Simulating Operating Characteristics of Longitudinal Markov Ordinal Outcome Trials

Frank Harrell
VICTR Research Methods Program
Department of Biostatistics
Vanderbilt University School of Medicine
CONNECTS VUMC Study Design and Analysis Core 5

2021-01-24

Introduction

This report provides a template for simulating operating characteristics for treatment comparisons in a longitudinal study with an ordinal outcome Y where the serial correlation is modeled using a first-order Markov process. This within-patient correlation model is specified by conditioning on the ordinal outcome at the previous time point just as though it were any covariate, with one exception. Since the time gaps between measurements (which are made on days 1, 3, 7, 14, 28) are not constant, an interaction term is added to the model so that the influence of the previous state wanes as the time gap increases.

Results for both a frequentist analysis and a Bayesian analysis are provided. Only one data look is taken, at the final sample size. For the Bayesian calculations, the simulations presented here assume that a normal distribution is an adequate approximation to the distribution of the treatment effect estimate, here a treatment B : treatment A transition log odds ratio. The data model is a proportional odds ordinal logistic model analyzed with the R `rms` package `lrm` function. Simulations are done by the `Hmisc` package `simMarkovOrd` and `estSeqMarkovOrd` functions. The `Hmisc` function `soprobMarkovOrd` is also exemplified. This function computes exact state occupancy probabilities from the simulation model. The `Hmisc` `intMarkovOrd` function is used to compute the proportional odds model intercepts that satisfy specified state occupancy probabilities for Y=1, 2, 3, 4 at day 28. These intercepts are used in the simulations.

The new functions are in `Hmisc` version 4.4-3. Until this is available from CRAN, source, Linux binary, and Windows binary versions are available here.

As implemented in the R `Hmisc` package `gbayes` function, the Bayesian posterior distribution used to approximate (thus avoiding another simulation loop for MCMC posterior draws) the real posterior distribution is normal, making for very quick approximate posterior probability calculations when the prior distribution used for the log OR is also Gaussian as used here. Three priors are used:

- a skeptical prior for assessing evidence for efficacy
- a flat prior for assessing evidence for efficacy
- a flat prior for studying posterior probabilities of inefficacy/harm
- an optimistic prior for inefficacy/harm

See here and here for more simulations and graphical presentations of them.

The `estSeqMarkovOrd` function also optionally provides, at the last data look only, the Cox proportional hazards χ^2 statistic for treatment for each simulated clinical trial, and the χ^2 statistic for testing proportional hazards in this unadjusted Cox model. The Cox test for treatment is done after the simulated serial ordinal responses for a patient are summarized with the time until achieving Y=1, with this time right censored at

28 if the patient never achieved Y=1 within 28d. The purpose of this additional simulation is to compare the frequentist power of a two-sample comparison of time until Y=1 with the frequentist power of the transition odds ratio at day 28 in the Markov proportional odds model, and to gauge the extent to which the time-to-event variable induced by the Markov model operates in proportional hazards across the two treatment groups.

Simulation Parameters

The ordinal outcome variable is taken to have values Y=1,2,3,4 measured on days 1, 3, 7, 14, 28.

The treatment effect on transition probabilities, on the log odds ratio scale, is taken to be linear in time, i.e., the treatment effect is zero on day 1 and linearly increases to its maximum on day 28. This maximum value is used in the graphical output and in specifying the underlying true treatment effect. Since a large number of true ORs are used in the simulation, we simulate 1000 clinical trials per OR. ORs will vary from benefit to harm over this sequence: 0.4, 0.5, 0.6, …, 1.0, 1.25. The sample size is 600 patients. Frequentist power is computed at $\alpha = 0.05$. Posterior probabilities will be computed for the following assertion and prior distribution combinations:

- Efficacy: P(OR < 1) with a skeptical prior P(OR < 0.5) = 0.05
- Efficacy: P(OR < 1) with a flat prior (log(OR) has mean 0 and SD 100)
- Inefficacy/harm: P(OR > 1) with a flat prior (log(OR) has mean 0 and SD 100)
- Inefficacy/harm: P(OR > 1) with an optimistic prior (log(OR) has mean log(0.85) = -0.1625 and SD of 0.5)

Simulation Model Specification

The `Hmisc` functions we will use requires a user-specified function `g` that computes the transition model linear predictor other than the proportional odds model intercept terms. `g` computes the linear predictor for one observation, or for a series of initial states (Y values at baseline = time 0). The arguments to `g` are the previous value of Y (the initial state if `t=1`), the current time `t`, the gap between the previous measurement time and `t`, covariate setting `X`, and `parameter` which specifies the true treatment effect.

Specify the simulation model and solve for the intercepts to meet certain criteria. The `Hmisc` `intMarkovOrd` function uses the standard R function `nlm` to compute the intercepts satisfying given occupancy probability targets, once the user specifies a vector of initial guesses for the intercepts. Note that the intercept values must be in decreasing order. `nlm` uses an efficient trial-and-error process to compute the intercept values. With measurements made on days 1, 3, 7, 14, 28, our target values are state occupancy probabilities of 0.7, 0.14, 0.1, and 0.05 for, respectively, Y=1, 2, 3, 4. These probabilities pertain to patients who start with Y=2 at baseline.

```
times <- c(1, 3, 7, 14, 28)
g <- function(yprev, t, gap, X, parameter=-0.5)
  (yprev - 2) * 0.5 * (1 - (gap - 1) / 14) - t / 8 +
  parameter * (X == 2) * (t - 1) / 27
start  <- c(1, 0, -2)   # guesses at intercepts
target <- c(0.7, 0.15, 0.1, 0.05)
ints <- intMarkovOrd(1:4, times, initial=2, absorb=4,
                     intercepts=start, g=g, target=target, t=28)
```

```
Iterations: 37
Sum of absolute errors: 6.634046e-07
Intercepts: 2.467 1.36 -3.94

Occupancy probabilities for group 1:
       1     2     3     4
1  0.088 0.138 0.758 0.017
3  0.084 0.130 0.751 0.035
7  0.132 0.178 0.644 0.046
14 0.283 0.249 0.418 0.049
28 0.700 0.150 0.100 0.050

Occupancy probabilities for group 2:

       1     2     3     4
1  0.088 0.138 0.758 0.017
3  0.087 0.133 0.745 0.035
7  0.146 0.189 0.622 0.044
14 0.336 0.255 0.362 0.047
28 0.784 0.106 0.063 0.047

Log odds ratios at time 28 from occupancy probabilities: -0.44 -0.352 -0.061
```

```r
# Repeat without absorbing state
intsna <- intMarkovOrd(1:4, times, initial=2,
                       intercepts=c(1, 0.5, -1), g=g, target=target, t=28)
```

```
Iterations: 68
Sum of absolute errors: 1.258402e-06
Intercepts: 2.629 1.741 0.531

Occupancy probabilities for group 1:

       1     2     3     4
1  0.076 0.090 0.234 0.600
3  0.058 0.070 0.191 0.681
7  0.092 0.103 0.245 0.560
14 0.228 0.186 0.285 0.302
28 0.700 0.150 0.100 0.050

Occupancy probabilities for group 2:

       1     2     3     4
1  0.076 0.090 0.234 0.600
3  0.060 0.072 0.195 0.674
7  0.102 0.111 0.255 0.532
14 0.275 0.200 0.273 0.251
28 0.795 0.109 0.065 0.031

Log odds ratios at time 28 from occupancy probabilities: -0.506 -0.506 -0.506
```

```r
# Reproduce the result with absorbing state using soprobMarkovOrd directly
s <- soprobMarkovOrd(1:4, times, initial=2, absorb=4,
                     intercepts=ints, g=g, X=1)
round(s, 3)
```

```
       1     2     3     4
1  0.088 0.138 0.758 0.017
3  0.084 0.130 0.751 0.035
7  0.132 0.178 0.644 0.046
14 0.283 0.249 0.418 0.049
```

```
28 0.700 0.150 0.100 0.050
```

Make sure the sum of absolute errors is small, otherwise the optimization algorithm may have been fooled and you'll need to try different starting values in `start`.

Before running the clinical trial simulation, simulate a large sample size for one trial to check that the simulation is working correctly. Unlike what follows later, we will carry the absorbing state forward so that we can compute state occupancy proportions easily.

```
s <- simMarkovOrd(n=40000, 1:4, times, initial=2, X=c(group=1),
                                       absorb=4, intercepts=ints, g=g, carry=TRUE)
relfreq <- function(x) table(x) / length(x)
w       <- with(s, tapply(y, time, relfreq))
do.call(rbind, w)
```

```
          1        2        3        4
1  0.086100 0.139850 0.758125 0.015925
3  0.084575 0.128175 0.752825 0.034425
7  0.133075 0.180550 0.640925 0.045450
14 0.278475 0.251725 0.419975 0.049825
28 0.700375 0.150350 0.098775 0.050500
```

Check the correlation structure after reshaping the dataset to have different times in columns for a subject.

```
setDT(s, key=c('id', 'time'))
w <- dcast(s, id ~ time, value.var='y')
w <- as.matrix(w)[, -1]
round(cor(w), 2)
```

```
      1    3    7   14   28
1  1.00 0.20 0.08 0.07 0.11
3  0.20 1.00 0.23 0.15 0.20
7  0.08 0.23 1.00 0.26 0.25
14 0.07 0.15 0.26 1.00 0.30
28 0.11 0.20 0.25 0.30 1.00
```

Simulation

Using the above model we simulate, for each treatment effect, 1000 clinical trials each with 600 observations. For each trial the treatment groups are randomly assigned with probability 1/2 each. The high-level `estSeqMarkovOrd` function calls the `simMarkovOrd` function to simulate each trial. We use a distribution of initial states 1, 2, 3 and sample from that distribution to get each patient's baseline state. Because the contrast of interest needs to take into account a treatment × time interaction, the `groupContrast` argument must be specified below. A second set of simulations is also run and stored for later, just for OR = 0.6 and using a logistic model that has two unecessary parameters: a square term for the time effect and a square term for the time × treatment interaction. This second simulation will be used to see how much power is lost by having more complexity involving treatment.

```
initial <- c('1'=0.02, '2'=0.75, '3'=0.23)    # probabilities of being in baseline states
ors     <- c(0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25)
formula  <- y ~ yprev * gap + time * group
formula2 <- y ~ yprev * gap + pol(time, 2) * group
nsim <- 1000
if(simdone) {
    est  <- readRDS('sim.rds')
    est2 <- readRDS('sim2.rds')
    } else {
```

```
    set.seed(1)
    # 1000*8 simulations (8 parameter values) takes about 8 minutes in all
  est <- estSeqMarkovOrd(1:4, times, initial, absorb=4, intercepts=ints,
    parameter=log(ors), looks=600, g=g, formula=formula,
    groupContrast=list(list(group=1, time=28), list(group=2, time=28)),
    timecriterion=function(y) y == 1, coxzph=TRUE, nsim=nsim, progress=FALSE)
  saveRDS(est, 'sim.rds')
  set.seed(2)
  est2 <- estSeqMarkovOrd(1:4, times, initial, absorb=4, intercepts=ints,
    parameter=log(0.6), looks=600, g=g, formula=formula2,
    groupContrast=list(list(group=1, time=28), list(group=2, time=28)),
    nsim=nsim, progress=FALSE)
  saveRDS(est2, 'sim2.rds')
}
setDT(est, key='parameter')
est[, OR := exp(parameter)]
setDT(est2)
est2[, OR := exp(parameter)]
```

Before doing the main analyses on transition model parameters, look at the computed event times and estimate the type I assertion probability α and power of the "time to Y=1" variable analyzed with a Cox model, and look at the the magnitude of non-proportional hazards. Also compute the average log hazard ratio to see how the reciprocal of its antilog relates to the odds ratio (reciprocal because the Cox model is fit on time until a *good* outcome).

```
s  <- est
et <- attr(s, 'etimefreq')
et <- apply(et, 2:4, sum)
cat('Frequency distribution of times to Y=1 by group\n')
```

```
Frequency distribution of times to Y=1 by group
```

```
for(pm in dimnames(et)[[1]]) {
    cat('\nOR:', exp(as.numeric(pm)), '\n')
    print(et[pm, ,])
}
```

```
OR: 0.4
      1     3     7    14     28   28+
1 24476 20826 30917 58944 110171 54914
2 24207 22566 36936 77192 109450 29401

OR: 0.5
      1     3     7    14     28   28+
1 24449 21026 30946 58960 109972 54829
2 24340 22033 35527 72491 111164 34263

OR: 0.6
      1     3     7    14     28   28+
1 24233 20985 31024 58815 109661 54834
2 24369 21698 33976 69267 112782 38356

OR: 0.7
      1     3     7    14     28   28+
1 24388 20849 30920 58554 110286 54710
```

```
2 24193 21618 33502 65708 112481 42791

OR: 0.8
      1     3     7    14      28   28+
1 24254 21076 31034 58922 110035 54589
2 24307 21070 32607 63034 111838 47234

OR: 0.9
      1     3     7    14      28   28+
1 24372 20916 31304 59253 110363 54466
2 24337 20707 31546 60923 110798 51015

OR: 1
      1     3     7    14      28   28+
1 24406 20780 31038 58880 109743 54980
2 24256 21010 30974 58893 110120 54920

OR: 1.25
      1     3     7    14      28   28+
1 24114 21062 30969 58940 110418 54684
2 24166 20798 29870 54362 106993 63624
```

```
rn <- function(x) round(x, 2)
rn(s[, .(HR=exp(- mean(loghr))), by=OR])
```

```
     OR   HR
1: 0.40 0.77
2: 0.50 0.82
3: 0.60 0.86
4: 0.70 0.90
5: 0.80 0.94
6: 0.90 0.97
7: 1.00 1.00
8: 1.25 1.08
```

```
hi <- function(x, lab) {hist(x, nclass=50, main='', xlab=lab); invisible()}
s[, hi(lrchisq, 'Cox Group Test Chi-square')]
```

```
NULL
```

```
cat('Power of Cox test for time until Y=1\n')
```

```
Power of Cox test for time until Y=1
```

```
rn(s[, .(power=mean(lrchisq > 3.84)), by=OR])
```

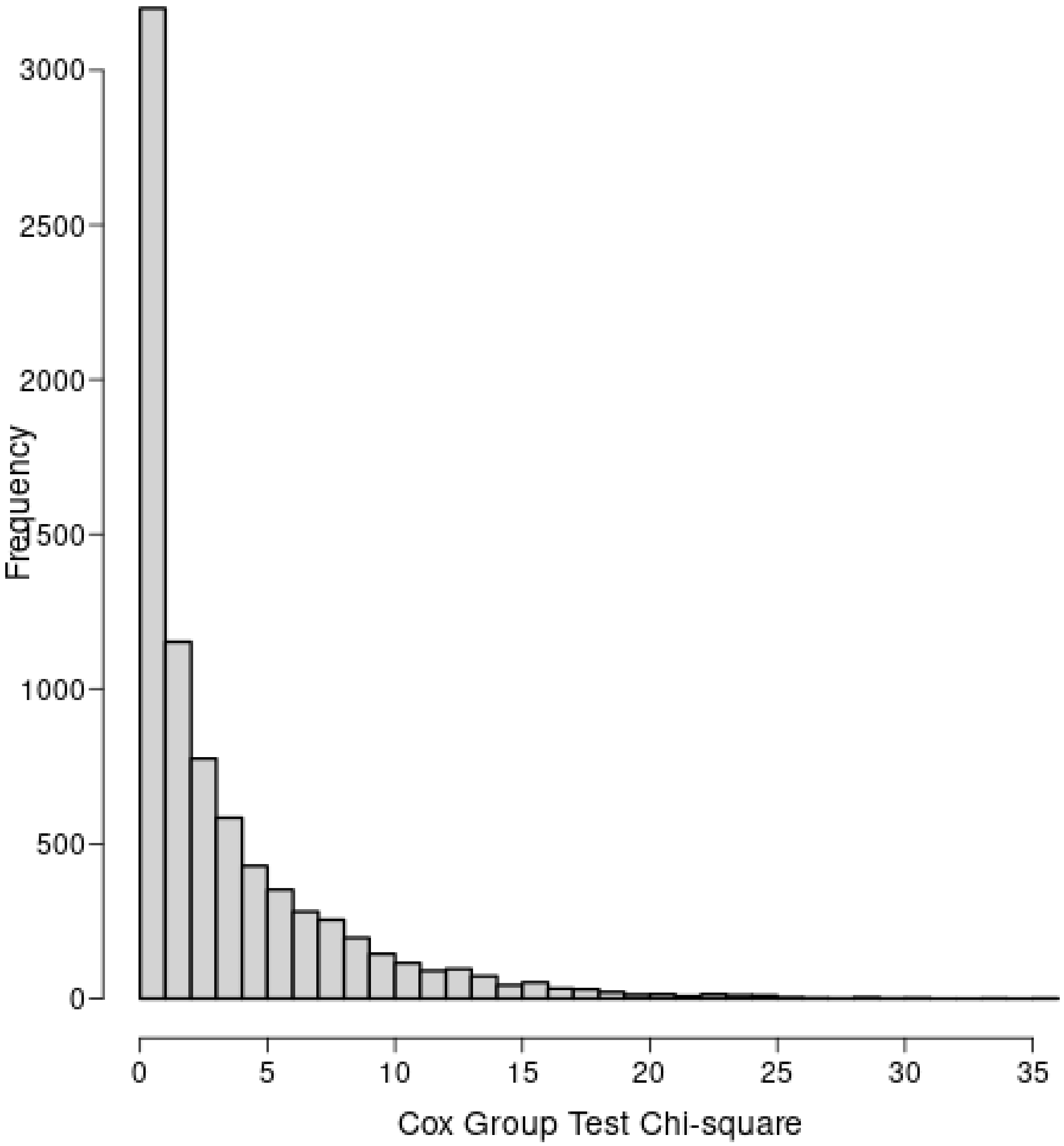
```
     OR power
1: 0.40  0.84
2: 0.50  0.61
3: 0.60  0.41
4: 0.70  0.21
5: 0.80  0.10
6: 0.90  0.06
7: 1.00  0.03
8: 1.25  0.12
```

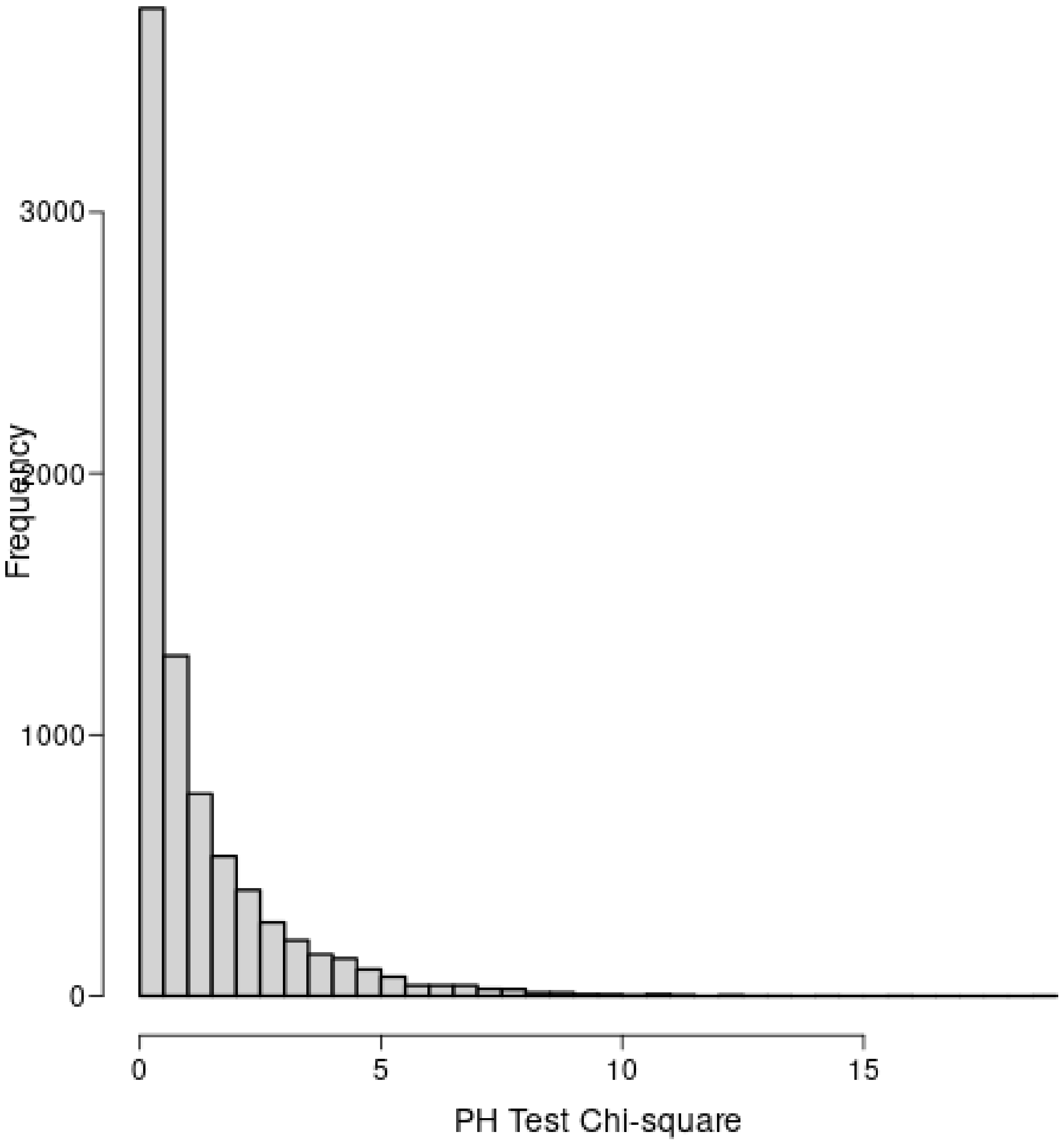
```
s[, hi(phchisq, 'PH Test Chi-square')]
```

```
NULL
```

```
rn(s[, .(rejectPH=mean(phchisq > 3.84)), by=OR])
```

```
     OR rejectPH
1: 0.40     0.12
2: 0.50     0.10
3: 0.60     0.10
4: 0.70     0.06
5: 0.80     0.06
6: 0.90     0.05
7: 1.00     0.05
8: 1.25     0.06
```


3000
2500
2000
1500
1000
500
0
Frequency
0
5
10
15
20
25
30
35
Cox Group Test Chi-square



The hazard ratio, after taking the reciprocal so that a large HR means a worse outcome, is not equatable to the odds ratio except at the null value of 1.0.

There is little evidence of non-proportional hazards.

Compute the power of the Markov model treatment comparison, and compare the χ^2 statistic from the transition model to that from the Cox model. For plotting, we take the square root of the χ^2 statistics to get a more symmetric distribution over the simulations. There is one panel per odds ratio. For OR=0.6 also show the power of a transition model that overfitted time as quadratic.

```
s[, tchi := est * est / vest]
cat('Power of Markov model test at 28d\n\n')
```

```
Power of Markov model test at 28d
```

```
s[, .(power=mean(tchi > 3.84)), by=OR]
```

```
     OR power
1: 0.40 0.997
2: 0.50 0.972
3: 0.60 0.831
4: 0.70 0.548
5: 0.80 0.254
6: 0.90 0.078
7: 1.00 0.053
8: 1.25 0.267
```

```
cat('Power of quadratic in time Markov model at 28d when OR=0.6\n\n')
```

```
Power of quadratic in time Markov model at 28d when OR=0.6
```

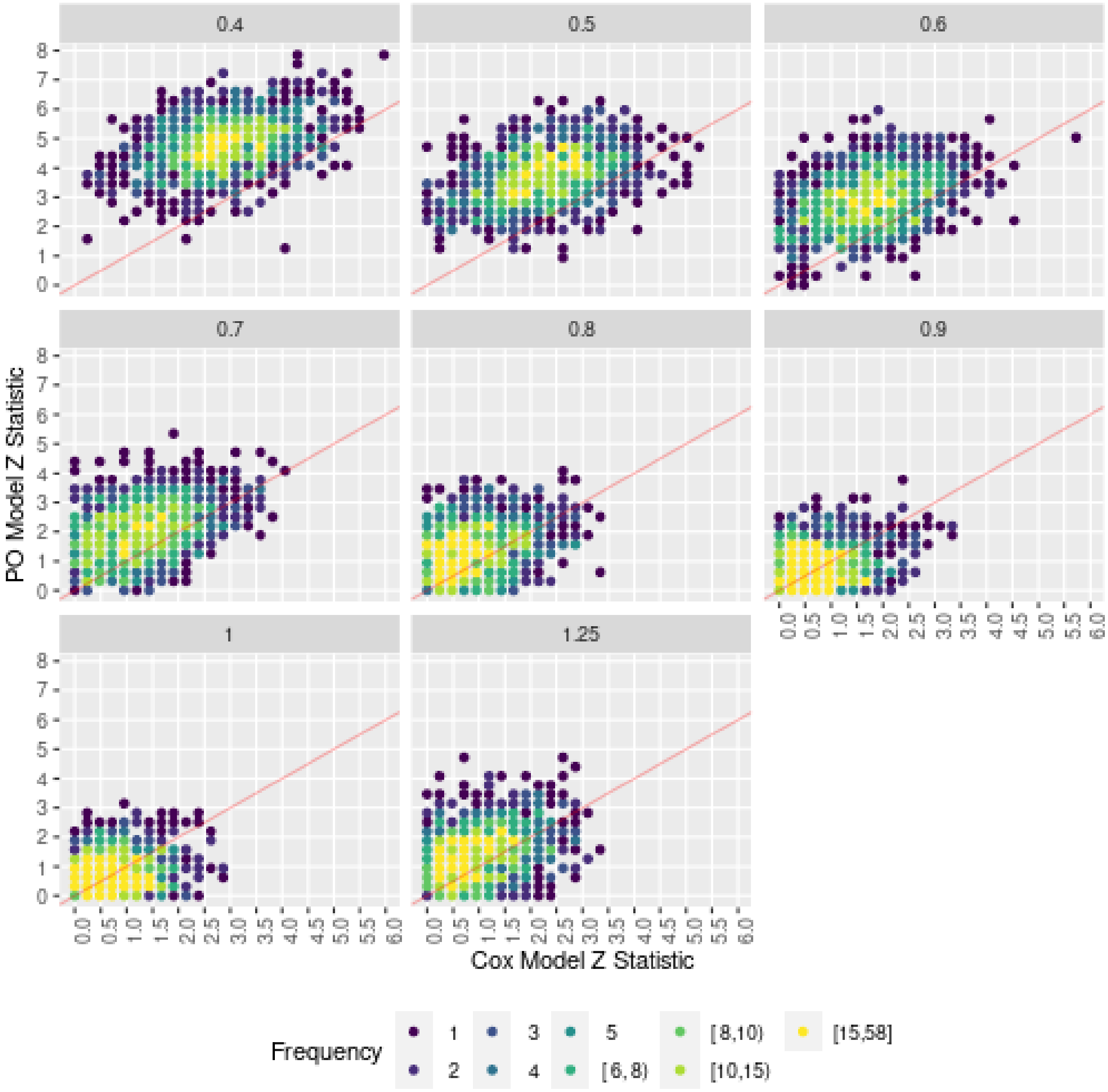
```
est2[, .(power=mean(est * est / vest > 3.84))]
```

```
   power
1: 0.713
```

```
rho <- s[, spearman(tchi, lrchisq)]
cat('Spearman rho correlation between Markov and Cox model chi-square:', rn(rho), '\n')
```

```
Spearman rho correlation between Markov and Cox model chi-square: 0.66
```

```
gr <- function(x, y, z)
    ggfreqScatter(x, y, ylab='PO Model Z Statistic', xlab='Cox Model Z Statistic',
                                by=z, bins=25) +
          geom_abline(intercept=0, slope=1, col='red', alpha=0.3) +
        theme(legend.position='bottom', axis.text.x = element_text(angle = 90))
s[, gr(sqrt(lrchisq), sqrt(tchi), exp(parameter))]
```



The power of the longitudinal transition model is superior to that of the time to Y=1 comparison. The χ^2 statistics from the Markov ordinal longitudinal model are seen to dominate those from the Cox model for time until Y=1 as judged by the scatterplot. The Spearman ρ between the two statistics is 0.66. There is a significant loss of power to detect OR=0.6 when the quadratic model is substituted for the true linear (in time) model.

Bayesian Power

Assume that the distribution of the maximum likelihood estimates is approximately Gaussian, and use a Gaussian prior distribution for the parameters of interest. Then we can use the maximum likelihood estimates already simulated to get approximate Gaussian Bayesian posterior distributions, and quickly compute such things as Bayesian power, e.g., the probability that the posterior probability of a beneficial effect exceeds 0.95 at the end of the study.

Define the Bayesian assertions and priors to be used for them

- Assertion 1: log(OR) < 0 under prior with prior mean 0 and sigma: P(OR>2)=0.025
- Assertion 2: log(OR) < 0 under flat prior
- Assertion 3: log(OR) > 0 under flat prior (sigma=100)
- Assertion 4: log(OR) > 0 under optimistm prior with mean log(0.85), sigma=0.5

```
asserts <- list(list('Efficacy',                   '<', 0, cutprior=log(2), tailprob=0.025),
                list('Efficacy flat',              '<', 0, mu=0,         sigma=100),
                list('Inefficacy/harm flat',       '>', 0, mu=0,         sigma=100),
                list('Inefficacy/harm optimistic', '>', 0, mu=log(0.85), sigma=0.5))

s <- gbayesSeqSim(est, asserts=asserts)

head(s)
```

```
   sim  parameter look        est       vest     loghr   lrchisq   phchisq  OR
1:   1 -0.9162907  600 -0.8464192 0.03248727 0.2731569  9.392125 2.0164343 0.4
2:   2 -0.9162907  600 -1.3748037 0.03884948 0.3514165 15.332860 1.2068413 0.4
3:   3 -0.9162907  600 -1.3463036 0.04016280 0.4252279 23.252161 0.2551884 0.4
4:   4 -0.9162907  600 -0.9642204 0.03604569 0.3115848 12.365327 0.3557162 0.4
5:   5 -0.9162907  600 -1.0711671 0.03659671 0.3452703 15.097609 0.8866927 0.4
6:   6 -0.9162907  600 -0.7515171 0.03502086 0.2838002 10.177597 2.0171424 0.4
       tchi        p1      mean1       sd1        p2      mean2       sd2
1: 22.05250 0.9999857 -0.6718937 0.1605884 0.9999987 -0.8464165 0.1802420
2: 48.65149 1.0000000 -1.0489714 0.1721687 1.0000000 -1.3747983 0.1971023
3: 45.12966 1.0000000 -1.0190612 0.1743575 1.0000000 -1.3462982 0.2004062
4: 25.79285 0.9999962 -0.7485003 0.1672763 0.9999998 -0.9642169 0.1898567
5: 31.35251 0.9999996 -0.8286862 0.1682626 1.0000000 -1.0711632 0.1913023
6: 16.12690 0.9998070 -0.5871186 0.1654081 0.9999704 -0.7515145 0.1871383
             p3      mean3       sd3           p4      mean4       sd4
1: 1.326520e-06 -0.8464165 0.1802420 2.977781e-06 -0.7677677 0.1695615
2: 1.528830e-12 -1.3747983 0.1971023 1.944171e-11 -1.2117546 0.1833694
3: 9.221416e-12 -1.3462982 0.2004062 1.031643e-10 -1.1824504 0.1860206
4: 1.900564e-07 -0.9642169 0.1898567 5.772599e-07 -0.8631950 0.1774921
5: 1.075963e-08 -1.0711632 0.1913023 4.501535e-08 -0.9551381 0.1786716
6: 2.961916e-05 -0.7515145 0.1871383 5.331920e-05 -0.6791462 0.1752650
```

```
attr(s, 'asserts')
```

```
                       label  cutprior tailprob         mu      sigma assertion
1                   Efficacy 0.6931472    0.025  0.0000000   0.353653       < 0
2              Efficacy flat        NA       NA  0.0000000 100.000000       < 0
3       Inefficacy/harm flat        NA       NA  0.0000000 100.000000       > 0
4 Inefficacy/harm optimistic        NA       NA -0.1625189   0.500000       > 0
```

```
alabels <- attr(s, 'alabels')  # named vector to map p1 p2 p3 p4 to labels
```

First let’s examine the effect of the priors by making two pairwise comparisons: differences in posterior probabilities of efficacy under skeptical vs. flat prior, and differences in posterior probabilities of inefficacy under flat and optimistic priors.

```
w <- data.table(s)
u <- w[, .(p12max=max(abs(p1 - p2)), p12mean=mean(abs(p1 - p2)),
           p34max=max(abs(p3 - p4)), p34mean=mean(abs(p3 - p4))), by=.(look)]
z  <- melt(u, measure.vars=c('p12max', 'p12mean', 'p34max', 'p34mean'),
           variable.name='which', value.name='diff')
```

```
k <- c(p12max='Efficacy max',   p12mean='Efficacy mean',
       p34max='Inefficacy max', p34mean='Inefficacy mean')
z[, w := k[which]]
z
```

```
   look   which       diff               w
1:  600  p12max 0.02980499    Efficacy max
2:  600 p12mean 0.01065486   Efficacy mean
3:  600  p34max 0.05205469  Inefficacy max
4:  600 p34mean 0.01235886 Inefficacy mean
```

Compute the probability of hitting assertion-specific targets at the planned study end.

```
# Reshape results into taller and thinner data table so can plot over 3 assertions

ps <- names(alabels)
m  <- melt(w,
           measure.vars=list(ps, paste0('mean', 1:4), paste0('sd', 1:4)),
           variable.name='assert', value.name=c('p', 'mean', 'sd'))
m[, assert := alabels[assert]]
head(m)
```

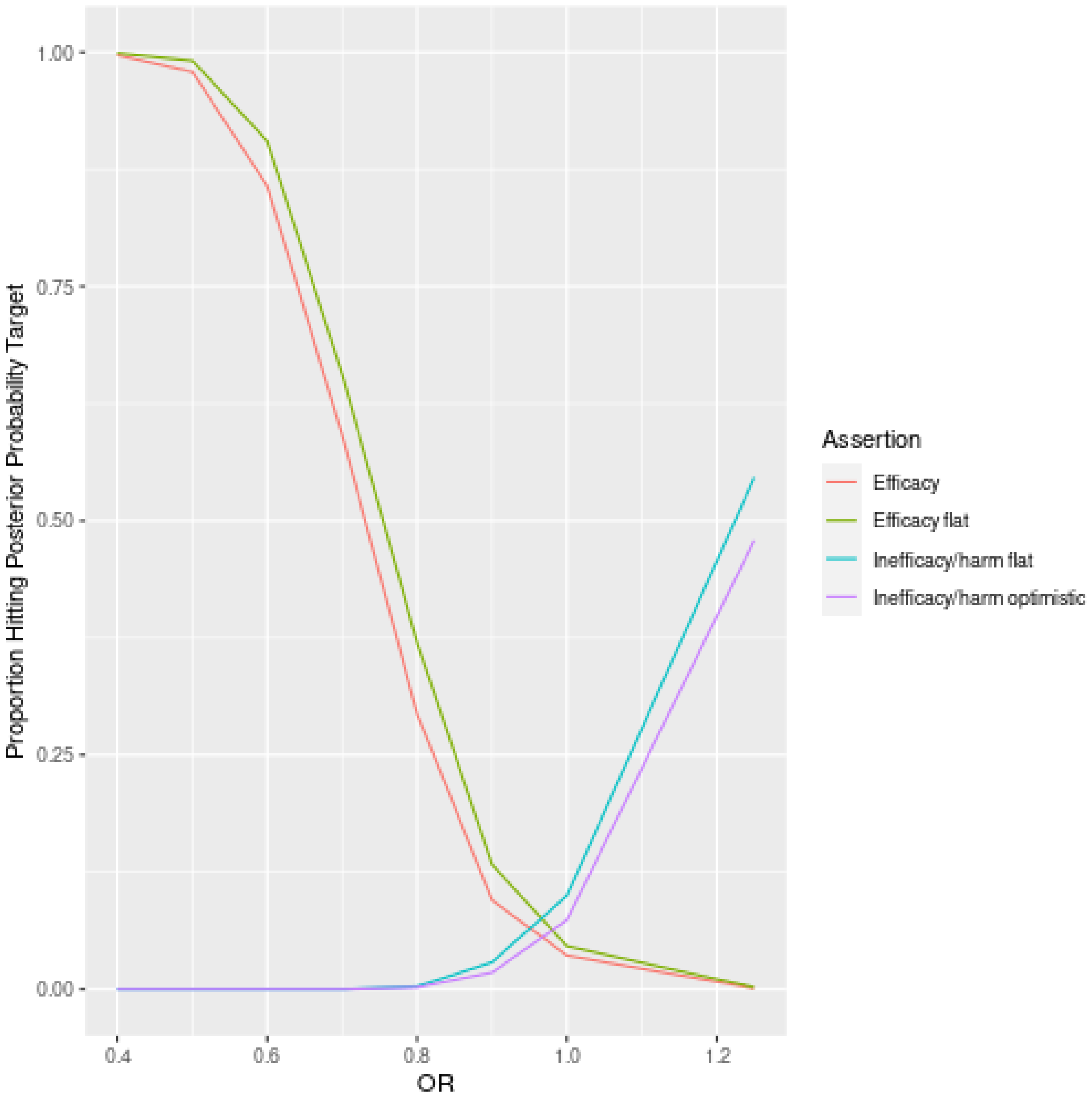
```
   sim  parameter look        est       vest     loghr   lrchisq   phchisq  OR
1:   1 -0.9162907  600 -0.8464192 0.03248727 0.2731569  9.392125 2.0164343 0.4
2:   2 -0.9162907  600 -1.3748037 0.03884948 0.3514165 15.332860 1.2068413 0.4
3:   3 -0.9162907  600 -1.3463036 0.04016280 0.4252279 23.252161 0.2551884 0.4
4:   4 -0.9162907  600 -0.9642204 0.03604569 0.3115848 12.365327 0.3557162 0.4
5:   5 -0.9162907  600 -1.0711671 0.03659671 0.3452703 15.097609 0.8866927 0.4
6:   6 -0.9162907  600 -0.7515171 0.03502086 0.2838002 10.177597 2.0171424 0.4
       tchi   assert         p       mean        sd
1: 22.05250 Efficacy 0.9999857 -0.6718937 0.1605884
2: 48.65149 Efficacy 1.0000000 -1.0489714 0.1721687
3: 45.12966 Efficacy 1.0000000 -1.0190612 0.1743575
4: 25.79285 Efficacy 0.9999962 -0.7485003 0.1672763
5: 31.35251 Efficacy 0.9999996 -0.8286862 0.1682626
6: 16.12690 Efficacy 0.9998070 -0.5871186 0.1654081
```

```
# Define targets
target <- c(Efficacy                     = 0.95,
            'Efficacy flat'              = 0.95,
            'Inefficacy/harm flat'       = 0.9,
            'Inefficacy/harm optimistic' = 0.9)

m[, target := target[assert]]     # spreads targets to all rows
# hit = 0/1 indicator if hitting target at the single fixed sample size
u <- m[, .(hit = mean(p > target)), by=.(OR, assert)]
u
```

```
      OR                     assert   hit
 1: 0.40                   Efficacy 0.997
 2: 0.50                   Efficacy 0.979
 3: 0.60                   Efficacy 0.857
 4: 0.70                   Efficacy 0.591
 5: 0.80                   Efficacy 0.294
 6: 0.90                   Efficacy 0.095
 7: 1.00                   Efficacy 0.036
 8: 1.25                   Efficacy 0.001
 9: 0.40              Efficacy flat 0.999
10: 0.50              Efficacy flat 0.991
11: 0.60              Efficacy flat 0.905
12: 0.70              Efficacy flat 0.656
13: 0.80              Efficacy flat 0.370
14: 0.90              Efficacy flat 0.133
15: 1.00              Efficacy flat 0.046
16: 1.25              Efficacy flat 0.002
17: 0.40       Inefficacy/harm flat 0.000
18: 0.50       Inefficacy/harm flat 0.000
19: 0.60       Inefficacy/harm flat 0.000
20: 0.70       Inefficacy/harm flat 0.000
21: 0.80       Inefficacy/harm flat 0.003
22: 0.90       Inefficacy/harm flat 0.029
23: 1.00       Inefficacy/harm flat 0.100
24: 1.25       Inefficacy/harm flat 0.546
25: 0.40 Inefficacy/harm optimistic 0.000
26: 0.50 Inefficacy/harm optimistic 0.000
27: 0.60 Inefficacy/harm optimistic 0.000
28: 0.70 Inefficacy/harm optimistic 0.000
29: 0.80 Inefficacy/harm optimistic 0.002
30: 0.90 Inefficacy/harm optimistic 0.018
31: 1.00 Inefficacy/harm optimistic 0.074
32: 1.25 Inefficacy/harm optimistic 0.479
      OR                     assert   hit
```

```
ggplot(u, aes(x=OR, y=hit, color=assert)) + geom_line() +
  xlab('OR') + ylab('Proportion Hitting Posterior Probability Target') +
  guides(color=guide_legend(title='Assertion'))
```



Frequentist Power of Single Day Ordinal Outcomes

Let's consider only the detection of OR=0.6 for an unadjusted proportional odds two-group comparison of ordinal outcomes measured at a single day. When simulating the data using the same model as above, we carry absorbing states forward here. So once Y=4 occurs on a given day, Y=4 is considered to be in effect at all later days.

```
nsim <- 1000
if(simdone.single) ch <- readRDS('simch.rds') else {
    ch <- matrix(NA, nrow=nsim, ncol=length(times))
    colnames(ch) <- as.character(times)
    set.seed(3)
    for(isim in 1 : nsim) {
        s1 <- simMarkovOrd(n=300, 1:4, times, initial=2, X=c(group=1),
                                               absorb=4, intercepts=ints, g=g, carry=TRUE)
        s2 <- simMarkovOrd(n=300, 1:4, times, initial=2, X=c(group=2),
                                               absorb=4, intercepts=ints, g=g, carry=TRUE)
        s <- rbind(s1, s2)
        for(tim in times) {
            f <- lrm(y ~ group, data=s, subset=time == tim)
            ch[isim, as.character(tim)] <- f$stats['Model L.R.']
        }
    }
    saveRDS(ch, 'simch.rds')
}
apply(ch, 2, function(x) mean(x > 3.84))
```

```
    1     3     7    14    28
0.057 0.061 0.101 0.324 0.621
```

The power for testing differences on day 1 has to only be α because the true treatment effect is zero on that day. The power increases as time marches on. But even on day 28 the power is significantly below the power of the ordinal longitudinal model that uses all days.

More Information

- Full R markdown script
- COVID-19 statistical resources
- Bayesian design and analysis resources

Computing Environment

To cite R in publication use:

R Core Team (2020). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.