Delayed effect asymptotic approximations and simulation

Spending time examples

Keaven Anderson

2025-09-18

Contents

1 Overview

We consider asymptotic approximations and corresponding simulations for group sequential designs with delayed treatment effects. We demonstrate:

- the importance of both adequate events and adequate follow-up duration to ensure power in a fixed design, and
- the importance of guaranteeing a reasonable amount of α-spending for the final analysis in a group sequential design.

We demonstrate the concept of spending time as an effective way to adapt. Traditionally Lan and DeMets (1983), spending has been done according to targeting a specific number of events for an outcome at the end of the trial. However, for delayed treatment effect scenarios there is substantial literature (e.g., Lin et al. 2020; Roychoudhury et al. 2021) documenting the importance of adequate follow-up duration in addition to requiring an adequate number of events under the traditional proportional hazards assumption.

While other approaches could be taken, we have found the spending time approach generalizes well for addressing a variety of scenarios. The fact that spending does not need to correspond to information fraction was perhaps first raised by Lan and DeMets (1989) where calendar-time spending was discussed. However, we note that Michael A. Proschan, Lan, and Wittes (2006) have raised other scenarios where spending alternatives are considered. These approaches are generally more complicated to implement than the spending time approach suggested here. Two specific spending alternative approaches are examined here:

- Spending according to the minimum of planned and observed event counts. This is suggested for the delayed effect examples.
- Spending fixed amounts at analyses as suggested by Fleming, Harrington, and O'Brien (1984).

Regardless of the spending approach, it is important that timing of analyses be pre-defined in a way that is independent of observed treatment effect. This independence assumption is required to use the usual asymptotic normal calculations for computing group sequential boundary crossing probabilities. As noted by Michael A. Proschan, Follmann, and Waclawiw (1992), issues arising due to nefarious strategies when treatment differences are known at interim analysis can be largely limited by appropriate approaches. We revisit these for the α-spending approach that we extend here and the fixed interim α-spending method of Fleming, Harrington, and O'Brien (1984). Asymptotic distribution theory is always determined by statistical information at analyses that information is proportional to event counts for the logrank test. However, spending only needs to increase with event counts and there is no requirement that spending be a direct function of event count. While spending will always be an implicit or indirect function of event count, the function can be much easier to express by using the concept of spending time as in the cases noted above.

2 Computing preliminaries

Attach packages.

```
library(gsDesign)
library(gsDesign2)
library(tibble)
library(dplyr)
library(gt)
library(ggplot2)
library(parallelly)
library(doFuture)
library(purrr)
library(simtrial)
library(tictoc)
library(cowplot)
```

```
# Set to TRUE if large simulations are to be run and saved.
# Set to FALSE if simulations have already been run and saved and you want to load them.
run_simulations <- TRUE
```

Set up parallel computing.

```
cores_used <- 32
message("Parallelizing across ", cores_used, " CPUs")

# Set up the future plan to use the specified number of parallel workers
plan("multisession", workers = cores_used)
```

Set seed and number of simulations.

```
# Set a seed for reproducibility and use `.options.future = list(seed = TRUE)`
# for parallel-safe RNG. Use foreach + %dofuture% to perform parallel computation.
```

```r
set.seed(42)
nsim <- 100000
```

3 Design assumptions

We begin with assumptions for the group sequential design as well as for alternate scenarios. The approach taken for the asymptotic normal approximation for the logrank test uses an average hazard ratio approach for approximating treatment effect (Mukhopadhyay et al. 2020). When group sequential analyses are performed, we justify the joint distribution of logrank tests by the theory of Tsiatis (1982). Simulations are easily run for each scenario to verify asymptotic approximations of operating characteristics.

```r
# Only one enrollment assumption used
# This could be important to vary
enrollRates <- define_enroll_rate(duration = 12, rate = 1)
# Planned study duration
analysisTimes <- 36
# Constant dropout rate across all scenarios
dropoutRate <- 0.001
# 1-sided Type I error
alpha <- 0.025
# Targeted Type II error (1 - power) for original plan (scenario 1)
beta <- 0.1
scenarios <- tribble(
  ~i, ~ControlMedian, ~Delay, ~hr1, ~hr2,              ~Scenario,
   1,             12,      4,    1,  0.6,        "Original plan",
   2,             10,      6,  1.2,  0.5, "Events accrue faster",
   3,             16,      6,    1, 0.55, "Events accrue slower",
   4,             12,      4,  0.7,  0.7, "Proportional hazards",
   5,             12,      4,    1,    1,      "Null hypothesis"
)
```

We consider 5 failure rate scenarios for group sequential design. Control observations are always exponential with a scenario-specific failure rate. Experimental observations are piecewise exponential with a hazard ratio `hr1` for duration `Delay` (Delay) and hazard ratio `hr2` thereafter.

```r
scenarios |>
  transmute(
    Scenario = factor(Scenario),
    "Control Median" = factor(ControlMedian),
    Delay = factor(Delay),
    hr1 = factor(hr1),
    hr2 = factor(hr2)
  ) |>
  gt(groupname_col = NULL) |>
  fmt_number(columns = c(`Control Median`, Delay), decimals = 0) |>
  fmt_number(columns = c(hr1, hr2), decimals = 2, drop_trailing_zeros = TRUE) |>
  cols_align(align = "center", columns = everything()) |>
  tab_header(title = "Table 1: Failure rate scenarios studied") |>
  tab_options(table.font.size = px(12))
```

Table 1: Failure rate scenarios studied

Scenario	Control Median	Delay	hr1	hr2
Original plan	12	4	1	0.6
Events accrue faster	10	6	1.2	0.5
Events accrue slower	16	6	1	0.55
Proportional hazards	12	4	0.7	0.7
Null hypothesis	12	4	1	1

4 Fixed design

```
# Fixed design, single stratum
# Find sample size for 36 month trial under given
# enrollment and sample size assumptions
failRates <- define_fail_rate(
  duration = c(scenarios$Delay[1], 100),
  fail_rate = log(2) / scenarios$ControlMedian[1],
  hr = c(scenarios$hr1[1], scenarios$hr2[1]),
  dropout_rate = dropoutRate
)

xx <- gs_design_ahr(
  enroll_rate = enrollRates,
  fail_rate = failRates,
  alpha = alpha,
  beta = beta,
  info_frac = NULL,
  analysis_time = analysisTimes,
  ratio = 1,
  binding = FALSE,
  upper = gs_b,
  upar = qnorm(0.975),
  lower = gs_b,
  lpar = qnorm(0.975),
  h1_spending = TRUE,
  test_upper = TRUE,
  test_lower = TRUE,
) |> to_integer()

Events <- xx$analysis$event
N <- xx$analysis$n
enrollRates$rate <- N / enrollRates$duration
```

```
upar <- qnorm(0.975)
lpar <- upar
output <- NULL
for (i in 1:nrow(scenarios)) {
  # Set failure rates
  # Failure rates from 1st scenario for design
  failRates <- define_fail_rate(
    duration = c(scenarios$Delay[i], 100),
    fail_rate = log(2) / scenarios$ControlMedian[i],
    hr = c(scenarios$hr1[i], scenarios$hr2[i]),
```

```
  dropout_rate = dropoutRate
)
# Require events only
yEvents <- gs_power_ahr(
  enroll_rate = enrollRates,
  fail_rate = failRates,
  event = Events,
  analysis_time = NULL,
  upar = upar,
  upper = gs_b,
  lpar = lpar,
  lower = gs_b
)
# Require time only
yTime <- gs_power_ahr(
  enroll_rate = enrollRates,
  fail_rate = failRates,
  event = NULL,
  analysis_time = analysisTimes,
  upar = upar,
  upper = gs_b,
  lpar = lpar,
  lower = gs_b
)
# Require events and time for analysis timing
yBoth <- gs_power_ahr(
  enroll_rate = enrollRates,
  fail_rate = failRates,
  event = Events,
  analysis_time = analysisTimes,
  upar = upar,
  upper = gs_b,
  lpar = lpar,
  lower = gs_b
)
output <- rbind(
  output,
  tibble(
    i = i,
    Scenario = scenarios$Scenario[i],
    "Control Median" = scenarios$ControlMedian[i],
    Delay = scenarios$Delay[i],
    HR1 = scenarios$hr1[i],
    HR2 = scenarios$hr2[i],
    Timing = "Event-based",
    N = N,
    Events = yEvents$analysis$event,
    Time = yEvents$analysis$time,
    AHR = yEvents$analysis$ahr,
    Power = yEvents$bound$probability[1],
    `HR at bound` = yEvents$bound$`~hr at bound`[1]
  ),
  tibble(
```

```
      i = i,
      Scenario = scenarios$Scenario[i],
      "Control Median" = scenarios$ControlMedian[i],
      Delay = scenarios$Delay[i],
      HR1 = scenarios$hr1[i],
      HR2 = scenarios$hr2[i],
      Timing = "Time-based",
      N = N,
      Events = yTime$analysis$event,
      Time = yTime$analysis$time,
      AHR = yTime$analysis$ahr,
      Power = yTime$bound$probability[1],
      `HR at bound` = yTime$bound$`~hr at bound`[1]
    ),
    tibble(
      i = i,
      Scenario = scenarios$Scenario[i],
      "Control Median" = scenarios$ControlMedian[i],
      Delay = scenarios$Delay[i],
      HR1 = scenarios$hr1[i],
      HR2 = scenarios$hr2[i],
      Timing = "Max of time- and event-cutoff",
      N = N,
      Events = yBoth$analysis$event,
      Time = yBoth$analysis$time,
      AHR = yBoth$analysis$ahr,
      Power = yBoth$bound$probability[1],
      `HR at bound` = yBoth$bound$`~hr at bound`[1]
    )
  )
}
```

A sample size of 422 and 312 targeted events was derived based on Scenario 1 assumptions using the method of Zhao, Zhang, and Anderson (2024). This method computes an average hazard ratio at the time of analysis that approximates what is produced by a Cox regression model. We will verify this below using simulated trials. The design has the following additional characteristics:

- An exponential dropout rate of 0.001 in both treatment groups.
- A constant random enrollment rate is planned for 12 months; piecewise constant enrollment is also an option in the software.
- Sample size rounded up to an even number.
- Targeted events rounded up to an integer.
- Planned analysis time at 36 months.
- Spending is based on the information fraction at the planned analysis times using a Lan-DeMets spending function to approximate the O'Brien-Fleming spending function.
- One-sided testing only; no futility bound.
- Sample size rounded to an even number, event counts rounded to the nearest integer for interim analyses and rounded up for the final analysis.

Table 2 summarizes operating characteristics for the 3 scenarios with each of the analysis timing methods. For event-based timing, the second scenario is more poorly powered than for calendar-based timing. For calendar-based timing, the third scenario is more poorly powered than for event-based timing. Thus, requiring both calendar duration and the targeted number of events before performing analysis ensures good power for all 3 scenarios; neither calendar- nor event-based timing achieve this goal. We note a few things about the alternate strategies:

Table 2: Asymptotic approximations of design operating characteristics
Evaluating final analysis timing

Scenario	Control Median	Delay	HR1	HR2	N	Events	Time	AHR	Power	HR at bound
Event-based										
Original plan	12	4	1.0	0.60	422	312.0	36.0	0.692	0.901	0.801
Events accrue faster	10	6	1.2	0.50	422	312.0	30.6	0.771	0.630	0.801
Events accrue slower	16	6	1.0	0.55	422	312.0	47.5	0.661	0.953	0.801
Proportional hazards	12	4	0.7	0.70	422	312.0	35.0	0.700	0.882	0.801
Null hypothesis	12	4	1.0	1.00	422	312.0	30.1	1.000	0.025	0.801
Time-based										
Original plan	12	4	1.0	0.60	422	312.0	36.0	0.692	0.901	0.801
Events accrue faster	10	6	1.2	0.50	422	335.5	36.0	0.748	0.755	0.807
Events accrue slower	16	6	1.0	0.55	422	268.5	36.0	0.682	0.878	0.787
Proportional hazards	12	4	0.7	0.70	422	317.0	36.0	0.700	0.887	0.802
Null hypothesis	12	4	1.0	1.00	422	342.2	36.0	1.000	0.025	0.809
Max of time- and event-cutoff										
Original plan	12	4	1.0	0.60	422	312.0	36.0	0.692	0.901	0.801
Events accrue faster	10	6	1.2	0.50	422	335.5	36.0	0.748	0.755	0.807
Events accrue slower	16	6	1.0	0.55	422	312.0	47.5	0.661	0.953	0.801
Proportional hazards	12	4	0.7	0.70	422	317.0	36.0	0.700	0.887	0.802
Null hypothesis	12	4	1.0	1.00	422	342.2	36.0	1.000	0.025	0.809

- The approximate hazard ratio required for statistical significance is close to 0.8 for all scenarios. Thus, if this is used as one measure of clinical significance, we have not gamed the timing of the final analysis to reach significance with a substantially lesser treatment effect.
- For the scenario with more events accruing early and a longer effect delay, the increased events (380 vs 342) and bigger underlying treatment effect (AHR of 0.78 vs 0.76) achieved by delaying the analysis time resulted in a meaningful power increase (64% to 77%),
- For the scenario with slower event accrual, delaying the analysis until the targeted events accrue compared to analyzing at the targeted time resulted in a larger number of expected events (342 vs. 295), more substantial average hazard reduction (0.715 vs 0.735) and greater power (87% vs 75%).

Since trials represent a very large investment of patients, provider efforts (care and follow-up), and money, the gains from setting timing based on the maximum of event- and calendar-targets has important benefits. While this strategy can result in a delay in study completion, we can still do group sequential analysis to stop early for a larger than a conservatively planned treatment effect used for design.

```
output |>
  select(-i) |>
  group_by(Timing) |>
  gt() |>
  fmt_number(columns = c(Events, Time), decimals = 1) |>
  fmt_number(columns = c(AHR, Power, "HR at bound"), decimals = 3) |>
  tab_header("Table 2: Asymptotic approximations of design operating characteristics",
    subtitle = "Evaluating final analysis timing"
  ) |>
  tab_options(table.font.size = px(11))
```

4.1 Simulation of fixed design

We use simulation to verify the power approximations above as well as whether Type I error is controlled at the targeted one-sided 2.5%. Table 3 shows that with 10^5 simulated trials, the computations from Table 2 provide good approximations. The `SE(Power)` provides the standard error for the simulation power approximation

for each scenario as a measure of accuracy. The average in the HR column is the geometric average of the Cox regression estimate of the hazard ratio.

```
if (run_simulations) {
  sim_summary <- NULL
  tic(paste("Fixed design simulation duration for", nsim, "simulated trials"))
  for (i in seq_along(scenarios$i)) {
    # Get failure rate and event rate assumptions from scenario i = 1
    rates <- define_fail_rate(
      duration = c(scenarios$Delay[i], 100),
      fail_rate = log(2) / scenarios$ControlMedian[i],
      hr = c(scenarios$hr1[i], scenarios$hr2[i]),
      dropout_rate = dropoutRate
    )
    # Run a simulation for power
    sim <- sim_fixed_n(
      n_sim = nsim,
      enroll_rate = xx$enroll_rate,
      sample_size = max(xx$analysis$n),
      target_event = max(xx$analysis$event),
      total_duration = max(xx$analysis$time),
      fail_rate = rates
    )

    # Summarize results
    sim_summary <- rbind(
      sim_summary,
      sim |> select(-names(sim)[1:4]) |> group_by(cut) |>
        summarise(
          Scenario = scenarios$Scenario[i], Simulations = nsim,
          Power = mean(z >= qnorm(0.975)),
          "SE(Power)" = sd(z >= qnorm(0.975)) / sqrt(nsim),
          HR = exp(mean(ln_hr)), "E(Duration)" = mean(duration),
          "SD(Duration)" = sd(duration), "E(Events)" = mean(event),
          "SD(Events)" = sd(event)
        )
    )
  }
  save(sim_summary, file = "FixedDesignSimulation.RData")
  toc()
} else {
  load("FixedDesignSimulation.RData")
}
```

```
#> Fixed design simulation duration for 1e+05 simulated trials: 700.34 sec elapsed
```

```
sim_summary |>
  transmute(
    Scenario = factor(Scenario),
    Cut = factor(cut),
    Power = round(Power, 4),
    "SE(Power)" = round(`SE(Power)`, 4),
    HR = round(HR, 3),
    "E(Duration)" = round(`E(Duration)`, 1),
    "SD(Duration)" = round(`SD(Duration)`, 1),
```

Table 3: Simulation results by scenario and data cutoff method

Scenario	Power	SE(Power)	HR	E(Duration)	SD(Duration)	E(Events)	SD(Events)
Max(min follow-up, event cut)							
Events accrue faster	0.7351	0.0014	0.752	36.0	0.6	335.4	8.2
Events accrue slower	0.9492	0.0007	0.662	47.4	2.9	312.0	0.0
Null hypothesis	0.0252	0.0005	1.000	36.0	0.6	342.1	8.1
Original plan	0.9035	0.0009	0.691	36.8	1.3	315.6	5.2
Proportional hazards	0.8893	0.0010	0.700	36.3	0.9	318.5	6.7
Max(planned duration, event cut)							
Events accrue faster	0.7336	0.0014	0.752	36.0	0.0	335.4	8.2
Events accrue slower	0.9492	0.0007	0.662	47.4	2.9	312.0	0.0
Null hypothesis	0.0252	0.0005	1.000	36.0	0.0	342.1	8.2
Original plan	0.9033	0.0009	0.691	36.8	1.2	315.6	5.2
Proportional hazards	0.8894	0.0010	0.700	36.3	0.8	318.5	6.7
Minimum follow-up							
Events accrue faster	0.7350	0.0014	0.752	36.0	0.6	335.4	8.2
Events accrue slower	0.8740	0.0010	0.682	36.0	0.6	268.5	9.8
Null hypothesis	0.0252	0.0005	1.000	36.0	0.6	342.1	8.1
Original plan	0.8992	0.0010	0.692	36.0	0.6	311.9	9.0
Proportional hazards	0.8880	0.0010	0.700	36.0	0.6	316.9	8.9
Planned duration							
Events accrue faster	0.7335	0.0014	0.752	36.0	0.0	335.4	8.3
Events accrue slower	0.8743	0.0010	0.682	36.0	0.0	268.5	9.8
Null hypothesis	0.0252	0.0005	1.000	36.0	0.0	342.1	8.2
Original plan	0.8989	0.0010	0.692	36.0	0.0	311.9	9.0
Proportional hazards	0.8881	0.0010	0.700	36.0	0.0	316.9	8.9
Targeted events							
Events accrue faster	0.6012	0.0015	0.776	30.5	1.8	312.0	0.0
Events accrue slower	0.9492	0.0007	0.662	47.4	2.9	312.0	0.0
Null hypothesis	0.0255	0.0005	1.000	30.0	1.5	312.0	0.0
Original plan	0.8953	0.0010	0.692	35.9	2.0	312.0	0.0
Proportional hazards	0.8823	0.0010	0.700	34.9	1.9	312.0	0.0

```
    "E(Events)" = round(`E(Events)`, 1),
    "SD(Events)" = round(`SD(Events)`, 1)
  ) |>
  arrange(Cut, Scenario) |>
  gt(groupname_col = "Cut") |>
  fmt_number(columns = c(Power, `SE(Power)`), decimals = 4) |>
  fmt_number(columns = c(HR), decimals = 3) |>
  fmt_number(
    columns = c(`E(Duration)`, `SD(Duration)`, `E(Events)`, `SD(Events)`),
    decimals = 1
  ) |>
  cols_align(align = "center", columns = everything()) |>
  tab_header(title = "Table 3: Simulation results by scenario and data cutoff method") |>
  tab_options(table.font.size = px(11))
```

Simulations were performed with the `simtrial::sim_fixed_n()` function using parallel computing with 32 CPUs. While the user sets up the number of CPU cores used and parallel backend, the parallel computing is otherwise implemented automatically by `simtrial::sim_fixed_n()`.

5 Group sequential design

Now we derive the planned design with the *Original plan* assumptions above. The design has the following additional characteristics. Other than the group sequential assumptions, these are the same as for the fixed design above.

- An exponential dropout rate of 0.001 in both treatment groups.
- A constant random enrollment rate is planned for 12 months; piecewise constant enrollment is also an option in the software.
- Planned analysis times at 20, 28, and 36 months.
- Spending is based on the information fraction at the planned analysis times using a Lan-DeMets spending function to approximate the O'Brien-Fleming spending function.
- One-sided testing only; no futility bound.
- Sample size rounded to an even number, event counts rounded to the nearest integer for interim analyses and rounded up for the final analysis.

```
# Failure rates from 1st scenario for design
failRates <- define_fail_rate(
  duration = c(scenarios$Delay[1], 100),
  fail_rate = log(2) / scenarios$ControlMedian[1],
  hr = c(scenarios$hr1[1], scenarios$hr2[1]),
  dropout_rate = dropoutRate
)
upar_design <- list(sf = gsDesign::sfLDOF, total_spend = alpha)
# Find sample size for 36 month trial under given
# enrollment and sample size assumptions
# Interim at planned information fraction
xx <- gs_design_ahr(
  enroll_rate = enrollRates,
  fail_rate = failRates,
  analysis_time = c(20, 28, 36),
  # analysisTimes,
  info_frac = NULL,
  # planned_IF,
  upper = gs_spending_bound,
  upar = upar_design,
  lpar = rep(-Inf, 3),
  lower = gs_b,
  test_lower = FALSE,
  beta = beta,
  alpha = alpha
) |> to_integer()
```

The planned analyses and bounds are in the table below. In spite of two interim analyses with 0.66 and 0.86 of the final planned observations, the O'Brien-Fleming spending approach preserves enough α for the final analysis to be tested at a nominal 0.0201 level. While the 0.86 information fraction at the final interim may seem high, the intent to do analyses every 8 months starting at month 20 dictates this. At the final analysis, a hazard ratio of approximately 0.79 or smaller is required for a positive finding. Under the alternate hypothesis, there is a substantial probability of crossing an efficacy bound prior to the final analysis as indicated by the cumulative power of 0.35 at the first interim analysis and 0.75 at the second. We see that the expected geometric mean of the observed hazard ratio (AHR) decreases from 0.74 at the first interim to 0.71 at the second, and 0.69 at the final analysis. Thus, the longer the trial continues, the stronger the expected treatment effect is. We will see that continuing the trial long enough for the expected treatment effect to be strong is important in the scenarios that follow.

Table 4: Group sequential design based on original assumptions
Delayed effect for 4 months, HR = 0.6 thereafter

				Cumulative boundary crossing probability	
Bound	Z	Nominal p[1]	~HR at bound[2]	Alternate hypothesis	Null hypothesis
Analysis: 1 Time: 19.9 N: 430 Events: 209 AHR: 0.74 Information fraction: 0.66					
Efficacy	2.53	0.0057	0.7046	0.3454	0.0057
Analysis: 2 Time: 27.9 N: 430 Events: 273 AHR: 0.71 Information fraction: 0.86					
Efficacy	2.20	0.0138	0.7661	0.7453	0.0156
Analysis: 3 Time: 36 N: 430 Events: 318 AHR: 0.69 Information fraction: 1					
Efficacy	2.05	0.0201	0.7944	0.9009	0.0250

[1]One-sided p-value for experimental vs control treatment. Value < 0.5 favors experimental, > 0.5 favors control.
[2]Approximate hazard ratio to cross bound.

```
xx |>
  summary() |>
  as_gt(
    title = "Table 4: Group sequential design based on original assumptions",
    subtitle = "Delayed effect for 4 months, HR = 0.6 thereafter"
  ) |>
  tab_options(table.font.size = px(12))
```

5.1 Same design with weighted logrank

We consider the same sample size and analysis timing but using the modestly weighted logrank (MWLR) test of Magirr and Burman (2019). Weights increase from 1 at time 0 to a maximum of 2 at the combined treatment group median and beyond. We see that power is higher than for the unweighted logrank above. Note that due to the higher weighting of later failures, the information fraction is lower for interim analyses than when using the logrank test above. Thus, the increased power relative to the logrank test above is due both to the weighting and spending changes.

```
gs_arm <- gs_create_arm(xx$enroll_rate, xx$fail_rate, xx$ratio)
mb_design <- gs_power_wlr(
  enroll_rate = xx$enroll_rate,
  fail_rate = xx$fail_rate,
  analysis_time = xx$analysis$time,
  # Magirr-Burman weighting for MWLR
  weight = list(method = "mb", param = list(tau = NULL, w_max = 2)),
  upper = gs_spending_bound,
  upar = upar_design,
  lpar = rep(-Inf, 3),
  lower = gs_b,
  test_lower = FALSE
) |> to_integer()
```

```
mb_design |>
  summary() |>
  as_gt(
    title = "Table 5: Group sequential design based on original assumptions",
    subtitle = "Testing with MWLR using maximum weight of 2"
```

Table 5: Group sequential design based on original assumptions
Testing with MWLR using maximum weight of 2

				Cumulative boundary crossing probability	
Bound	Z	Nominal p[1]	~wHR at bound[2]	Alternate hypothesis	Null hypothesis
Analysis: 1 Time: 19.9 N: 430 Events: 209 AHR: 0.72 Information fraction: 0.49[3]					
Efficacy	2.99	0.0014	0.6615	0.2725	0.0014
Analysis: 2 Time: 27.9 N: 430 Events: 273 AHR: 0.68 Information fraction: 0.78[3]					
Efficacy	2.29	0.0109	0.7577	0.8043	0.0114
Analysis: 3 Time: 36 N: 430 Events: 318 AHR: 0.66 Information fraction: 1[3]					
Efficacy	2.02	0.0215	0.7970	0.9492	0.0250

[1] One-sided p-value for experimental vs control treatment. Value < 0.5 favors experimental, > 0.5 favors control.
[2] Approximate hazard ratio to cross bound.
[3] wAHR is the weighted AHR.

```
  ) |>
  tab_options(table.font.size = px(12))
```

We check that the cumulative α-spending at each analysis is as expected:

```
sfLDOF(alpha = 0.025, t = mb_design$analysis$info_frac0)$spend
```

```
#> [1] 0.001409526 0.011392119 0.025000000
```

6 Accumulation of events and average treatment effect over time

In addition to the different failure rate scenarios, we consider 3 enrollment scenarios to achieve the targeted sample size:

- The original design assumptions.
- A rate 50% higher than the original design, with 4 months less time for the expected sample size to reach the target.
- A rate 25% lower than the original design with 4 months more time for the expected sample size to reach the target.

```
enroll_rate_scenarios <- rbind(
  define_enroll_rate(duration = 12, rate = xx$enroll_rate$rate[1]) |>
    mutate(Enrollment = "Original"),
  define_enroll_rate(duration = 8, rate = 1.5 * xx$enroll_rate$rate[1]) |>
    mutate(Enrollment = "Faster"),
  define_enroll_rate(duration = 16, rate = 0.75 * xx$enroll_rate$rate[1]) |>
    mutate(Enrollment = "Slower")
)
enroll_rate_scenarios |>
  select(-stratum) |>
  gt() |>
  tab_header(title = "Table 6: Enrollment rate scenarios") |>
  fmt_number(columns = "rate", decimals = 1) |>
  cols_label(
    duration = "Target duration (months)", rate = "Enrollment rate per month",
    Enrollment = "Scenario"
```

Table 6: Enrollment rate scenarios

Target duration (months)	Enrollment rate per month	Scenario
12	35.8	Original
8	53.7	Faster
16	26.9	Slower

```
  ) |>
  tab_options(table.font.size = px(12))
```

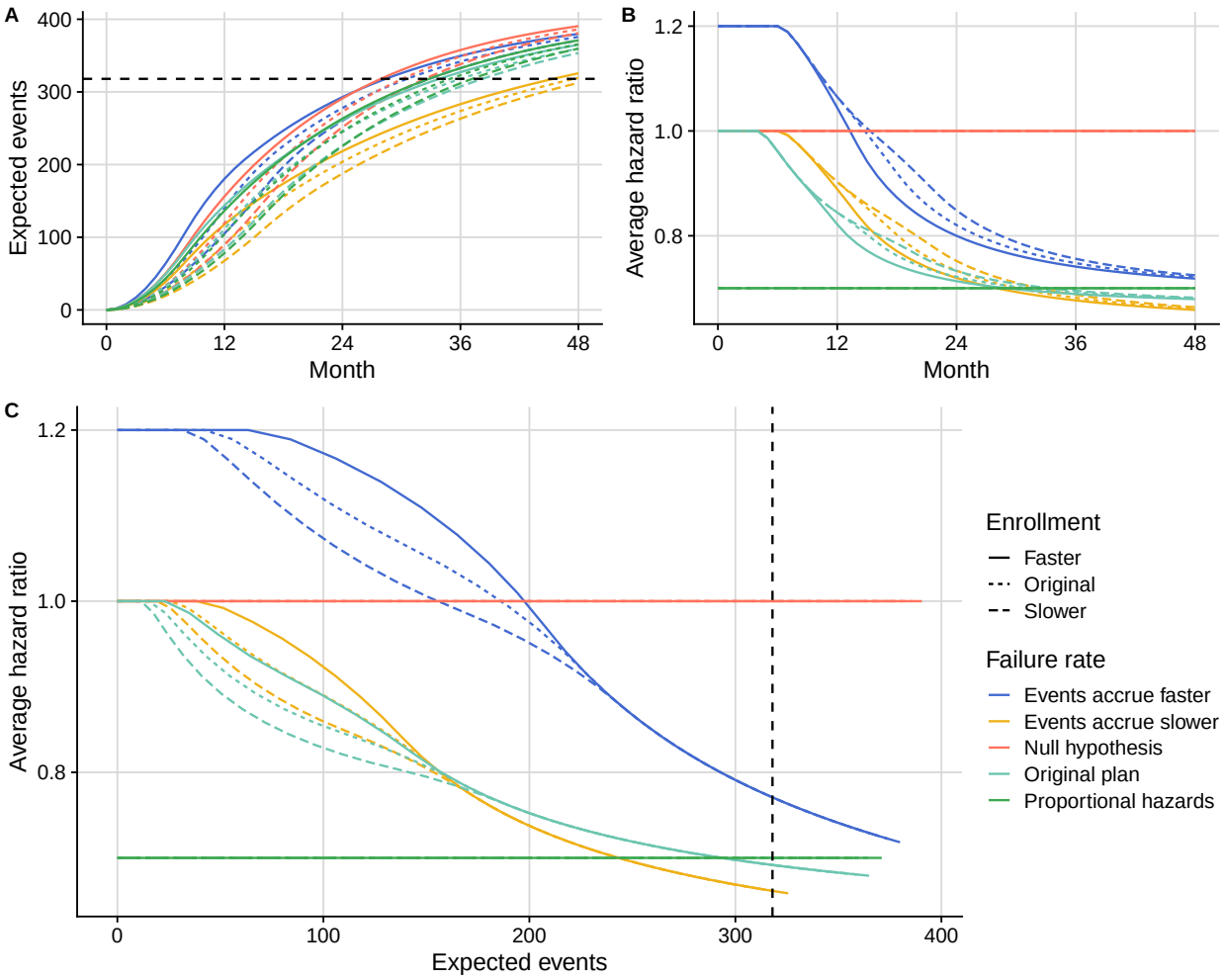
Now we compute expected events and average hazard ratio over time for each enrollment and event rate scenario. We see that for the higher event rate scenario events may accrue closer to 30 months rather than the planned 36 months (Figures A and C). For faster enrollment rates, the delayed effect as evidenced by average hazard ratio (AHR) is exaggerated (Figures B and C). The average hazard ratio appears to have an important decrease (improvement) still between 30 and 36 months (Figure B), suggesting it could be advantageous to have the trial continue for at least 36 months. For lower than planned event rates, it appears important to extend the trial past 36 months to accrue the targeted events.

```
output <- NULL
for (i in 1:nrow(scenarios)) {
  # Set failure rates
  failRates <- define_fail_rate(
    duration = c(scenarios$Delay[i], 100),
    fail_rate = log(2) / scenarios$ControlMedian[i],
    hr = c(scenarios$hr1[i], scenarios$hr2[i]),
    dropout_rate = dropoutRate
  )
  for (j in 1:nrow(enroll_rate_scenarios)) {
    # Find sample size for 30 month trial under given
    # enrollment and sample size assumptions
    yy <- ahr(
      enroll_rate = enroll_rate_scenarios[j, ],
      fail_rate = failRates,
      total_duration = c(0.001, 1:48),
      ratio = 1
    ) |>
      mutate(
        Enrollment = enroll_rate_scenarios$Enrollment[j],
        Scenario = scenarios$Scenario[i]
      )
    output <- rbind(output, yy)
  }
}
```

```
p1 <- output |> ggplot(aes(x = time, y = event, color = Scenario, lty = Enrollment)) +
  geom_line() +
  geom_hline(yintercept = max(xx$analysis$event), linetype = "dashed") +
  scale_x_continuous(breaks = seq(0, 48, 12)) +
  guides(lty = "none", color = "none") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  ylab("Expected events") +
```

```
  xlab("Month")

p2 <- output |> ggplot(aes(x = time, y = ahr, color = Scenario, lty = Enrollment)) +
  geom_line() +
  scale_x_continuous(breaks = seq(0, 48, 12)) +
  guides(lty = "none", color = "none") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  ylab("Average hazard ratio") +
  xlab("Month")

p3 <- output |> ggplot(aes(x = event, y = ahr, color = Scenario, lty = Enrollment)) +
  geom_line() +
  geom_vline(xintercept = max(xx$analysis$event), linetype = "dashed") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  guides(color = guide_legend(title = "Failure rate")) +
  ylab("Average hazard ratio") +
  xlab("Expected events") +
  theme(legend.position = "right")
```

```
top_row <- cowplot::plot_grid(p1, p2, labels = c("A", "B"), label_size = 12)

cowplot::plot_grid(
  top_row, p3,
  labels = c("", "C"), label_size = 12, ncol = 1,
  rel_heights = c(0.4, 0.6)
)
```

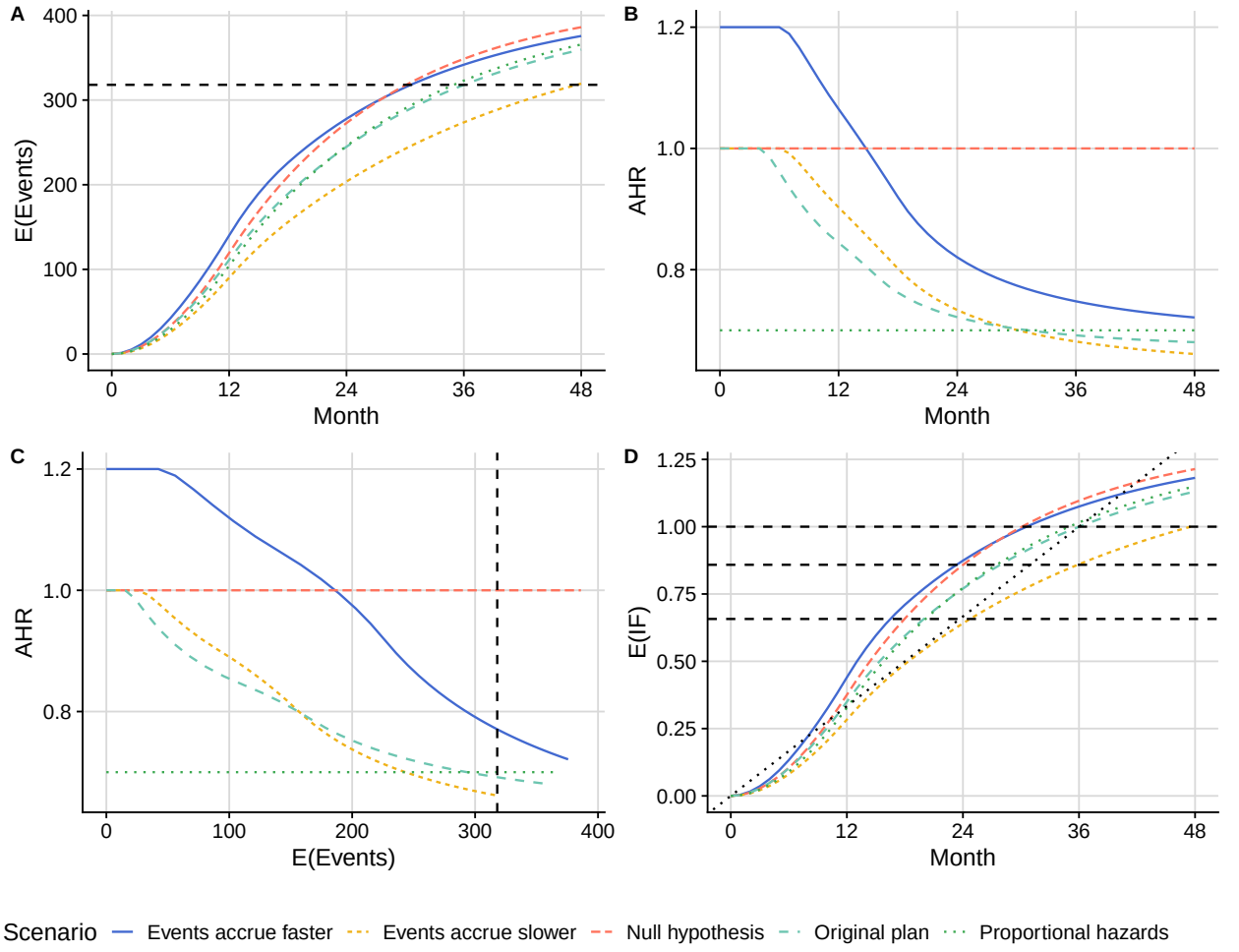


To simplify the above plots without losing the basic interpretation, we consider the different failure rate scenarios only for the original enrollment rate scenario. We add a plot of the expected information fraction for the scenarios above using the planned maximum information fraction from the original design as the denominator.

```
p1 <- output |>
  filter(Enrollment == "Original") |>
  ggplot(aes(x = time, y = event, lty = Scenario, color = Scenario)) +
  geom_line() +
  geom_hline(yintercept = max(xx$analysis$event), linetype = "dashed") +
  scale_x_continuous(breaks = seq(0, 48, 12)) +
  guides(lty = "none", color = "none") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  ylab("E(Events)") +
  xlab("Month")

p2 <- output |>
  filter(Enrollment == "Original") |>
  ggplot(aes(x = time, y = ahr, lty = Scenario, color = Scenario)) +
  geom_line() +
```

```
  scale_x_continuous(breaks = seq(0, 48, 12)) +
  guides(lty = "none", color = "none") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  ylab("AHR") +
  xlab("Month")

p3 <- output |>
  filter(Enrollment == "Original") |>
  ggplot(aes(x = event, y = ahr, lty = Scenario, color = Scenario)) +
  geom_line() +
  geom_vline(xintercept = max(xx$analysis$event), linetype = "dashed") +
  guides(lty = "none", color = "none") +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  guides(lty = "none") +
  ylab("AHR") +
  xlab("E(Events)")

p4 <- output |>
  filter(Enrollment == "Original") |>
  mutate(IF = info0 / max(xx$analysis$info0)) |>
  ggplot(aes(x = time, y = IF, lty = Scenario, color = Scenario)) +
  geom_line() +
  scale_x_continuous(breaks = seq(0, 48, 12)) +
  cowplot::theme_half_open() +
  cowplot::background_grid() +
  ggsci::scale_color_observable() +
  ylab("E(IF)") +
  xlab("Month") +
  geom_hline(yintercept = 1, linetype = "dashed") +
  geom_hline(yintercept = xx$analysis$info_frac0[2], linetype = "dashed") +
  geom_hline(yintercept = xx$analysis$info_frac0[1], linetype = "dashed") +
  geom_abline(intercept = 0, slope = 1 / max(xx$analysis$time), linetype = "dotted") +
  theme(legend.position = "bottom")

legend <- get_legend(p4)
p4a <- p4 + theme(legend.position = "none")
#  guides(lty = "none", color = "none")

# Now put all of these plots in a grid
top_row <- cowplot::plot_grid(
  p1, p2, p3, p4a,
  labels = c("A", "B", "C", "D"),
  label_size = 12,
  ncol = 2,
  rel_heights = c(0.5, 0.5)
)
cowplot::plot_grid(
  top_row, legend,
  labels = c("", ""),
```

```
  label_size = 12,
  ncol = 1,
  rel_heights = c(0.9, 0.1)
)
```



7 Alternate analysis timing and α-spending strategies

Timing of analyses and α-spending are done 3 ways:

1) Event-based timing and spending based on planned events above.
2) Time-based analyses and spending based on planned calendar time and planned spending above.
3) Analyses at maximum of time when targeted events and targeted calendar time are reached, spending as minimum of planned and actual.

Another approach that might be considered is to analyze after, say, every 8 months until the targeted events are reached with the first interim analysis at 20 months; this has not been undertaken here as the 3rd approach above is presumed adequate to demonstrate the principle we are suggesting.

We will focus on examples with the planned enrollment rates as these are adequate for demonstrating the issues we are considering. You can see in the plot above that different enrollment rates can impact time to achieve targeted events which may be particularly challenging for the calendar-based spending approach.

```
# Planned enrollment rates from design
enrollRates <- xx$enroll_rate
# Planned information fraction at interim(s) and final
planned_IF <- xx$analysis$event / max(xx$analysis$event)
# Spending time entered in timing for planned spending
upar_planned_spending <- list(
  sf = gsDesign::sfLDOF, total_spend = alpha, timing = planned_IF
)
# Information fraction (event-based) spending
max_info0 <- max(xx$analysis$info0)
upar_IF_spending <- list(
  sf = gsDesign::sfLDOF,
  total_spend = alpha,
  param = NULL,
  timing = NULL,
  max(max_info = xx$analysis$info0)
)
# Lower spending
lpar <- rep(-Inf, 2)
# Events for event-based analysis timing
Events <- xx$analysis$event
# Planned analysis times
analysisTimes <- xx$analysis$time
```

The following table displays the results.

```
output <- NULL
for (i in 1:nrow(scenarios)) {
  # Set failure rates
  # Failure rates from 1st scenario for design
  failRates <- define_fail_rate(
    duration = c(scenarios$Delay[i], 100),
    fail_rate = log(2) / scenarios$ControlMedian[i],
    hr = c(scenarios$hr1[i], scenarios$hr2[i]),
    dropout_rate = dropoutRate
  )
  # Event-based timing and spending
  yIF <- gs_power_ahr(
    enroll_rate = enrollRates,
    fail_rate = failRates,
    event = Events,
    analysis_time = NULL,
    upper = gs_spending_bound,
    upar = upar_IF_spending,
    lower = gs_b,
    lpar = rep(-Inf, length(Events))
  ) |> to_integer()
  # Interim spending based on planned IF; timing based on calendar times
  yPlanned <- gs_power_ahr(
    enroll_rate = enrollRates,
    fail_rate = failRates,
    event = NULL,
    analysis_time = analysisTimes,
    upper = gs_spending_bound,
    upar = upar_planned_spending,
    lower = gs_b,
    lpar = rep(-Inf, length(analysisTimes))
```

```
  )
  # Interim timing at max of planned time, events
  # Spending is min of planned and actual
  upar_both <- list(
    sf = gsDesign::sfLDOF, total_spend = alpha, param = NULL,
    timing = pmin(planned_IF, Events / max(Events)), max_info = NULL
  )
  yBoth <- gs_power_ahr(
    enroll_rate = enrollRates,
    fail_rate = failRates,
    event = Events,
    analysis_time = analysisTimes,
    upper = gs_spending_bound,
    upar = upar_both,
    lower = gs_b,
    lpar = rep(-Inf, length(Events))
  )

  output <- rbind(
    output,
    # Event-based spending
    tibble(
      i = i, Scenario = scenarios$Scenario[i], "Control Median" = scenarios$ControlMedian[i],
      Delay = scenarios$Delay[i], HR1 = scenarios$hr1[i], HR2 = scenarios$hr2[i],
      Spending = "IF", Analysis = yIF$analysis$analysis,
      N = yIF$analysis$n, Events = yIF$analysis$event, Time = yIF$analysis$time,
      "Nominal p" = pnorm(-yIF$bound$z),
      Power = yIF$bound$probability, AHR = yIF$analysis$ahr,
      `HR at bound` = yIF$bound$`~hr at bound`, IF = yIF$analysis$info_frac0,
      "Spending time" = yIF$analysis$info_frac0
    ),
    # Planned time, event-based spending
    tibble(
      i = i, Scenario = scenarios$Scenario[i], "Control Median" = scenarios$ControlMedian[i],
      Delay = scenarios$Delay[i], HR1 = scenarios$hr1[i], HR2 = scenarios$hr2[i],
      Spending = "Planned", Analysis = yPlanned$analysis$analysis,
      N = yPlanned$analysis$n, Events = yPlanned$analysis$event, Time = yPlanned$analysis$time,
      "Nominal p" = pnorm(-yPlanned$bound$z),
      Power = yPlanned$bound$probability, AHR = yPlanned$analysis$ahr,
      `HR at bound` = yPlanned$bound$`~hr at bound`, IF = yPlanned$analysis$info_frac0,
      "Spending time" = upar_planned_spending$timing
    ),
    # Max of planned and actual
    tibble(
      i = i, Scenario = scenarios$Scenario[i], "Control Median" = scenarios$ControlMedian[i],
      Delay = scenarios$Delay[i], HR1 = scenarios$hr1[i], HR2 = scenarios$hr2[i],
      Spending = "min(IF, Planned)", Analysis = yBoth$analysis$analysis,
      N = yBoth$analysis$n, Events = yBoth$analysis$event, Time = yBoth$analysis$time,
      "Nominal p" = pnorm(-yBoth$bound$z),
      Power = yBoth$bound$probability, AHR = yBoth$analysis$ahr,
      `HR at bound` = yBoth$bound$`~hr at bound`, IF = yBoth$analysis$info_frac0,
      "Spending time" = pmin(yBoth$analysis$info_frac0, planned_IF)
    )
  )
}
```

The critical difference here is demonstrated when events accrue faster than expected due to the treatment benefit emerging later. Analysis timing is the same for both spending methods. Using information fraction

spending uses up 0.0194 of 0.025 α at the interim analysis rather than the planned 0.012. This means that the nominal p-value for the final test is 0.0162 as opposed to 0.0222 if the planned interim spending is used. This also results in a power increase from 63.8% with information-based spending to 68.4% when planned spending is used. Being able to preserve the largest nominal bound for the final analysis is in the general spirit of setting higher bounds at interim analysis if an early stop is to be justified.

```
output |>
  select(-c(i, "Control Median", Delay, HR1, HR2)) |>
  mutate(
    Spending = factor(Spending, levels = c("IF", "Planned", "min(IF, Planned)")),
    Scenario = factor(Scenario)
  ) |>
  arrange(Scenario, Spending, Analysis) |>
  gt(groupname_col = "Scenario") |>
  fmt_number(columns = c(N, Events, Time), decimals = 1) |>
  fmt_number(
    columns = c(Power, AHR, `HR at bound`, IF, `Spending time`, `Nominal p`),
    decimals = 3
  ) |>
  cols_align(align = "center", columns = everything()) |>
  tab_header(title = "Table 7: Design characteristics by spending strategy across scenarios") |>
  tab_options(table.font.size = px(10), latex.use_longtable = TRUE)
```

Table 7: Design characteristics by spending strategy across scenarios

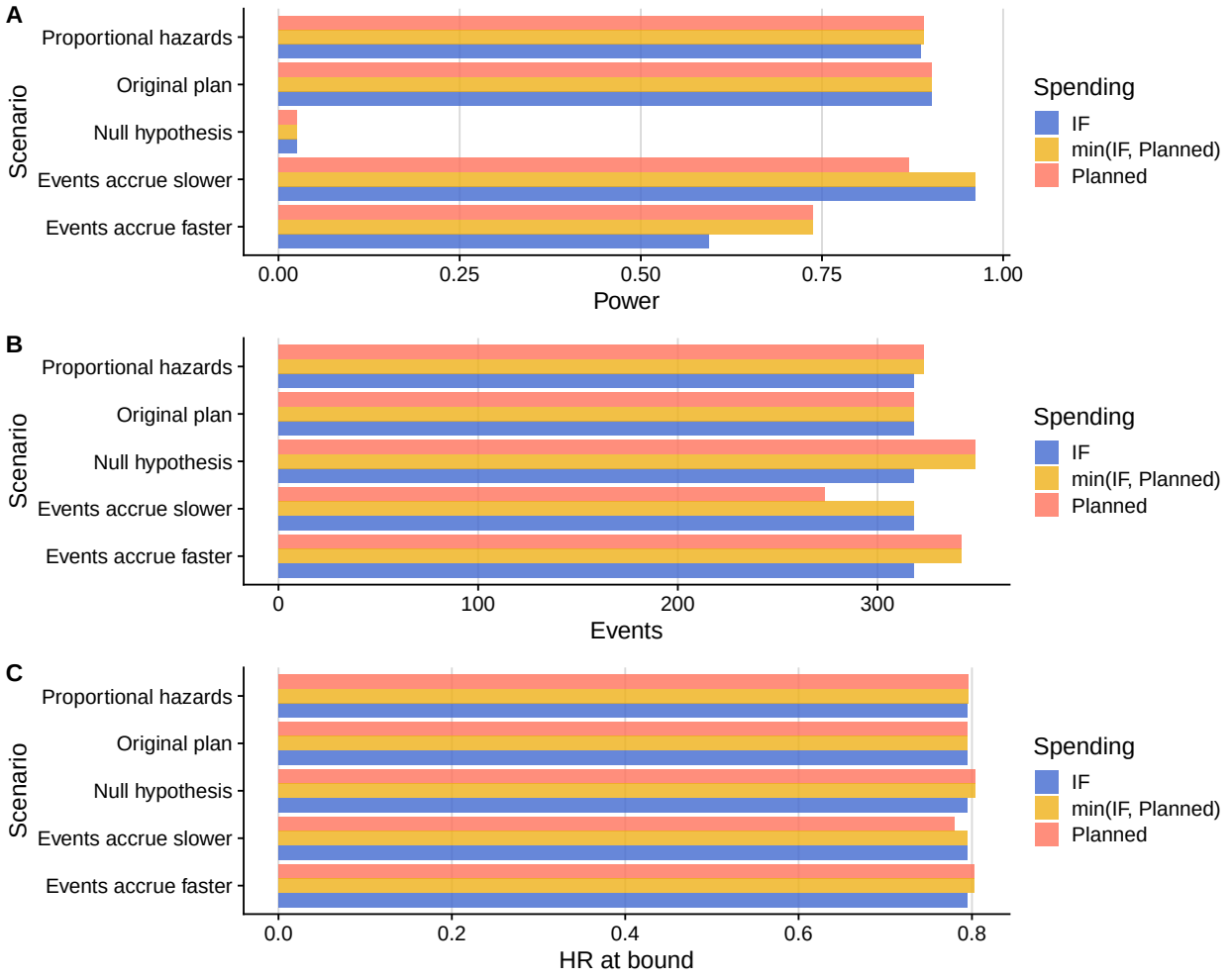
Spending	Analysis	N	Events	Time	Nominal p	Power	AHR	HR at bound	IF	Spending time
Events accrue faster										
IF	1	430.0	209.0	16.5	0.006	0.014	0.958	0.705	0.657	0.657
IF	2	430.0	273.0	23.3	0.014	0.253	0.828	0.766	0.858	0.858
IF	3	430.0	318.0	30.6	0.020	0.593	0.771	0.794	1.000	1.000
Planned	1	430.0	244.7	19.9	0.006	0.070	0.877	0.724	0.716	0.657
Planned	2	430.0	303.1	27.9	0.014	0.448	0.787	0.777	0.887	0.858
Planned	3	430.0	341.9	36.0	0.021	0.738	0.748	0.803	1.000	1.000
min(IF, Planned)	1	430.0	244.7	19.9	0.006	0.070	0.877	0.724	0.716	0.657
min(IF, Planned)	2	430.0	303.1	27.9	0.014	0.448	0.787	0.777	0.887	0.858
min(IF, Planned)	3	430.0	341.9	36.0	0.021	0.738	0.748	0.803	1.000	1.000
Events accrue slower										
IF	1	430.0	209.0	24.7	0.006	0.407	0.728	0.705	0.657	0.657
IF	2	430.0	273.0	35.9	0.014	0.837	0.682	0.766	0.858	0.858
IF	3	430.0	318.0	47.6	0.020	0.961	0.661	0.794	1.000	1.000
Planned	1	430.0	172.5	19.9	0.006	0.204	0.773	0.680	0.630	0.657
Planned	2	430.0	229.6	27.9	0.014	0.647	0.710	0.747	0.839	0.858
Planned	3	430.0	273.7	36.0	0.020	0.870	0.681	0.779	1.000	1.000
min(IF, Planned)	1	430.0	209.0	24.7	0.006	0.407	0.728	0.705	0.657	0.657
min(IF, Planned)	2	430.0	273.0	35.9	0.014	0.837	0.682	0.766	0.858	0.858
min(IF, Planned)	3	430.0	318.0	47.6	0.020	0.961	0.661	0.794	1.000	1.000
Null hypothesis										
IF	1	430.0	209.0	18.0	0.006	0.006	1.000	0.705	0.657	0.657
IF	2	430.0	273.0	24.0	0.014	0.016	1.000	0.766	0.858	0.858
IF	3	430.0	318.0	30.1	0.020	0.025	1.000	0.794	1.000	1.000
Planned	1	430.0	232.6	19.9	0.006	0.006	1.000	0.718	0.667	0.657
Planned	2	430.0	303.2	27.9	0.014	0.016	1.000	0.777	0.869	0.858
Planned	3	430.0	348.7	36.0	0.020	0.025	1.000	0.803	1.000	1.000
min(IF, Planned)	1	430.0	232.6	19.9	0.006	0.006	1.000	0.718	0.667	0.657
min(IF, Planned)	2	430.0	303.2	27.9	0.014	0.016	1.000	0.777	0.869	0.858
min(IF, Planned)	3	430.0	348.7	36.0	0.020	0.025	1.000	0.803	1.000	1.000
Original plan										
IF	1	430.0	209.0	19.9	0.006	0.345	0.745	0.705	0.657	0.657
IF	2	430.0	273.0	27.9	0.014	0.745	0.708	0.766	0.858	0.858
IF	3	430.0	318.0	36.0	0.020	0.901	0.692	0.794	1.000	1.000
Planned	1	430.0	209.0	19.9	0.006	0.345	0.745	0.705	0.657	0.657

Planned	2	430.0	273.0	27.9	0.014	0.745	0.708	0.766	0.858	0.858
Planned	3	430.0	318.0	36.0	0.020	0.901	0.692	0.794	1.000	1.000
min(IF, Planned)	1	430.0	209.0	19.9	0.006	0.345	0.745	0.705	0.657	0.657
min(IF, Planned)	2	430.0	273.0	27.9	0.014	0.745	0.708	0.766	0.858	0.858
min(IF, Planned)	3	430.0	318.0	36.0	0.020	0.901	0.692	0.794	1.000	1.000
Proportional hazards										
IF	1	430.0	209.0	20.1	0.006	0.519	0.700	0.705	0.657	0.657
IF	2	430.0	273.0	27.5	0.014	0.782	0.700	0.766	0.858	0.858
IF	3	430.0	318.0	35.0	0.020	0.886	0.700	0.794	1.000	1.000
Planned	1	430.0	207.2	19.9	0.006	0.514	0.700	0.704	0.641	0.657
Planned	2	430.0	275.7	27.9	0.014	0.786	0.700	0.767	0.853	0.858
Planned	3	430.0	323.1	36.0	0.020	0.890	0.700	0.796	1.000	1.000
min(IF, Planned)	1	430.0	209.0	20.1	0.006	0.519	0.700	0.705	0.647	0.647
min(IF, Planned)	2	430.0	275.7	27.9	0.014	0.786	0.700	0.767	0.853	0.853
min(IF, Planned)	3	430.0	323.1	36.0	0.020	0.890	0.700	0.796	1.000	1.000

Now we summarize design characteristics by spending strategy across scenarios. We see that using the minimum of planned and actual spending retains the highest power across scenarios (A). If events accrue more slowly than planned, the minimum of planned and actual spending adapts to longer duration (B), more events (C) and a stronger effect size (AHR) than using the planned calendar timing and spending. Using event-based (IF) spending results in lower power if events accrue faster than planned due to a lesser effect size (higher AHR) and fewer events than timing analyses at the maximum of planned and actual time. Under proportional hazards and the original delayed effect scenario, all strategies have similar power. The final analysis nominal p-value is highest for the minimum of planned and actual spending, regardless of scenario (E).

```
# Power plot
p1 <- output |>
  filter(Analysis == 3) |>
  ggplot(aes(x = Scenario, y = Power, fill = Spending)) +
  geom_bar(stat = "identity", position = "dodge") +
  cowplot::theme_half_open() +
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  ylab("Power") +
  xlab("Scenario")

# Events plot
p2 <- output |>
  filter(Analysis == 3) |>
  ggplot(aes(x = Scenario, y = Events, fill = Spending)) +
  geom_bar(stat = "identity", position = "dodge") +
  cowplot::theme_half_open() +
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  ylab("Events") +
  xlab("Scenario")

# Z-score plot
p3 <- output |>
  filter(Analysis == 3) |>
  ggplot(aes(x = Scenario, y = `HR at bound`, fill = Spending)) +
  geom_bar(stat = "identity", position = "dodge") +
  cowplot::theme_half_open() +
```

```
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  ylab("HR at bound") +
  xlab("Scenario")

# Display plots with increased height
cowplot::plot_grid(
  p1, p2, p3,
  ncol = 1, labels = c("A", "B", "C"), rel_heights = c(1, 1, 1)
)
```



8 Simulation to verify operating characteristics

We focus on the case where events accrue faster than planned to ensure that adapting spending is not impacting Type I error as well as to confirm power with the different strategies. We write a function to simulate a trial, including cutting off for analyses with different strategies.

8.1 Simulate a single trial and perform tests

```
# Convert failure rates to simulation format
# Scenario 1
# rates <- to_sim_pw_surv(xx$fail_rate)
# Null hypothesis
rates <- to_sim_pw_surv(xx$fail_rate |> mutate(hr = 1))

# Simulate a trial
simulate_a_trial <- function(
    n, enroll_rate, fail_rate, dropout_rate, target_events, target_time,
    block = c(
      "control", "control",
      "experimental", "experimental"
    )) {
  cut_date_events <- rep(0, length(target_events))
  cut_date_both <- rep(0, length(target_events))
  output <- NULL
  dat <- sim_pw_surv(
    n = n, stratum = data.frame(stratum = "All", p = 1),
    enroll_rate = enroll_rate, fail_rate = fail_rate,
    dropout_rate = dropout_rate, block = block
  )

  for (k in seq_along(target_events)) {
    # Time to targeted event count
    cut_date_events[k] <- get_cut_date_by_event(dat, target_events[k])
    # Time to max of targeted event count, targeted time
    cut_date_both[k] <- max(cut_date_events[k], target_time[k])
  }
  # get unique analysis time cutoffs
  cut_times <- unique(c(cut_date_events, target_time))

  # Do analyses at targeted times
  for (k in seq_along(cut_times)) {
    # Cut data
    dat_cut <- cut_data_by_date(dat, cut_times[k])
    # Get counting process dataset to be used for both tests of interest
    counting_data <- dat_cut |> counting_process(arm = "experimental")
    # Sum total events across times at which events are observed
    Events <- sum(dat_cut$event)
    # sum(counting_data$event_total) #
    # Compute logrank test
    test <- counting_data |>
      wlr(weight = fh(rho = 0, gamma = 0), return_variance = TRUE) |>
      as.data.frame() |>
      mutate(test = "lr", Time = cut_times[k], Events = Events) |>
      select(-c(method, parameter, estimate, se))
    # Compute Magirr-Burman test
    test_mb <- counting_data |>
      wlr(weight = mb(delay = Inf, w_max = 2)) |>
      # info conversion is a GUESS to match planned design; needs checking
      as.data.frame() |>
      mutate(test = "mwlr", Time = cut_times[k], Events = Events) |>
```

```
    select(-c(method, parameter, estimate, se))
  # Add a record for each of the above tests
  output <- rbind(output, test, test_mb)
}
# Accumulate analyses for each cutting method
output <-
  bind_rows(
    tibble(Cut = "Events", Time = cut_date_events),
    tibble(Cut = "Time", Time = target_time),
    tibble(Cut = "Both", Time = cut_date_both)
  ) |> left_join(output, by = "Time", relationship = "many-to-many")
# Cutoff at max of planned time, events, spending based on min of planned, actual events
lrboth <- output |> filter(Cut == "Both" & test == "lr")
lrboth$Analysis <- seq_along(lrboth$Events)
lrboth$sTime <- pmin(
  xx$analysis$event / max(xx$analysis$event),
  Events / max(xx$analysis$event)
)
lrboth$sTime[3] <- 1 # Full spend at final analysis
lrboth$spend <- diff(c(0, sfLDOF(alpha = alpha, t = lrboth$sTime)$spend))
lrboth$Bound <- gsBound1(
  theta = 0, I = lrboth$Events, a = rep(-20, 3),
  probhi = lrboth$spend, tol = 1e-06, r = 18
)$b
lrboth$IF <- lrboth$Events / max(xx$analysis$event)
lrboth$"Reject H0" <- ifelse(lrboth$z >= lrboth$Bound, TRUE, FALSE)
lrboth$Spending <- "min(planned, actual)"

# Cutoff at max of planned time, events, spending based on events
lrbothe <- lrboth
lrbothe$Analysis <- seq_along(lrbothe$Events)
lrbothe$sTime <- lrbothe$Events / max(xx$analysis$event)
# Final total spending time should be at least 1
lrbothe$sTime[3] <- max(1, lrbothe$sTime[3])
lrbothe$spend <- diff(c(0, sfLDOF(alpha = alpha, t = lrbothe$sTime)$spend))
# May be instances where final event count is reached by IA 2
# when planned time of IA2 is reached
if (lrbothe$sTime[2] < 1) {
  lrbothe$Bound <- gsBound1(
    theta = 0, I = lrbothe$Events, a = rep(-20, 3),
    probhi = lrbothe$spend, tol = 1e-06, r = 18
  )$b
} else {
  lrbothe$Bound <- c(
    gsBound1(
      theta = 0, I = lrbothe$Events[1:2], a = rep(-20, 2),
      probhi = lrbothe$spend[1:2], tol = 1e-06, r = 18
    )$b,
    Inf
  )
}
lrbothe$IF <- lrbothe$Events / max(xx$analysis$event)
lrbothe$"Reject H0" <- ifelse(lrbothe$z >= lrbothe$Bound, TRUE, FALSE)
```

```r
lrbothe$Spending <- "Events"

# mwlr with min(Planned, Actual) spending
mwlrboth <- output |> filter(Cut == "Both" & test == "mwlr")
mwlrboth$Analysis <- seq_along(mwlrboth$Events)
mwlrboth$sTime <- pmin(
  mwlrboth$info0 / max(mb_design$analysis$info0),
  mb_design$analysis$info_frac0
)
mwlrboth$sTime[3] <- max(1, mwlrboth$sTime[3]) # Full spend at final analysis
mwlrboth$spend <- diff(c(0, sfLDOF(alpha = alpha, t = mwlrboth$sTime)$spend))
mwlrboth$Bound <- gsBound1(
  theta = 0, I = mwlrboth$info0, a = rep(-20, 3),
  probhi = mwlrboth$spend, tol = 1e-06, r = 18
)$b
mwlrboth$IF <- mwlrboth$info0 / max(mb_design$analysis$info0)
mwlrboth$"Reject H0" <- ifelse(mwlrboth$z >= mwlrboth$Bound, TRUE, FALSE)
mwlrboth$Spending <- "min(planned, actual)"

# Cut at max of planned time, events, spending based on info0, mwlr
mwlrbothe <- mwlrboth
mwlrbothe$Analysis <- seq_along(mwlrbothe$info0)
mwlrbothe$sTime <- mwlrbothe$info0 / max(mb_design$analysis$info0)
mwlrbothe$sTime[3] <- max(1, mwlrbothe$sTime[3]) # Full spend at final analysis
mwlrbothe$spend <- diff(c(0, sfLDOF(alpha = alpha, t = mwlrbothe$sTime)$spend))
# May be instances where final info0 is reached by IA 2
# when planned time of IA2 is reached
if (mwlrbothe$sTime[2] < 1) {
  mwlrbothe$Bound <- gsBound1(
    theta = 0, I = mwlrbothe$info0, a = rep(-20, 3),
    probhi = mwlrbothe$spend, tol = 1e-06, r = 18
  )$b
} else {
  mwlrbothe$Bound <- c(
    gsBound1(
      theta = 0, I = mwlrbothe$info0[1:2], a = rep(-20, 2),
      probhi = mwlrbothe$spend[1:2], tol = 1e-06, r = 18
    )$b,
    Inf
  )
}
mwlrbothe$IF <- mwlrbothe$info0 / max(mb_design$analysis$info0)
mwlrbothe$"Reject H0" <- ifelse(mwlrbothe$z >= mwlrbothe$Bound, TRUE, FALSE)
mwlrbothe$Spending <- "IF"

# Cut at targeted events, lr
lrevents <- output |> filter(Cut == "Events" & test == "lr")
lrevents$Analysis <- seq_along(lrevents$Events)
lrevents$sTime <- xx$analysis$info_frac0
lrevents$spend <- diff(c(0, xx$bound$probability0))
lrevents$Bound <- xx$bound$z
lrevents$IF <- xx$analysis$info_frac0
lrevents$"Reject H0" <- ifelse(lrevents$z >= lrevents$Bound, TRUE, FALSE)
```

```
  lrevents$Spending <- "Events"

  # Cut at targeted events, mwlr
  mwlrevents <- output |> filter(Cut == "Events" & test == "mwlr")
  mwlrevents$Analysis <- seq_along(mwlrevents$Events)
  mwlrevents$sTime <- mwlrevents$info0 / max(mb_design$analysis$info0)
  mwlrevents$sTime[3] <- max(1, mwlrevents$sTime[3])
  mwlrevents$spend <- diff(c(0, sfLDOF(alpha = alpha, t = mwlrevents$sTime)$spend))
  mwlrevents$Bound <- gsBound1(
    theta = 0, I = mwlrevents$info0, a = rep(-20, 3),
    probhi = mwlrevents$spend, tol = 1e-06, r = 18
  )$b
  mwlrevents$IF <- mwlrevents$info0 / max(mb_design$analysis$info0)
  mwlrevents$"Reject H0" <- ifelse(mwlrevents$z >= mwlrevents$Bound, TRUE, FALSE)
  mwlrevents$Spending <- "IF"

  # Cut at targeted time, lr
  lrtime <- output |> filter(Cut == "Time" & test == "lr")
  lrtime$Analysis <- seq_along(lrtime$Events)
  lrtime$sTime <- xx$analysis$time / max(xx$analysis$time)
  lrtime$spend <- diff(c(0, sfLDOF(alpha = alpha, t = lrtime$sTime)$spend))
  lrtime$Bound <- gsBound1(
    theta = 0, I = lrtime$Events, a = rep(-20, 3),
    probhi = lrtime$spend, tol = 1e-06, r = 18
  )$b
  lrtime$IF <- lrtime$info0 / max(xx$analysis$info0)
  lrtime$"Reject H0" <- ifelse(lrtime$z >= lrtime$Bound, TRUE, FALSE)
  lrtime$Spending <- "Time"

  # Cut at targeted time, mwlr
  mwlrtime <- output |> filter(Cut == "Time" & test == "mwlr")
  mwlrtime$Analysis <- seq_along(mwlrtime$Events)
  mwlrtime$sTime <- lrtime$sTime
  mwlrtime$spend <- lrtime$spend
  mwlrtime$Bound <- gsBound1(
    theta = 0, I = mwlrtime$info0, a = rep(-20, 3),
    probhi = mwlrtime$spend, tol = 1e-06, r = 18
  )$b
  mwlrtime$IF <- mwlrtime$info0 / max(mb_design$analysis$info0)
  mwlrtime$"Reject H0" <- ifelse(mwlrtime$z >= mwlrtime$Bound, TRUE, FALSE)
  mwlrtime$Spending <- "Time"

  # Combine all results
  bind_rows(
    lrevents, mwlrevents, lrtime, mwlrtime,
    lrboth, mwlrboth, lrbothe, mwlrbothe
  )
}
```

For each trial simulated there are $2 \times 3 \times 4 = 24$ analyses performed, one for each of the 4 cutting-spending strategies (time, events, max of time and event with either event-based or minimum of planned and actual information fraction spending), 2 tests (logrank and Magirr-Burman) as well as 3 analyses (2 interims and final).

```
output <- simulate_a_trial(
  n = max(xx$analysis$n),
  enroll_rate = xx$enroll_rate,
  fail_rate = rates$fail_rate,
  dropout_rate = rates$dropout_rate,
  target_events = xx$analysis$event,
  target_time = xx$analysis$time
)
output |>
  transmute(
    Cut = factor(Cut, levels = c("Events", "Time", "Both")),
    Spending = factor(Spending, levels = c("Events", "Time", "IF", "min(planned, actual)")),
    test = factor(test, levels = c("lr", "mwlr")),
    Analysis = Analysis,
    Time = Time,
    Events = Events,
    IF = IF,
    sTime = sTime,
    spend = spend,
    Bound = Bound,
    z = z,
    "Reject H0" = factor(`Reject H0`),
    info0 = info0,
    info = info
  ) |>
  arrange(test, Cut, Spending, Analysis) |>
  gt(groupname_col = "test") |>
  fmt_number(columns = c(Time, Events, info0, info), decimals = 1) |>
  fmt_number(columns = c(IF, sTime, Bound, z), decimals = 2) |>
  fmt_number(columns = c(spend), decimals = 3) |>
  cols_align(align = "center", columns = everything()) |>
  tab_header(title = "Table 8: Example output from a single trial simulation") |>
  tab_options(table.font.size = px(9), latex.use_longtable = TRUE)
```

Table 8: Example output from a single trial simulation

Cut	Spending	Analysis	Time	Events	IF	sTime	spend	Bound	z	Reject H0	info0	info
lr												
Events	Events	1	18.9	209.0	0.66	0.66	0.006	2.53	-0.46	FALSE	52.2	52.2
Events	Events	2	26.5	273.0	0.86	0.86	0.010	2.20	-0.58	FALSE	68.2	68.2
Events	Events	3	32.8	318.0	1.00	1.00	0.009	2.05	-0.44	FALSE	79.5	79.5
Time	Time	1	19.9	217.0	0.68	0.55	0.003	2.79	-0.58	FALSE	54.2	54.2
Time	Time	2	27.9	286.0	0.90	0.77	0.008	2.32	-0.56	FALSE	71.5	71.5
Time	Time	3	36.0	340.0	1.07	1.00	0.014	2.00	-1.01	FALSE	84.8	84.7
Both	Events	1	19.9	217.0	0.68	0.68	0.007	2.48	-0.58	FALSE	54.2	54.2
Both	Events	2	27.9	286.0	0.90	0.90	0.011	2.14	-0.56	FALSE	71.5	71.5
Both	Events	3	36.0	340.0	1.07	1.07	0.007	2.11	-1.01	FALSE	84.8	84.7
Both	min(planned, actual)	1	19.9	217.0	0.68	0.66	0.006	2.53	-0.58	FALSE	54.2	54.2
Both	min(planned, actual)	2	27.9	286.0	0.90	0.86	0.010	2.20	-0.56	FALSE	71.5	71.5
Both	min(planned, actual)	3	36.0	340.0	1.07	1.00	0.009	2.06	-1.01	FALSE	84.8	84.7
mwlr												
Events	IF	1	18.9	209.0	0.49	0.49	0.001	2.99	-0.76	FALSE	105.9	105.6
Events	IF	2	26.5	273.0	0.77	0.77	0.009	2.32	-0.85	FALSE	165.5	165.2
Events	IF	3	32.8	318.0	0.98	1.00	0.014	2.01	-0.63	FALSE	210.5	210.5
Time	Time	1	19.9	217.0	0.52	0.55	0.003	2.79	-0.88	FALSE	112.0	111.7
Time	Time	2	27.9	286.0	0.83	0.77	0.008	2.34	-0.81	FALSE	178.5	178.3
Time	Time	3	36.0	340.0	1.07	1.00	0.014	2.02	-1.32	FALSE	231.5	231.0
Both	IF	1	19.9	217.0	0.52	0.52	0.002	2.90	-0.88	FALSE	112.0	111.7
Both	IF	2	27.9	286.0	0.83	0.83	0.012	2.22	-0.81	FALSE	178.5	178.3
Both	IF	3	36.0	340.0	1.07	1.07	0.011	2.06	-1.32	FALSE	231.5	231.0
Both	min(planned, actual)	1	19.9	217.0	0.52	0.49	0.001	2.99	-0.88	FALSE	112.0	111.7
Both	min(planned, actual)	2	27.9	286.0	0.83	0.78	0.010	2.29	-0.81	FALSE	178.5	178.3
Both	min(planned, actual)	3	36.0	340.0	1.07	1.00	0.014	2.03	-1.32	FALSE	231.5	231.0

8.2 Simulate multiple trials in parallel

Now we simulate all trials with a parallel computing strategy. We evaluate the different cutting strategies and compare spending-time-based and information-fraction-based testing for both the logrank and Magirr-Burman tests evaluated above.

Set up parallel computing.

```
# Set a seed for reproducibility and use `.options.future = list(seed = TRUE)`
# for parallel-safe RNG. Use foreach + %dofuture% to perform parallel computation.
set.seed(42)
```

```
if (run_simulations) {
  # Start timing
  tic("Total simulation time")

  # Initialize result
  # result <- data.frame() # For sequential
  result <- list(1, 2, 3, 4, 5) # For parallel
  # Run simulation for all scenarios
  for (scen in 1:5) {
    # Run for first 5 scenarios
    failRates <- define_fail_rate(
      duration = c(scenarios$Delay[scen], 100),
      fail_rate = log(2) / scenarios$ControlMedian[scen],
      hr = c(scenarios$hr1[scen], scenarios$hr2[scen]),
      dropout_rate = dropoutRate
    )
    rates <- to_sim_pw_surv(failRates)

    # Run simulations in parallel
    result[[scen]] <- foreach(
      i = 1:nsim,
      .combine = "rbind",
      .options.future = list(seed = TRUE)
    ) %dofuture% {
      # for (i in 1:nsim) {
      trial_result <- simulate_a_trial(
        n = max(xx$analysis$n),
        enroll_rate = xx$enroll_rate,
        fail_rate = rates$fail_rate,
        dropout_rate = rates$dropout_rate,
        target_events = xx$analysis$event,
        target_time = xx$analysis$time,
        block = c("control", "control", "experimental", "experimental")
      )
      trial_result$Sim <- i
      trial_result$Scenario <- scenarios$Scenario[scen]
      # Within each test, Cut, Spending combination, compute a variable 'FirstReject'
      # which is TRUE for the first analysis that 'Reject H0' is TRUE and otherwise FALSE
      trial_result |>
        dplyr::group_by(Sim, test, Cut, Spending) |>
        dplyr::mutate(
          FirstReject = ifelse(`Reject H0` & !duplicated(`Reject H0`), TRUE, FALSE)
        ) |>
        dplyr::ungroup()
```

```
    }
    save(result, file = "DelayedEffectSimulation.RData")
    toc()
  }
} else {
  load("DelayedEffectSimulation.RData")
}
```

```
#> Total simulation time: 741.97 sec elapsed
```

We wish to mimic asymptotic approximations with appropriate simulations for each scenario and analysis for: Spending, Events, Time, Power, AHR (later!), IF, and Spending time.

```
# Basic summary
result_summary <- NULL
for (i in 1:5) {
  result_summary <- rbind(
    result_summary,
    result[[i]] |>
      group_by(test, Cut, Spending, Analysis) |>
      summarize(
        "E(Spend)" = mean(spend),
        "E(Events)" = mean(Events),
        "E(Time)" = mean(Time),
        Power = mean(FirstReject),
        "E(IF)" = mean(IF),
        "E(sTime)" = mean(sTime)
      ) |>
      ungroup() |>
      group_by(test, Cut, Spending) |>
      arrange(Analysis, .by_group = TRUE) |>
      mutate(Power = cumsum(Power), Scenario = scenarios$Scenario[i])
  )
}

# Display results
result_summary |>
  transmute(
    Scenario = factor(Scenario),
    Test = factor(test),
    Cut = factor(Cut),
    Spending = factor(Spending),
    Analysis = factor(Analysis),
    "E(Spend)" = `E(Spend)`,
    "E(Events)" = `E(Events)`,
    "E(Time)" = `E(Time)`,
    Power = Power,
    "E(IF)" = `E(IF)`,
    "E(sTime)" = `E(sTime)`
  ) |>
  arrange(Scenario, Test, Cut, Spending, Analysis) |>
  gt(groupname_col = "Scenario") |>
  fmt_number(columns = c(`E(Time)`, `E(Events)`), decimals = 1) |>
  fmt_number(columns = c(`E(Spend)`, Power), decimals = 4) |>
  fmt_number(columns = c(`E(IF)`), decimals = 3) |>
```

```
fmt_number(columns = c(`E(sTime)`), decimals = 2) |>
cols_align(align = "center", columns = everything()) |>
tab_header(
  title = paste(
    "Table 9: Simulation results summary (based on",
    nsim, "simulations per scenario)"
  )
) |>
tab_options(table.font.size = px(9), latex.use_longtable = TRUE)
```

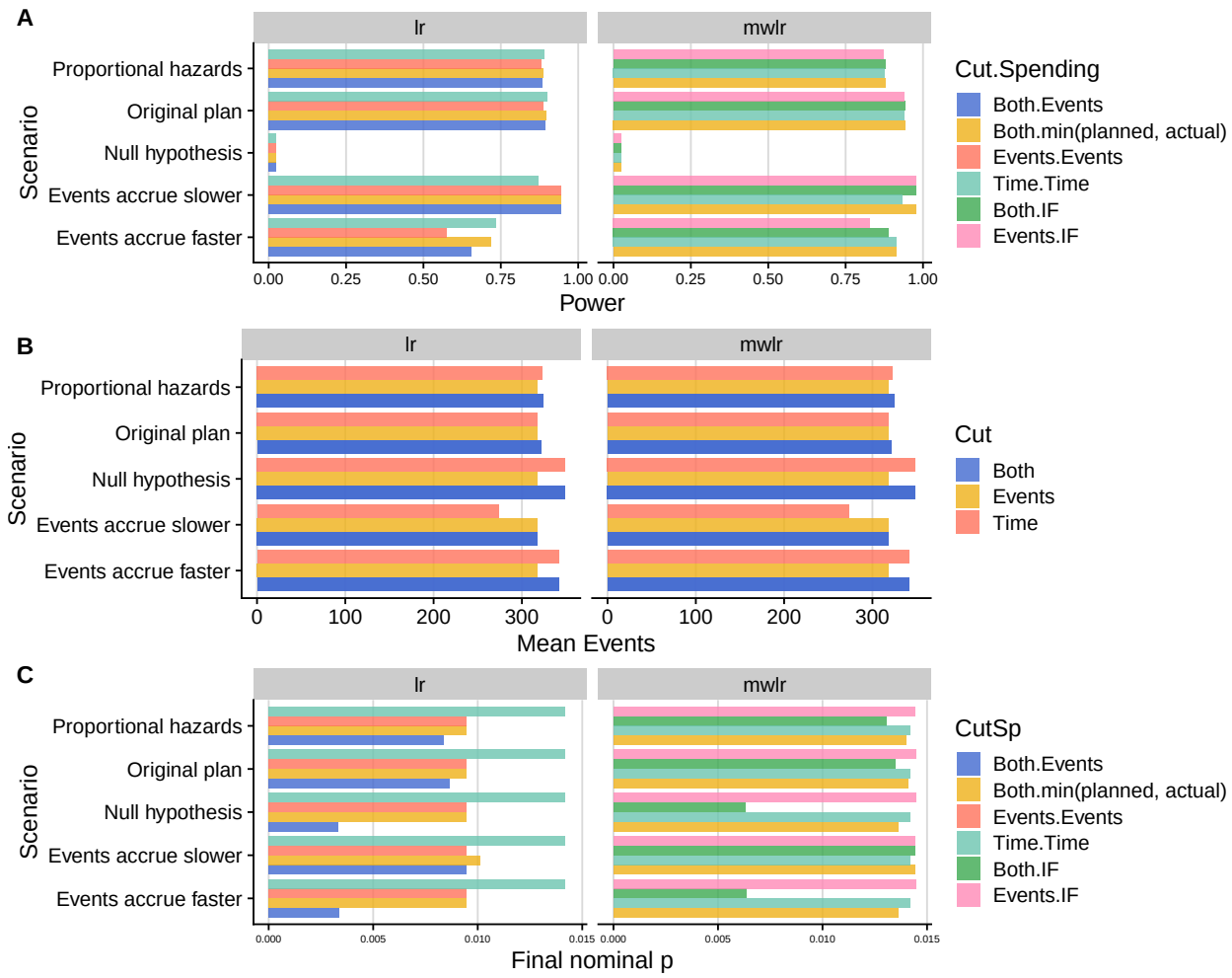
Table 9: Simulation results summary (based on 1e+05 simulations per scenario)

test	Test	Cut	Spending	Analysis	E(Spend)	E(Events)	E(Time)	Power	E(IF)	E(sTime)
Events accrue faster										
lr	lr	Both	Events	1	0.0106	244.5	19.9	0.0880	0.769	0.77
lr	lr	Both	Events	2	0.0110	303.0	27.9	0.4681	0.953	0.95
lr	lr	Both	Events	3	0.0034	341.8	36.0	0.6557	1.075	1.07
lr	lr	Both	min(planned, actual)	1	0.0057	244.5	19.9	0.0584	0.769	0.66
lr	lr	Both	min(planned, actual)	2	0.0099	303.0	27.9	0.4276	0.953	0.86
lr	lr	Both	min(planned, actual)	3	0.0094	341.8	36.0	0.7202	1.075	1.00
lr	lr	Events	Events	1	0.0057	209.0	16.5	0.0140	0.657	0.66
lr	lr	Events	Events	2	0.0099	273.0	23.3	0.2454	0.858	0.86
lr	lr	Events	Events	3	0.0094	318.0	30.5	0.5765	1.000	1.00
lr	lr	Time	Time	1	0.0026	244.5	19.9	0.0333	0.769	0.55
lr	lr	Time	Time	2	0.0082	303.0	27.9	0.3814	0.953	0.77
lr	lr	Time	Time	3	0.0142	341.8	36.0	0.7350	1.075	1.00
mwlr	mwlr	Both	min(planned, actual)	1	0.0014	244.5	19.9	0.0758	0.637	0.49
mwlr	mwlr	Both	min(planned, actual)	2	0.0100	303.0	27.9	0.6716	0.907	0.78
mwlr	mwlr	Both	min(planned, actual)	3	0.0136	341.8	36.0	0.9145	1.087	1.00
mwlr	mwlr	Both	IF	1	0.0051	244.5	19.9	0.1532	0.637	0.64
mwlr	mwlr	Both	IF	2	0.0135	303.0	27.9	0.7305	0.907	0.91
mwlr	mwlr	Both	IF	3	0.0063	341.8	36.0	0.8893	1.087	1.09
mwlr	mwlr	Events	IF	1	0.0014	209.0	16.5	0.0145	0.490	0.49
mwlr	mwlr	Events	IF	2	0.0092	273.0	23.3	0.4403	0.768	0.77
mwlr	mwlr	Events	IF	3	0.0145	318.0	30.5	0.8286	0.976	1.00
mwlr	mwlr	Time	Time	1	0.0026	244.5	19.9	0.1076	0.637	0.55
mwlr	mwlr	Time	Time	2	0.0082	303.0	27.9	0.6568	0.907	0.77
mwlr	mwlr	Time	Time	3	0.0142	341.8	36.0	0.9147	1.087	1.00
Events accrue slower										
lr	lr	Both	Events	1	0.0057	209.0	24.7	0.3913	0.657	0.66
lr	lr	Both	Events	2	0.0099	273.0	35.8	0.8204	0.858	0.86
lr	lr	Both	Events	3	0.0094	318.0	47.5	0.9442	1.000	1.00
lr	lr	Both	min(planned, actual)	1	0.0057	209.0	24.7	0.3913	0.657	0.66
lr	lr	Both	min(planned, actual)	2	0.0092	273.0	35.8	0.8150	0.858	0.85
lr	lr	Both	min(planned, actual)	3	0.0101	318.0	47.5	0.9446	1.000	1.00
lr	lr	Events	Events	1	0.0057	209.0	24.7	0.3913	0.657	0.66
lr	lr	Events	Events	2	0.0099	273.0	35.8	0.8204	0.858	0.86
lr	lr	Events	Events	3	0.0094	318.0	47.5	0.9442	1.000	1.00
lr	lr	Time	Time	1	0.0026	172.3	19.9	0.1262	0.542	0.55
lr	lr	Time	Time	2	0.0082	229.4	27.9	0.5925	0.721	0.77
lr	lr	Time	Time	3	0.0142	273.6	36.0	0.8727	0.860	1.00
mwlr	mwlr	Both	min(planned, actual)	1	0.0012	209.0	24.7	0.3182	0.480	0.48
mwlr	mwlr	Both	min(planned, actual)	2	0.0094	273.0	35.8	0.8764	0.769	0.77
mwlr	mwlr	Both	min(planned, actual)	3	0.0144	318.0	47.5	0.9777	0.978	1.00
mwlr	mwlr	Both	IF	1	0.0012	209.0	24.7	0.3182	0.480	0.48
mwlr	mwlr	Both	IF	2	0.0094	273.0	35.8	0.8764	0.769	0.77
mwlr	mwlr	Both	IF	3	0.0144	318.0	47.5	0.9777	0.978	1.00
mwlr	mwlr	Events	IF	1	0.0012	209.0	24.7	0.3182	0.480	0.48
mwlr	mwlr	Events	IF	2	0.0094	273.0	35.8	0.8764	0.769	0.77
mwlr	mwlr	Events	IF	3	0.0144	318.0	47.5	0.9777	0.978	1.00
mwlr	mwlr	Time	Time	1	0.0026	172.3	19.9	0.1839	0.345	0.55
mwlr	mwlr	Time	Time	2	0.0082	229.4	27.9	0.7072	0.569	0.77
mwlr	mwlr	Time	Time	3	0.0142	273.6	36.0	0.9313	0.772	1.00
Null hypothesis										
lr	lr	Both	Events	1	0.0088	232.4	20.0	0.0091	0.731	0.73
lr	lr	Both	Events	2	0.0128	303.1	27.9	0.0224	0.953	0.95
lr	lr	Both	Events	3	0.0033	348.6	36.0	0.0252	1.096	1.10
lr	lr	Both	min(planned, actual)	1	0.0057	232.4	20.0	0.0058	0.731	0.66
lr	lr	Both	min(planned, actual)	2	0.0099	303.1	27.9	0.0157	0.953	0.86

lr	lr	Both	min(planned, actual)	3	0.0094	348.6	36.0	0.0252	1.096	1.00
lr	lr	Events	Events	1	0.0057	209.0	18.0	0.0059	0.657	0.66
lr	lr	Events	Events	2	0.0099	273.0	24.0	0.0157	0.858	0.86
lr	lr	Events	Events	3	0.0094	318.0	30.1	0.0252	1.000	1.00
lr	lr	Time	Time	1	0.0026	232.3	19.9	0.0026	0.731	0.55
lr	lr	Time	Time	2	0.0082	303.1	27.9	0.0110	0.953	0.77
lr	lr	Time	Time	3	0.0142	348.6	36.0	0.0250	1.096	1.00
mwlr	mwlr	Both	min(planned, actual)	1	0.0014	232.4	20.0	0.0015	0.587	0.49
mwlr	mwlr	Both	min(planned, actual)	2	0.0100	303.1	27.9	0.0117	0.908	0.78
mwlr	mwlr	Both	min(planned, actual)	3	0.0136	348.6	36.0	0.0253	1.119	1.00
mwlr	mwlr	Both	IF	1	0.0036	232.4	20.0	0.0040	0.587	0.59
mwlr	mwlr	Both	IF	2	0.0151	303.1	27.9	0.0192	0.908	0.91
mwlr	mwlr	Both	IF	3	0.0063	348.6	36.0	0.0253	1.119	1.12
mwlr	mwlr	Events	IF	1	0.0014	209.0	18.0	0.0016	0.492	0.49
mwlr	mwlr	Events	IF	2	0.0091	273.0	24.0	0.0108	0.768	0.77
mwlr	mwlr	Events	IF	3	0.0145	318.0	30.1	0.0251	0.977	1.00
mwlr	mwlr	Time	Time	1	0.0026	232.3	19.9	0.0030	0.587	0.55
mwlr	mwlr	Time	Time	2	0.0082	303.1	27.9	0.0112	0.908	0.77
mwlr	mwlr	Time	Time	3	0.0142	348.6	36.0	0.0254	1.119	1.00
Original plan										
lr	lr	Both	Events	1	0.0062	213.2	20.4	0.3670	0.670	0.67
lr	lr	Both	Events	2	0.0101	277.0	28.4	0.7534	0.871	0.87
lr	lr	Both	Events	3	0.0087	321.6	36.8	0.8940	1.011	1.01
lr	lr	Both	min(planned, actual)	1	0.0057	213.2	20.4	0.3567	0.670	0.66
lr	lr	Both	min(planned, actual)	2	0.0099	277.0	28.4	0.7480	0.871	0.86
lr	lr	Both	min(planned, actual)	3	0.0094	321.6	36.8	0.8964	1.011	1.00
lr	lr	Events	Events	1	0.0057	209.0	19.9	0.3375	0.657	0.66
lr	lr	Events	Events	2	0.0099	273.0	27.8	0.7327	0.858	0.86
lr	lr	Events	Events	3	0.0094	318.0	36.0	0.8883	1.000	1.00
lr	lr	Time	Time	1	0.0026	208.8	19.9	0.2461	0.657	0.55
lr	lr	Time	Time	2	0.0082	272.9	27.9	0.6940	0.858	0.77
lr	lr	Time	Time	3	0.0142	318.0	36.0	0.9001	1.000	1.00
mwlr	mwlr	Both	min(planned, actual)	1	0.0013	213.2	20.4	0.2789	0.502	0.49
mwlr	mwlr	Both	min(planned, actual)	2	0.0096	277.0	28.4	0.8011	0.787	0.78
mwlr	mwlr	Both	min(planned, actual)	3	0.0141	321.6	36.8	0.9434	0.994	1.00
mwlr	mwlr	Both	IF	1	0.0016	213.2	20.4	0.2947	0.502	0.50
mwlr	mwlr	Both	IF	2	0.0099	277.0	28.4	0.8058	0.787	0.79
mwlr	mwlr	Both	IF	3	0.0135	321.6	36.8	0.9427	0.994	1.01
mwlr	mwlr	Events	IF	1	0.0013	209.0	19.9	0.2584	0.485	0.49
mwlr	mwlr	Events	IF	2	0.0093	273.0	27.8	0.7836	0.768	0.77
mwlr	mwlr	Events	IF	3	0.0144	318.0	36.0	0.9385	0.977	1.00
mwlr	mwlr	Time	Time	1	0.0026	208.8	19.9	0.3298	0.485	0.55
mwlr	mwlr	Time	Time	2	0.0082	272.9	27.9	0.7817	0.768	0.77
mwlr	mwlr	Time	Time	3	0.0142	318.0	36.0	0.9397	0.977	1.00
Proportional hazards										
lr	lr	Both	Events	1	0.0061	212.4	20.4	0.5328	0.668	0.67
lr	lr	Both	Events	2	0.0105	278.4	28.2	0.7949	0.876	0.88
lr	lr	Both	Events	3	0.0084	324.6	36.3	0.8856	1.021	1.02
lr	lr	Both	min(planned, actual)	1	0.0057	212.4	20.4	0.5236	0.668	0.66
lr	lr	Both	min(planned, actual)	2	0.0099	278.4	28.2	0.7874	0.876	0.86
lr	lr	Both	min(planned, actual)	3	0.0094	324.6	36.3	0.8877	1.021	1.00
lr	lr	Events	Events	1	0.0057	209.0	20.1	0.5156	0.657	0.66
lr	lr	Events	Events	2	0.0099	273.0	27.5	0.7775	0.858	0.86
lr	lr	Events	Events	3	0.0094	318.0	34.9	0.8797	1.000	1.00
lr	lr	Time	Time	1	0.0026	207.0	19.9	0.4045	0.651	0.55
lr	lr	Time	Time	2	0.0082	275.6	27.9	0.7424	0.866	0.77
lr	lr	Time	Time	3	0.0142	322.9	36.0	0.8914	1.015	1.00
mwlr	mwlr	Both	min(planned, actual)	1	0.0013	212.4	20.4	0.3209	0.501	0.49
mwlr	mwlr	Both	min(planned, actual)	2	0.0097	278.4	28.2	0.7302	0.794	0.78
mwlr	mwlr	Both	min(planned, actual)	3	0.0140	324.6	36.3	0.8785	1.008	1.00
mwlr	mwlr	Both	IF	1	0.0016	212.4	20.4	0.3345	0.501	0.50
mwlr	mwlr	Both	IF	2	0.0104	278.4	28.2	0.7398	0.794	0.79
mwlr	mwlr	Both	IF	3	0.0131	324.6	36.3	0.8776	1.008	1.02
mwlr	mwlr	Events	IF	1	0.0013	209.0	20.1	0.3113	0.487	0.49
mwlr	mwlr	Events	IF	2	0.0092	273.0	27.5	0.7139	0.769	0.77
mwlr	mwlr	Events	IF	3	0.0144	318.0	34.9	0.8709	0.977	1.00
mwlr	mwlr	Time	Time	1	0.0026	207.0	19.9	0.3863	0.480	0.55
mwlr	mwlr	Time	Time	2	0.0082	275.6	27.9	0.7208	0.780	0.77
mwlr	mwlr	Time	Time	3	0.0142	322.9	36.0	0.8765	1.000	1.00

```
power_summary <- result_summary |>
  group_by(Scenario, test, Cut, Spending) |>
  summarize(
```

```
    Power = last(Power),
    Events = last(`E(Events)`),
    "Final spend" = last(`E(Spend)`)
  ) |>
  mutate(CutSp = interaction(Cut, Spending))

# Create plots to visualize results
p1 <- power_summary |>
  ggplot(aes(x = Scenario, y = Power, fill = CutSp)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~test) +
  cowplot::theme_half_open() +
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  theme(axis.text.x = element_text(size = 9)) +
  ylab("Power") +
  xlab("Scenario") +
  labs(fill = "Cut.Spending")

p2 <- power_summary |>
  ggplot(aes(x = Scenario, y = `Events`, fill = Cut)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~test) +
  cowplot::theme_half_open() +
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  theme(plot.margin = margin(r = 108)) +
  ylab("Mean Events") +
  xlab("Scenario")

p3 <- power_summary |>
  ggplot(aes(x = Scenario, y = `Final spend`, fill = CutSp)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~test) +
  cowplot::theme_half_open() +
  cowplot::background_grid(major = "x") +
  ggsci::scale_fill_observable(alpha = 0.8) +
  coord_flip() +
  theme(axis.text.x = element_text(size = 6)) +
  ylab("Final nominal p") +
  xlab("Scenario")

# Display plots with increased height
cowplot::plot_grid(
  p1, p2, p3,
  ncol = 1, labels = c("A", "B", "C"), rel_heights = c(1, 1, 1)
)
```



9 Summary

We have suggested the concept of spending time and provided 3 examples where the approach can be applied to simply adapt a design in a useful way. That is, if design assumptions are not met:

- execution is automatically adapted without protocol amendment,
- if there is a delayed treatment effect a trial can continue long enough and adequate α-spending can be preserved to the final analysis where the full treatment effect can be realized and evaluated,
- if testing in multiple populations, spending can be automatically adapted so that α is preserved until the final analysis and tests in all populations of interest can be maintained.

There are many potential solutions to issues that may arise related to error spending in group sequential trials. Michael A. Proschan, Lan, and Wittes (2006) discuss approaches such as using calendar time or adjusting future spending time on an adaptive basis at the time of analysis. We propose the concept of spending time as a more flexible alternative than information fraction for allocation of error spending between interim and final analyses in a group sequential trial. In particular, we have used spending time to ensure adequate Type I error spending is preserved for the final analysis in a trial, and timing of final analysis is not premature.

- For the case of a single endpoint where there may be a delayed treatment effect, we have proposed spending using the minimum of planned and observed information fraction at the time of an analysis. This is combined with potentially delaying the final analysis to achieve at least a minimum trial duration or follow-up duration for all study participants as well as achieving targeted events (statistical

information).

- When testing multiple hypotheses on a single endpoint, we propose the minimum spending fraction based on all hypotheses being evaluated; this is in line with the suggestion of Follman, Proschan, and Geller (1994) for testing multiple experimental arms versus a common control in a group sequential trial. As an example, consider an oncology trial where both progression free survival (PFS) and overall survival (OS) are tested. Timing of interim analyses may be primarily determined by time and PFS event counts, while the final analysis is determined by time and OS event counts. Spending time can be used to determine interim and final α-spending separately for PFS and OS in a way that adapts simply for all hypotheses tested. This includes the ability to use multiple testing procedures such as the graphical method of Maurer and Bretz (2013).

For the biomarker positive and overall population testing, adapting the overall sample size and spending based on enrolling the targeted biomarker positive sample size and spending according to events achieved in the biomarker subgroup for both population was effective at achieving targeted power and completing the trial in a timely fashion. The Fleming-Harrington-O'Brien spending approach mimicking Haybittle-Peto bounds was also effective at maintaining power and controlling Type I error. Depending on whether or not you wish to be conservative for early stopping (Haybittle-Peto) or more liberal with early stopping (O'Brien-Fleming-like bound), either approach can be used.

The scenarios considered are diverse, but not comprehensive. However, they illustrate several issues that should be considered at the time of design of a group sequential trial. Examples have been demonstrated with the design software of the gsDesign2 R package and the simulation R package simtrial.

10 Session information

```
sessioninfo::session_info(pkgs = "attached", info = "packages")
```

```
#> = Session info =============================================================
#> - Packages -----------------------------------------------------------------
#>  ! package    * version date (UTC) lib source
#>  P cowplot    * 1.2.0   2025-07-07 [?] RSPM (R 4.5.0)
#>  P doFuture   * 1.1.2   2025-07-14 [?] RSPM (R 4.5.0)
#>  P dplyr      * 1.1.4   2023-11-17 [?] RSPM (R 4.5.0)
#>  P foreach    * 1.5.2   2022-02-02 [?] RSPM (R 4.5.0)
#>  P future     * 1.67.0  2025-07-29 [?] RSPM (R 4.5.0)
#>  P ggplot2    * 4.0.0   2025-09-11 [?] CRAN (R 4.5.1)
#>  P gsDesign   * 3.7.0   2025-08-25 [?] CRAN (R 4.5.1)
#>  P gsDesign2  * 1.1.5   2025-06-27 [?] CRAN (R 4.5.1)
#>  P gt         * 1.0.0   2025-04-05 [?] CRAN (R 4.5.0)
#>  P parallelly * 1.45.1  2025-07-24 [?] RSPM (R 4.5.0)
#>  P purrr      * 1.1.0   2025-07-10 [?] RSPM (R 4.5.0)
#>  P simtrial   * 1.0.0   2025-06-11 [?] CRAN (R 4.5.1)
#>  P tibble     * 3.3.0   2025-06-08 [?] RSPM (R 4.5.0)
#>  P tictoc     * 1.2.1   2024-03-18 [?] CRAN (R 4.5.0)
#>
#>  [1] C:/Users/me/delayed-effect-simulation/renv/library/windows/R-4.5/x86_64-w64-mingw32
#>  [2] C:/Users/me/AppData/Local/R/cache/R/renv/sandbox/windows/R-4.5/x86_64-w64-mingw32/0eea1ca5
#>
#>  * -- Packages attached to the search path.
#>  P -- Loaded and on-disk path mismatch.
#>
#> ------------------------------------------------------------------------------
```

References

Fleming, Thomas R, David P Harrington, and Peter C O'Brien. 1984. "Designs for Group Sequential Tests." *Controlled Clinical Trials* 5 (4): 348–61.

Follman, Dean A., Michael A. Proschan, and Nancy L. Geller. 1994. "Monitoring Pairwise Comparisons in Multi-Armed Clinical Trials." *Biometrics* 50: 325–36.

Lan, K. K. G., and David L. DeMets. 1983. "Discrete Sequential Boundaries for Clinical Trials." *Biometrika* 70: 659–63.

———. 1989. "Group Sequential Procedures: Calendar Versus Information Time." *Statistics in Medicine* 8: 1191–98.

Lin, Ray S, Ji Lin, Satrajit Roychoudhury, Keaven M. Anderson, Tianle Hu, Bo Huang, Larry F Leon, et al. 2020. "Alternative Analysis Methods for Time to Event Endpoints Under Nonproportional Hazards: A Comparative Analysis." *Statistics in Biopharmaceutical Research* 12 (2): 187–98.

Magirr, Dominic, and Carl-Fredrik Burman. 2019. "Modestly Weighted Logrank Tests." *Statistics in Medicine* 38 (20): 3782–90.

Maurer, Willi, and Frank Bretz. 2013. "Multiple Testing in Group Sequential Trials Using Graphical Approaches." *Statistics in Biopharmaceutical Research* 5: 311–20.

Mukhopadhyay, Pralay, Wenmei Huang, Paul Metcalfe, Fredrik Öhrn, Mary Jenner, and Andrew Stone. 2020. "Statistical and Practical Considerations in Designing of Immuno-Oncology Trials." *Journal of Biopharmaceutical Statistics* 30 (6): 1130–46.

Proschan, Michael A, Dean A Follmann, and Myron A Waclawiw. 1992. "Effects of Assumption Violations on Type I Error Rate in Group Sequential Monitoring." *Biometrics*, 1131–43.

Proschan, Michael A., K. K. Gordon Lan, and Janet Turk Wittes. 2006. *Statistical Monitoring of Clinical Trials. A Unified Approach.* New York, NY: Springer.

Roychoudhury, Satrajit, Keaven M. Anderson, Jiabu Ye, and Pralay Mukhopadhyay. 2021. "Robust Design and Analysis of Clinical Trials with Non-Proportional Hazards: A Straw Man Guidance from a Cross-Pharma Working Group." *Statistics in Biopharmaceutical Research*, 1–37.

Tsiatis, Anastasios A. 1982. "Repeated Significance Testing for a General Class of Statistics Use in Censored Survival Analysis." *Journal of the American Statistical Association* 77: 855–61.

Zhao, Yujie, Yilong Zhang, and Keaven M Anderson. 2024. "Group Sequential Design Under Non-Proportional Hazards: Methodologies and Examples." In *Biostatistics in Biopharmaceutical Research and Development: Clinical Trial Design, Volume 1*, 219–34. Springer.