# 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 '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

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

```r
# 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
```

```
6.12 sec elapsed
```

```r
# 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
```

```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)
```

## 1.3 Constants

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

## 1.4 Functions

```
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

```
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
# ... with 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
```

```r
### 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()
```

```
112.24 sec elapsed
```

```r
# 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_t~1 thresh mnth     nA    nB    nC    nD est_n~2 est_s~3     est
   <chr>    <chr>    <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <chr>   <chr>     <dbl>
 1 pelvic_~ cumula~   0.01 2013~     3     7     1     2 bcpnn_~ log2    -0.0316
 2 pelvic_~ cumula~   0.01 2013~     3     7     1     4 bcpnn_~ log2     0.144
 3 pelvic_~ cumula~   0.01 2013~     3     7     1     5 bcpnn_~ log2     0.222
 4 pelvic_~ cumula~   0.01 2013~     4    10     1     5 bcpnn_~ log2     0.170
 5 pelvic_~ cumula~   0.01 2013~     4    11     1     7 bcpnn_~ log2     0.259
 6 pelvic_~ cumula~   0.01 2013~     5    11     1     7 bcpnn_~ log2     0.289
 7 pelvic_~ cumula~   0.01 2013~     5    11     2     9 bcpnn_~ log2     0.243
 8 pelvic_~ cumula~   0.01 2013~     8    11     2     9 bcpnn_~ log2     0.315
 9 pelvic_~ cumula~   0.01 2013~     9    11     2     9 bcpnn_~ log2     0.323
10 pelvic_~ cumula~   0.01 2014~     9    11     2    10 bcpnn_~ log2     0.365
# ... with 4,513 more rows, 3 more variables: ci_lo <dbl>, ci_hi <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] 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_t~1 thresh mnth      nA    nB    nC    nD est_n~2 est_s~3     est
   <chr>    <chr>    <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <chr>   <chr>     <dbl>
 1 pelvic_~ cumula~   0.01 2013~     3     7     1     2 bcpnn_~ log2    -0.0316
 2 pelvic_~ cumula~   0.01 2013~     3     7     1     4 bcpnn_~ log2     0.144
 3 pelvic_~ cumula~   0.01 2013~     3     7     1     5 bcpnn_~ log2     0.222
 4 pelvic_~ cumula~   0.01 2013~     4    10     1     5 bcpnn_~ log2     0.170
 5 pelvic_~ cumula~   0.01 2013~     4    11     1     7 bcpnn_~ log2     0.259
 6 pelvic_~ cumula~   0.01 2013~     5    11     1     7 bcpnn_~ log2     0.289
 7 pelvic_~ cumula~   0.01 2013~     5    11     2     9 bcpnn_~ log2     0.243
 8 pelvic_~ cumula~   0.01 2013~     8    11     2     9 bcpnn_~ log2     0.315
 9 pelvic_~ cumula~   0.01 2013~     9    11     2     9 bcpnn_~ log2     0.323
10 pelvic_~ cumula~   0.01 2014~     9    11     2    10 bcpnn_~ log2     0.365
# ... with 4,513 more rows, 4 more variables: ci_lo <dbl>, ci_hi <dbl>,
#   dte <date>, dte_reach_sig <date>, and abbreviated variable names
#   1: dat_type, 2: est_name, 3: est_scale
```

```
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

## 3 Session information

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


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

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

```
[1] "19 Jun 2023"
```

```r
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
```

```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 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_1.1.0       readr_2.1.4

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.0   xfun_0.37          listenv_0.9.0      splines_4.2.2
 [5] lattice_0.20-45    colorspace_2.1-0   vctrs_0.5.2        generics_0.1.3
 [9] htmltools_0.5.4    yaml_2.3.7         survival_3.4-0     MCMCpack_1.6-3
[13] utf8_1.2.3         rlang_1.0.6        pillar_1.8.1       glue_1.6.2
[17] withr_2.5.0        bit64_4.0.5        lifecycle_1.0.3    MatrixModels_0.5-1
[21] munsell_0.5.0      gtable_0.3.1       coda_0.19-4        codetools_0.2-18
[25] evaluate_0.20      SparseM_1.81       tzdb_0.3.0         fastmap_1.1.0
[29] quantreg_5.94      parallel_4.2.2     fansi_1.0.4        Rcpp_1.0.10
[33] xtable_1.8-4       scales_1.2.1       jsonlite_1.8.4     parallelly_1.34.0
[37] bit_4.0.5          mcmc_0.9-7         hms_1.1.2          digest_0.6.31
[41] grid_4.2.2         cli_3.6.0          tools_4.2.2        magrittr_2.0.3
[45] tibble_3.1.8       pkgconfig_2.0.3    Matrix_1.5-3       MASS_7.3-58.1
[49] ellipsis_0.3.2     timechange_0.2.0   assertthat_0.2.1   rmarkdown_2.20
[53] rstudioapi_0.14    iterators_1.0.14   R6_2.5.1           globals_0.16.2
[57] compiler_4.2.2
```