m~ 0.010  2015~    31    16     7    25 4.27    3.02  3.28       1
12 (a) pelvic_m~ 0.010  2015~    31    16     9    25 3.36    2.49  3.28       1
13 (a) pelvic_m~ 0.010  2016~    35    16     9    25 3.83    2.59  3.28       1
14 (a) pelvic_m~ 0.010  2016~    39    16     9    25 4.27    2.68  3.28       1
15 (a) pelvic_m~ 0.010  2016~    39    16    10    30 5.17    2.84  3.28       1
16 (a) pelvic_m~ 0.010  2016~    41    18    10    30 5.04    2.78  3.28       1
17 (a) pelvic_m~ 0.010  2017~    50    18    11    31 5.85    2.81  3.28       1
18 (a) pelvic_m~ 0.010  2017~    63    19    12    33 7.04    2.88  3.28       1
19 (a) pelvic_m~ 0.010  2017~    73    19    12    34 8.18    3.04  3.28       1
20 (a) pelvic_m~ 0.010  2017~    82    20    12    34 8.65    3.08  3.28       1
21 (a) pelvic_m~ 0.015  2013~     3     7     1     4 0.0657  1.5   3.28       0
22 (a) pelvic_m~ 0.015  2013~     4    10     1     5 0.129   1.71  3.28       0
23 (a) pelvic_m~ 0.015  2013~     5    11     2     9 0.224   1.72  3.28       0
24 (a) pelvic_m~ 0.015  2013~     9    11     2     9 0.801   2.48  3.28       0
25 (a) pelvic_m~ 0.015  2014~     9    11     2    10 0.976   2.7   3.28       0
# ... with 1,682 more rows, 1 more variable: dte <date>, and abbreviated
#   variable name 1: reached_cv
```

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


nrow(maxsprt_dat)
```

```
[1] 1707
```

```
maxsprt_dat <-
  left_join(
    maxsprt_dat,
    maxsprt_signif %>% select(grps, dat_type, thresh, dte_reach_sig),
    c("grps", "dat_type", "thresh")
  )
nrow(maxsprt_dat)
```

```
[1] 1707
```

```
maxsprt_dat
```

```
# A tibble: 1,707 x 14
   grps  dat_t~1 thresh mnth     nA    nB    nC    nD maxllr   rre    cv reach~2
   <chr> <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>   <int>
 1 (a) ~ cumula~ 0.010  2013~     3     7     1     4 0.0657  1.5   3.28       0
 2 (a) ~ cumula~ 0.010  2013~     4    10     1     5 0.129   1.71  3.28       0
 3 (a) ~ cumula~ 0.010  2013~     5    11     2     9 0.224   1.72  3.28       0
 4 (a) ~ cumula~ 0.010  2013~     9    11     2     9 0.801   2.48  3.28       0
 5 (a) ~ cumula~ 0.010  2014~     9    11     2    10 0.976   2.7   3.28       0
 6 (a) ~ cumula~ 0.010  2014~    10    12     3    12 0.885   2.27  3.28       0
 7 (a) ~ cumula~ 0.010  2014~    12    14     5    19 1.22    2.22  3.28       0
 8 (a) ~ cumula~ 0.010  2014~    30    15     7    24 4.05    2.95  3.28       1
 9 (a) ~ cumula~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28       1
10 (a) ~ cumula~ 0.010  2015~    31    15     7    25 4.45    3.08  3.28       1
# ... with 1,697 more rows, 2 more variables: dte <date>, dte_reach_sig <date>,
#   and abbreviated variable names 1: dat_type, 2: reached_cv
```

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

# these are where the maxllr has dropped under the CV after exceeding it previously
maxsprt_dat %>%
  dplyr::filter(
    is.na(reach_sig) |
      is.na(reached_cv) |
      (as.logical(reached_cv) != reach_sig)
  )
```

```
# A tibble: 5 x 15
  grps   dat_t~1 thresh mnth     nA    nB    nC    nD maxllr   rre    cv reach~2
  <chr>  <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>   <int>
1 (c) p~ cumula~ 0.085  2013~     3     7    84  1449   2.61  5.48  2.88       0
2 (c) p~ cumula~ 0.085  2013~     3     7    84  1449   2.61  5.48  2.88       0
3 (c) p~ cumula~ 0.085  2013~     3     7    84  1449   2.61  5.48  2.88       0
4 (c) p~ cumula~ 0.085  2013~     3     7    84  1449   2.61  5.48  2.88       0
5 (e) h~ cumula~ 0.065  2015~    14    60   772  7436   2.71  2.01  2.71       0
# ... with 3 more variables: dte <date>, dte_reach_sig <date>, reach_sig <lgl>,
#   and abbreviated variable names 1: dat_type, 2: reached_cv
```

```r
maxsprt_dat <-
  maxsprt_dat %>%
  select(-reached_cv)
```

```r
maxsprt_dat %>%
  write_parquet(., sink = "out/sra_cum_maxsprt.parquet")
```

3 Ready plot data

```
plt_dat <-
  bind_rows(
    maxsprt_dat %>%
      select(comparator, dte, cv, reached_cv, val = maxllr) %>%
      mutate(stat = "MaxSPRT (max LLR)"),
    bcpnn_data %>%
      select(comparator, dte, val = ci_lo) %>%
      mutate(cv = 0, reached_cv = as.integer(val > cv), stat = "IC (BCPNN, Lower 95% CI)")
  )


sig_reach_dat <-
  plt_dat %>%
  arrange(stat, comparator, dte) %>%
  group_by(stat, comparator) %>%
  dplyr::filter(reached_cv == 1) %>%
  dplyr::filter(row_number() == 1) %>%
  select(stat, comparator, dte_reached = dte) %>%
  # now create separation between reached CV values when it occurs
  group_by(stat, dte_reached) %>%
  mutate(rep_dte = 1:n()) %>%
  ungroup() %>%
  mutate(dte_reached = dte_reached + days(10 * (rep_dte - 1))) %>%
  select(-rep_dte)


plt_dat <-
  left_join(
    plt_dat,
    sig_reach_dat,
    c("stat", "comparator")
  )

plt_dat %>%
  ggplot(., aes(x = dte, y = val, col = comparator )) +
  geom_hline(aes(yintercept = cv), alpha = 0.5) +
  geom_vline(aes(xintercept = dte_reached, col = comparator), alpha = 0.5) +
  geom_line(alpha = 0.5) +
  geom_point() +
```

```r
facet_wrap(~ stat, ncol = 1, scales = "free_y") +
scale_colour_tableau() +
theme_bw() +
labs(
  x = "Quarter",
  y = "Statistic",
  col = "Comparison"
)
```

4 Session information

```r
# Sys.info()[!(names(Sys.info()) %in% c("login", "nodename"))] %>%
#   as.data.frame(.)
format(Sys.time(), '%d %b %Y')
```

```
[1] "05 Nov 2023"
```

```r
sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

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_3.5.0
 [5] knitr_1.42        ggrepel_0.9.3     ggplot2_3.4.1     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_1.1.0       readr_2.1.4

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.10                lattice_0.20-45
 [3] listenv_0.9.0              assertthat_0.2.1
 [5] digest_0.6.31              utf8_1.2.3
 [7] parallelly_1.36.0          R6_2.5.1
 [9] MatrixModels_0.5-1         evaluate_0.20
[11] coda_0.19-4                pillar_1.8.1
[13] rlang_1.0.6                rstudioapi_0.14
[15] SparseM_1.81               Matrix_1.5-1
[17] rmarkdown_2.20             splines_4.2.2
```

```
[19] bit_4.0.5                  munsell_0.5.0
[21] compiler_4.2.2             xfun_0.37
[23] pkgconfig_2.0.3            EmpiricalCalibration_3.1.1
[25] globals_0.16.2             mcmc_0.9-7
[27] htmltools_0.5.4            tidyselect_1.2.0
[29] tibble_3.1.8               codetools_0.2-18
[31] fansi_1.0.4                tzdb_0.3.0
[33] withr_2.5.0                MASS_7.3-58.1
[35] grid_4.2.2                 jsonlite_1.8.4
[37] xtable_1.8-4               gtable_0.3.1
[39] lifecycle_1.0.3            magrittr_2.0.3
[41] scales_1.2.1               cli_3.6.0
[43] xml2_1.3.3                 ellipsis_0.3.2
[45] generics_0.1.3             vctrs_0.5.2
[47] boot_1.3-28                iterators_1.0.14
[49] tools_4.2.2                bit64_4.0.5
[51] glue_1.6.2                 hms_1.1.2
[53] parallel_4.2.2             fastmap_1.1.0
[55] survival_3.4-0             yaml_2.3.7
[57] timechange_0.2.0           colorspace_2.1-0
[59] Sequential_4.3.1           gt_0.10.0
[61] quantreg_5.94              MCMCpack_1.6-3
```