

Simulating Operating Characteristics of a Sequential Longitudinal Markov Ordinal Outcome Trial

Frank Harrell
Department of Biostatistics
Vanderbilt University School of Medicine

2021-05-27

Contents

Summary	1
Overview	2
R Setup	3
Simulation Model	4
Transition Odds Ratios vs. Change in Mean Time Unrecovered	8
Simulation of Bayesian Sequential Trial	10
Simulate a Single Large Trial	10
Approximating the Standard Error of the Treatment Effect	12
Simulation	18
Partial Proportional Odds Model Simulation	18

Summary

The performance of pairwise treatment comparison tests is simulated for 3-level longitudinal ordinal responses. The setup is a special case of longitudinal continuous response assessments with an override for deaths, which are counted as the highest response category. Here the continuous daily assessment is dichotomized so that on a given day the response $Y=0$ for fully recovered and alive, $Y=1$ for alive but not fully recovered, and $Y=2$ for dead. Simulations are done for estimating frequentist test power for both a longitudinal discrete time Markov proportional odds ordinal logistic model and a Cox model for time until recovery, the latter considering death at any time to represent no recovery by 21 days (time to recovery right censored at 21d). $Y=0$ and 2 are absorbing states, so once a patient recovers she is assumed to stay recovered. Under ideal conditions (constant treatment effect over time, proportional odds holds for the treatment effect, proportional hazards holds for time to recovery) the transition model odds ratio equals the Cox model hazard ratio, and the two tests provide equivalent test statistics and power. Note that for the general case where there are many levels of Y , the power of the longitudinal ordinal model will greatly exceed that of the time to recovery Cox test as demonstrated [here](#). Note also that when Y has 3 levels and two of them are absorbing, the Markov transition model has no need to use the previous state as a covariate (since it's constant), which is another reason that the Cox-ordinal longitudinal model equivalence will not hold in general. For the current limited- Y case the Markov model provides added flexibility and a simpler interpretation in the presence of competing events, by estimating these quantities:

- the probability that $Y \geq 1$ as a function of time and treatment (being not recovered or dead)

- the probability that $Y=1$ as a function of time and treatment (being alive but not recovered)
- the probability of being dead as a function of time and treatment
- the mean number of days unrecovered (days with $Y>0$) by treatment

The last quantity is particularly clinically interpretable, being the expected number of days within the 21 day follow-up window for which the patient has died or is alive but not yet recovered.

For comparing two equal size treatment groups of $n = 89$ patients each, the frequentist power at $\alpha = 0.05$ for both statistical tests is given below, along with the hazard and odds ratios to be detected, and the corresponding reduction in mean days unrecovered.

Effect RatioHR = OR	Reduction inDays Unrecovered	CoxPower	OrdinalPower
0.55	1.7	0.92	0.92
0.6	1.4	0.81	0.81

Bayesian operating characteristics will be added to this report by 2021-05-28.

Overview

Consider a clinical trial for the treatment of Stevens-Johnson syndrome / toxic epidermal necrolysis, a rare and life-threatening severe cutaneous adverse reaction induced primarily by drugs. Our goal is to estimate Bayesian operating characteristics for a two treatment arm comparison with four pre-planned data looks with optional stopping for efficacy, inferiority, or futility. A first-order discrete time Markov Bayesian partial proportional odds (PO) model is used, with daily patient assessment on days 1, 2, ..., 21 after randomization. The underlying daily measurement is the percent body surface area for which skin has not re-attached (re-epithelialization), with death considered as worse than any level of skin detachment. The most powerful analysis will treat % BSA involvement as a continuous ordinal measurement with death override. But we now concern ourselves with a simpler daily outcome: $Y=0$ (alive and completely recovered), $Y=1$ (alive but some skin remains detached), and $Y=2$ (dead). Consider $Y=0$ and $Y=2$ to be absorbing states, so per-patient serial data records stop once either of these is achieved.

Once Markov transition probabilities are transformed into state occupancy probabilities (SOPs), these can be used as primary estimands or a combination of the 21 days of SOPs can be summarized. In the study the primary estimand is the difference between treatments in the mean time not recovered, which is the difference in the sums over the 21 days of the probability that $Y>0$. Were there to be no deaths, the mean time not recovered is just one day less than the mean time to recovery. For the purpose of the simulations presented here the estimand will be the transition odds ratio in the Markov PO model. Note that the PO model places death as worse than non-recovery, but makes no assumption about the spacing of categories, i.e., going from death to not recovered is not assumed to be the same as going from not recovered to recovered.

It is common practice to analyze such data by collapsing each patient's raw data into a single number—time to recovery—and then applying a Cox model comparison or its competing risk analog. For these analyses, death is considered a censoring event, and patients who die are censored by treating them as non-recovered on day 28. Thus a death at day 2 is considered as serious as a death at day 27 and these are considered as serious as being alive without recovery on day 28. It can be difficult to interpret time until a good outcome when the outcome can be interrupted by a bad outcome. State transition models provide estimates of probabilities of easy-to-interpret conditions, e.g. day-specific $P(\text{dead})$, $P(\text{alive but not recovered})$, or $P(\text{either being dead or being alive but not recovered})$. The easy generalization of such models to fully use the continuous version of the daily outcome (% body surface area involvement) is another advantage.

As an aside, it is customary to summarize death-censored time-to-recovery using the median. The median requires one to have continuous measurements, and ties in the day of recovery across patients makes the median simultaneously inefficient and non-robust. Adding one patient to the study can alter the median by a whole day. One would need to have hourly assessments for the median number of days to have good statistical behaviour. These same problems do not exist for the mean time to recovery.

More statistical background and statistical computing details are available [here](#). As detailed [here](#), the compound symmetric correlation pattern assumption of a simpler random effects ordinal model is severely violated for designs like ours, hence the use of the Markov serial correlation pattern. Note that even when the proportional odds model is strongly violated for treatment, a simple function of the odds ratio (OR) from the model almost exactly [reproduces the Wilcoxon statistic](#) or *concordance probability*. The Wilcoxon statistic, when scaled to 0-1 (i.e., the concordance probability) is to within an average absolute error of 0.002 equal to $\frac{OR^{0.66}}{1+OR^{0.66}}$. The most important non-PO effect to model is due to the possible change in the mix of outcomes over follow-up time—an effect not necessarily involving treatment. This is modeled using the [Peterson and Harrell \(1990\) partial proportional odds model](#).

The new study in question is a 3-arm study with a planned maximum sample size of 267 patients with looks after follow-up is complete on 111, 162, and 213 patients. Here we consider 2-arm comparisons so the combined sample sizes are 74, 108, 142, and the final look at 178 patients.

R Setup

```

outfmt <- if(knitr::is_html_output ()) 'html' else 'pdf'
markup <- if(knitr::is_latex_output()) 'latex' else 'html'
require(rms)           # automatically engages rms(Hmisc)

require(data.table)   # used to compute various summary measures

if(outfmt == 'html') require(plotly)
fdev <- switch(outfmt, html='png', pdf='pdf')
knitrSet(lang='markdown', w=7, h=7, dev=fdev, fig.path=paste0(fdev, '/sim2-'))
options(prType=markup)
## If producing html, ggplot2 graphics are converted to plotly graphics
## so that hovertext will show extra information
## Assumes aes(..., label=txt) used
ggp <- if(outfmt != 'html')
  function(ggobject, ...) ggobject else Hmisc::ggplotlyr

# When outputting to pdf, make LaTeX place figures right at the point
# in which they are created; requires float.sty (see yaml header)
if(outfmt == 'pdf') knitr::opts_chunk$set(fig.pos = 'H', out.extra='')

# Function in Hmisc that will sense chunk has results='asis' and
# print in Rmarkdown verbatim mode if needed
pr <- markupSpecs$markdown$pr

# Define the squote, equote, pr, findstart, sop12, and dosim functions
source('funs.r')

# Read the source code for the hashCheck function from
# https://github.com/harrelfe/rscripts/blob/master/hashCheck.r
# This makes it easy to see if any objects changed that require re-running
# a simulation, and reports on any changes
getRs('hashCheck.r', put='source')

# Load runParallel function from github
# Makes parallel package easier to use and does recombination
# over batches
getRs('runParallel.r', put='source')

```

```

# Quick way to get the variance-covariance matrix from a vgam fit, per author Thomas Yee
# See ~/r/VGAM/notes.md
vgamcov <- function(fit) {
  covun <- chol2inv(fit@R)
  cnames <- names(fit@coefficients)
  dimnames(covun) <- list(cnames, cnames)
  covun
}

```

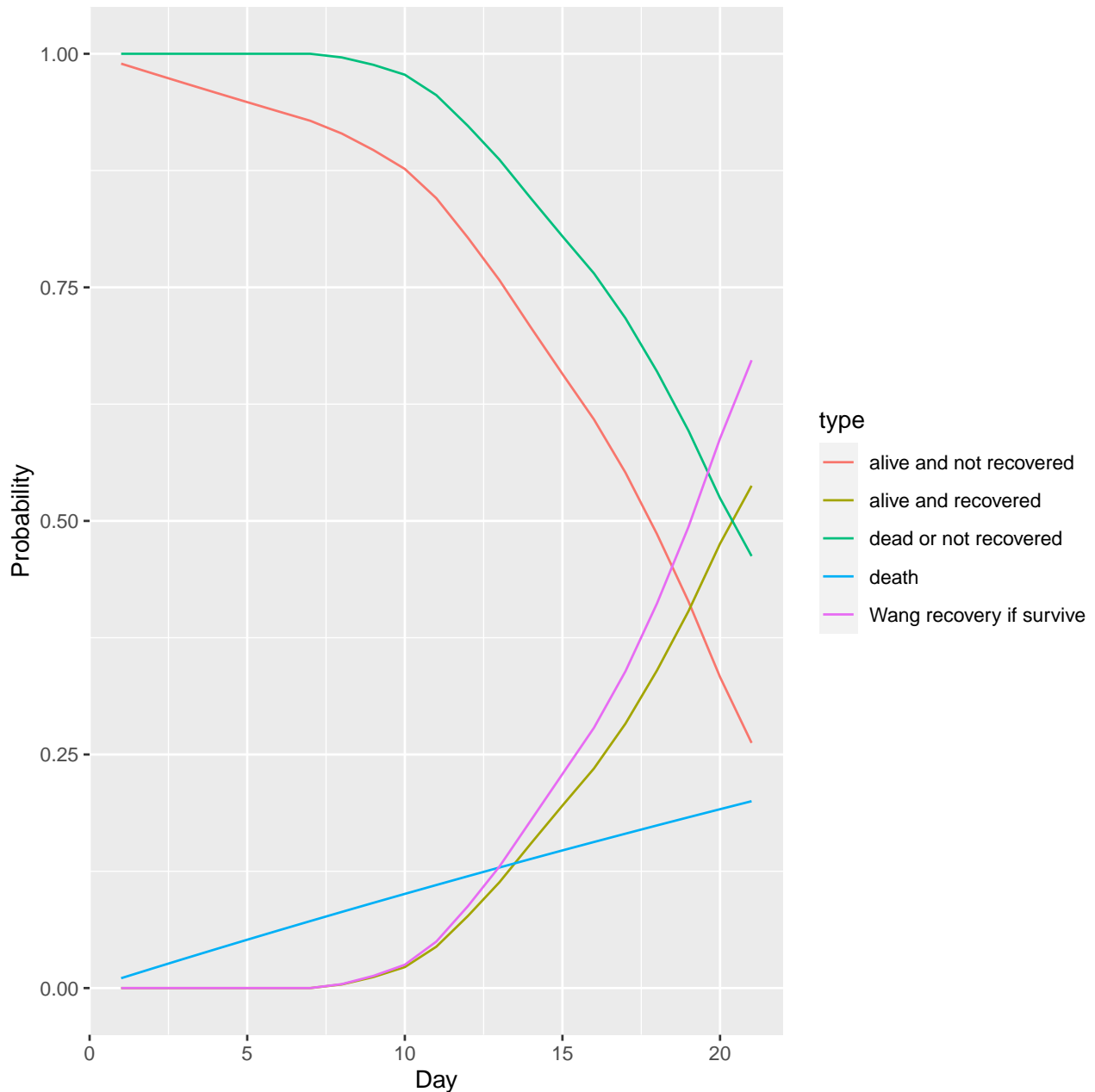
Simulation Model

To be able to simulate serial 3-level ordinal outcomes ($Y=0,1,2$ above) it is necessary to have estimates of control arm daily outcome probabilities. Our estimates come from the corticosteroid arm of the trial by Wang *et al*, Figure 3 Panel D. The cumulative incidence estimates of complete re-epithelialization there correspond to survivors. The new trial's follow-up period for the primary $Y=0,1,2$ endpoint is 21 days. We assume that the 21d mortality for the expected severely ill patient population being studied to be 0.2, and assume an exponential distribution for the time until death such that $S(21) = 0.8$, leading to the cumulative incidence of death by day t of $1 - \exp(-\lambda t)$ where $\lambda = \frac{-\log(0.8)}{21} = 0.010625$. We smooth the recovery probabilities of Wang *et al* and interpolate the estimates to 1, 2, ..., 21 days, then scale each day's estimate by the probability of the event upon which they are conditioned (being alive).

```

d <- read.csv('wangFig3D.csv', col.names=c('t', 'p'), header=FALSE)
d$p <- d$p / 100
s <- with(d, supsmu(t, p))
w <- approx(s, xout=1:21)
t <- w$x
p <- w$y
lambda <- -log(0.8) / 21
pdeath <- 1 - exp(- lambda * (1 : 21))
precoveredAlive <- p * (1 - pdeath)
pnotrecoveredAlive <- 1 - pdeath - precoveredAlive
type <- rep(c('death', 'alive and not recovered', 'alive and recovered', 'dead or not recovered',
             'Wang recovery if survive'), each=21)
y <- c(pdeath, pnotrecoveredAlive, precoveredAlive, 1 - precoveredAlive, p)
g <- ggplot(data.frame(Day=1:21, Probability=y), aes(x=Day, y=Probability, color=type)) + geom_line()
ggp(g)

```



Compute parameters of a Markov ordinal model that provide the best compromise fit the the SOPs for death and for alive not recovered, over 21d. Since there are only 3 states and two of them are absorbing, the previous state is always $Y=1$ so previous state is ignored in the transition functions g_1 and g_2 . g_1 and g_2 are the same for control patients. g_1 parameterizes a constant treatment effect over time, and g_2 parameterizes a linearly increasing effect such that there is no effect at day 1 and the maximum effect is reached at day 10. The `parameter` for g_1 is the log odds transition ratio for treatment, and for g_2 is the log odds ratio for treatment at 10d.

```
# Define function that computes logit of transition probabilities
# Allow for non-PO with regard to a quadratic time effect

g1 <- function(yprev, t, gap, X, parameter=0, extra) {
  if(! all(yprev == 1)) {prn(yprev); stop('program logic error')}
  kappa <- extra[1:4]
```

```

lp <- matrix(0.,
             nrow=length(yprev), ncol=2,
             dimnames=list(as.character(yprev), c('1', '2')))
# 2 columns = no. distinct y less 1 = length of intercepts
# lp[yp, ] is a 3-vector because the kappa components are split out into a 3-vector
# kappa[1:2] are the PO linear and quadratic time trend parameters and
# kappa[3:4] are the non-PO extra values for death
for(yp in yprev)
  lp[as.character(yp), ] <- kappa[1] * t + kappa[2] * t ^ 2 +
    cbind(0, 1) * (kappa[3] * t + kappa[4] * t ^ 2) +
    parameter * (X == 2)
lp
}

g2 <- function(yprev, t, gap, X, parameter=0, extra) {
  if(! all(yprev == 1)) {prn(yprev); stop('program logic error')}
  kappa <- extra[1:4]
  lp <- matrix(0.,
               nrow=length(yprev), ncol=2,
               dimnames=list(as.character(yprev), c('1', '2')))
  # 2 columns = no. distinct y less 1 = length of intercepts
  # lp[yp, ] is a 3-vector because the kappa components are split out into a 3-vector
  # kappa[1:2] are the PO linear and quadratic time trend parameters and
  # kappa[3:4] are the non-PO extra values for death
  for(yp in yprev)
    lp[as.character(yp), ] <- kappa[1] * t + kappa[2] * t ^ 2 +
      cbind(0, 1) * (kappa[3] * t + kappa[4] * t ^ 2) +
      parameter * (X == 2) * pmin(t - 1, 9) / 9
  lp
}

# SOP targets for control patients:
target <- cbind(precoveredAlive, pnotrecoveredAlive, pdeath)
rownames(target) <- as.character(1:21)

z <- findstart(intercepts=c(9.7, -4.5),
               extra=c(-.67, 0.014, .64, -.011), initial=1, absorb=c(0, 2),
               times=1:21, y=0:2, g=g1, seed=12)

```

Minimum sum of absolute errors: 0.3088099

Extra achieving this minimum:

```
[1] -0.37658356 -0.39228213 -0.03521566 -0.44798181
```

Iterations: 44

Sum of absolute errors: 0.3088099

Intercepts: 9.574 -4.503

Extra parameters:

```
[1] -0.6542 0.0131 0.6268 -0.0104
```

```
sop12(y=0:2, times=1:21, initial=1, absorb=c(0,2),
      intercepts=z$intercepts, g=g1, extra=z$extra, which=1)
```

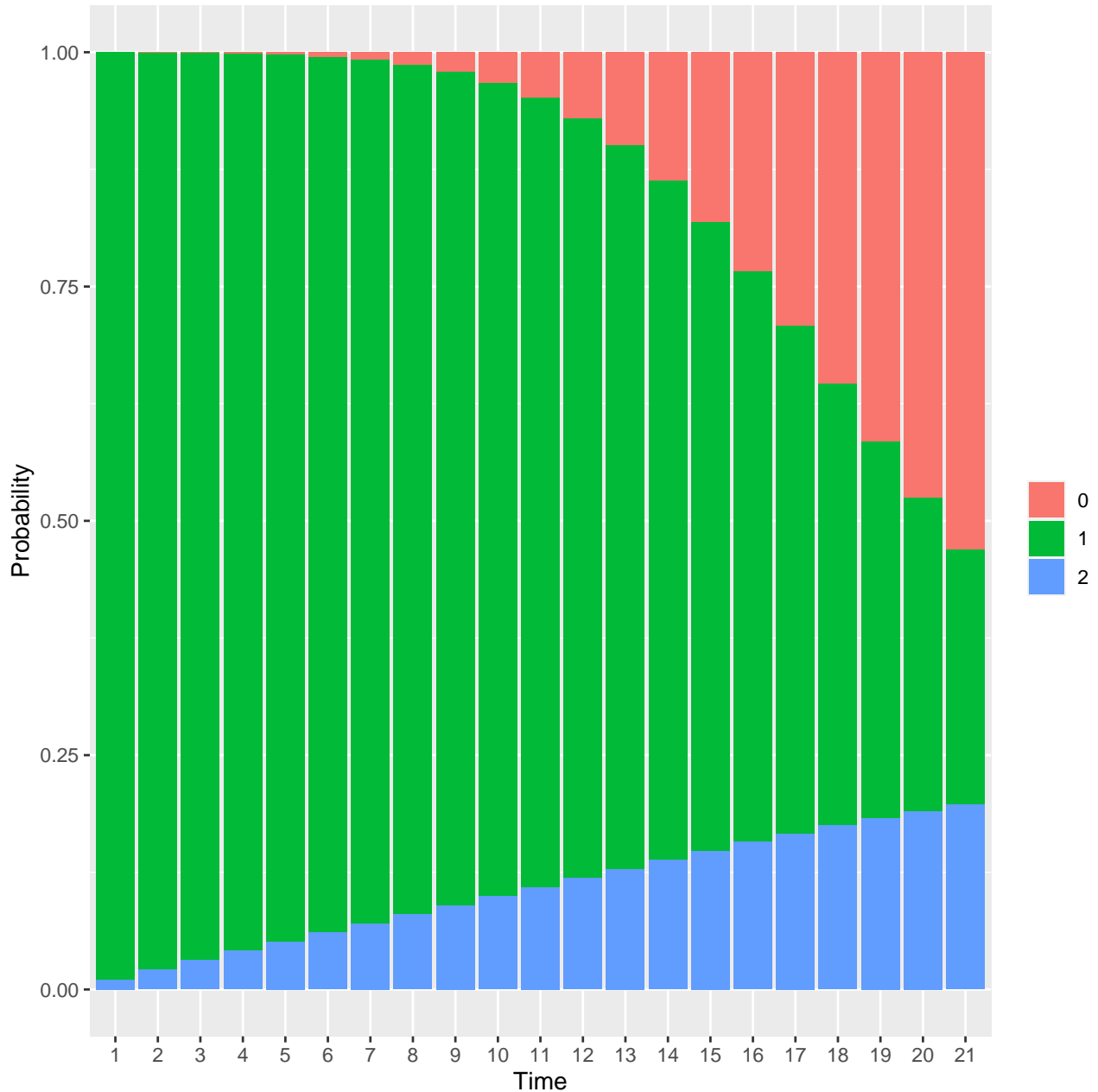
Occupancy probabilities for group 1:

	0	1	2
1	0.000	0.989	0.011
2	0.000	0.979	0.021
3	0.001	0.968	0.031
4	0.002	0.957	0.041
5	0.003	0.946	0.051
6	0.005	0.935	0.061
7	0.008	0.922	0.070
8	0.013	0.907	0.080
9	0.021	0.889	0.090
10	0.033	0.868	0.099
11	0.049	0.842	0.109
12	0.071	0.811	0.119
13	0.100	0.772	0.129
14	0.137	0.725	0.138
15	0.182	0.670	0.148
16	0.234	0.609	0.157
17	0.292	0.541	0.166
18	0.354	0.471	0.175
19	0.416	0.401	0.183
20	0.476	0.334	0.190
21	0.530	0.272	0.197

```
# Save the optimal values of the extra vector as default values for the extra argument  
# so we don't need to specify them in later steps
```

```
formals(g1)$extra <- z$extra  
formals(g2)$extra <- z$extra  
saveRDS(list(g1=g1, g2=g2), 'gsim2.rds')
```

```
# Save intercepts  
ints <- z$intercepts
```



Parameter values were found that made the Markov PO model accurately reflect the existing data, with a total absolute error of 0.3 over 63 state occupancy probabilities. Note that a strong non-proportional odds component for the absolute follow-up time effect was necessary to reflect the changing relative frequencies of the three outcomes over time.

Transition Odds Ratios vs. Change in Mean Time Unrecovered

The primary treatment effect in a Markov PO model is stated in terms of the transition odds ratio (OR), i.e., the ratio of odds of $Y(t) \geq y | Y(t-1) = a$ for treatment B vs. treatment A (with only one non-absorbing state the $Y(t-1) = a$ condition is superfluous). The primary result will be stated in terms of the B - A difference in mean time unhealed (which includes time not alive). We need to be able to relate one to the other. Two simulations are done. In the first, the treatment effect is constant over time in the simulated data and in the model used to analyze them. In the second simulation we allow in the simulations and in the final

analysis for a treatment \times time interaction. For these simulations we define the OR $\exp \beta$ to be the treatment effect at 10d, assume the OR stays there after 10d, and assume that the log odds ratio is linear in time from $t = 1$ to $t = 10$ with the OR being 1.0 at $t = 1$ (no treatment effect on the first day post randomization). So define the treatment effect on day t on the log odds scale to be $\frac{\min(t-1,9)}{9}\beta$. But for the fitted model for these second simulations we use a quadratic time \times treatment interaction to reflect imperfect knowledge about a time-varying treatment effect.

Vary β and compute the difference and ratio of mean days not recovered separately for the two true treatment effect models.

```
ors <- seq(0.1, 1, by=0.05)
m <- length(ors)
for(wg in 1 : 2) {
  mnr <- numeric(m)
  i <- 0
  for(or in ors) {
    sop <- soprobMarkovOrd(0:2, 1:21, initial=1, absorb=c(0,2),
                          intercepts=ints, g=if(wg == 1) g1 else g2,
                          X=2, parameter=log(or))

    i <- i + 1
    mnr[i] <- sum(1. - sop[, 1]) # mean time not recovered
  }
  rn <- function(x) if(max(x) <= 1) round(x, 2) else round(x, 1)
  w <- data.frame(OR=ors, `Mean Days Unrecovered`=rn(mnr),
                 `Difference in Days`=rn(mnr[m] - mnr),
                 Ratio=rn(mnr / mnr[m]), check.names=FALSE)
  cap <- if(wg == 1) 'Constant treatment effect'
         else 'Changing treatment effect'
  print(knitr::kable(w, digits=3, caption=cap))
  if(wg == 1) mnr1 <- mnr else mnr2 <- mnr
}
```

Table 2: Constant treatment effect

OR	Mean Days Unrecovered	Difference in Days	Ratio
0.10	11.2	6.9	0.62
0.15	12.3	5.7	0.68
0.20	13.2	4.9	0.73
0.25	13.9	4.2	0.77
0.30	14.4	3.6	0.80
0.35	14.9	3.1	0.83
0.40	15.4	2.7	0.85
0.45	15.7	2.3	0.87
0.50	16.1	2.0	0.89
0.55	16.4	1.7	0.91
0.60	16.6	1.4	0.92
0.65	16.9	1.2	0.93
0.70	17.1	1.0	0.95
0.75	17.3	0.8	0.96
0.80	17.5	0.6	0.97
0.85	17.6	0.4	0.98
0.90	17.8	0.3	0.98
0.95	17.9	0.1	0.99
1.00	18.1	0.0	1.00

Table 3: Changing treatment effect

OR	Mean Days Unrecovered	Difference in Days	Ratio
0.10	12.0	6.0	0.67
0.15	13.0	5.1	0.72
0.20	13.7	4.4	0.76
0.25	14.3	3.8	0.79
0.30	14.8	3.3	0.82
0.35	15.2	2.8	0.84
0.40	15.6	2.5	0.86
0.45	16.0	2.1	0.88
0.50	16.3	1.8	0.90
0.55	16.5	1.6	0.91
0.60	16.8	1.3	0.93
0.65	17.0	1.1	0.94
0.70	17.2	0.9	0.95
0.75	17.4	0.7	0.96
0.80	17.5	0.5	0.97
0.85	17.7	0.4	0.98
0.90	17.8	0.3	0.99
0.95	18.0	0.1	0.99
1.00	18.1	0.0	1.00

We may focus on a transition OR=0.5 which translates to a 2d reduction in days not recovered (1.8d for time-varying treatment effect model) or ratio of mean times of 0.89 and 0.9.

Simulation of Bayesian Sequential Trial

To simulate the operating characteristics of a sequential Bayesian procedure we use a normal approximation to the data likelihood, use flat priors for all but the treatment effect, and apply a skeptical normal prior distribution for the treatment effect. Then the posterior distribution for the treatment effect is also normal. This greatly reduces computation time. The mean of the normal prior distribution for the treatment effect $\log(\text{OR})$ is zero and the standard deviation is chosen so that $P(\text{OR} < 0.25) = P(\text{OR} > 4) = 0.05$.

In addition to simulating the posterior distribution of the transition OR, the `estSeqMarkovOrd` at the last data look also simulates a Cox proportional hazards χ^2 statistic for time to recovery. Time to recovery is the first time at which $Y=0$, and if none or any $Y=2$, time to recovery is censored at 21d. The purpose of this additional simulation is to compare the frequentist power of a two-sample comparison of time until recovery with the frequentist power of the transition odds ratio at days 10-21 in the Markov proportional odds model.

The simulation model is given above and is encapsulated in the `g` functions previously defined. For `g2` the analysis model is not informed of the linearly increasing treatment effect from day 1 to day 10 that stays constant from days 10-21. Instead, the analysis model used on each simulated dataset has a quadratic time \times treatment interaction.

Simulate a Single Large Trial

Let's simulate a single 10000-patient trial to check that we can recover a true log odds ratio of $\log(0.5) = -0.69$ overall (model 1) or at $t = 10$ (model 2).

```
require(VGAM)

n <- 10000
for(wg in 1 : 2) {
```

```

cat(if(wg == 1) 'Constant treatment effect\n\n' else '\nTime-varying treatment effect\n\n')
set.seed(8)
g <- if(wg == 1) g1 else g2
s1 <- simMarkovOrd(n=n / 2, y=0:2, 1:21, initial=1, X=c(group=1),
                  absorb=c(0, 2), intercepts=ints, g=g, parameter=0)
s2 <- simMarkovOrd(n=n / 2, y=0:2, 1:21, initial=1, X=c(group=2),
                  absorb=c(0, 2), intercepts=ints, g=g, parameter=log(0.5))

s2$id <- s2$id + n
s <- rbind(s1, s2)
s$yprev <- as.factor(s$yprev)
s$group <- as.factor(s$group)
form <- y ~ pol(time, 2) + group
if(wg == 2) form <- y ~ pol(time, 2) * group
f <- vglm(form, cumulative(parallel=FALSE ~ pol(time, 2), reverse=TRUE), data=s)
k <- coef(f)
# Compute treatment effect; is a constant for model 1
if(wg == 1) cat('beta =', round(k['group2'], 3), '\n')
else {
  # Plot estimated treatment effect over time
  time <- 1 : 21
  effect <- k[7] + time * k[8] + time * time * k[9]
  plot(time, effect, type='l')
  lines(c(1, 10, 21), c(0, log(0.5), log(0.5)), col=gray(0.8))
  # Repeat, doing this another way
  contrvgam <- function(fit) { #, n1=NULL, n2=NULL, parameter=NULL) { # for customized vgam contra.
    beta <- coef(fit)
    i <- c('group2', 'pol(time, 2)time:group2', 'pol(time, 2)time^2:group2')
    b <- beta[i]
    effect <- b[1] + 10 * b[2] + 100 * b[3]
    # Treatment effect at 10 days (quadratic time x treatment interaction)
    # if(length(n1)) return(list(Contrast=effect, SE=approxse(or=exp(parameter), n1=n1, n2=n2)))
    v <- vgamcov(fit)
    if(any(i %nin% rownames(v))) {saveRDS(fit, 'problemfit.rds'); stop()}
    v <- v[i, i]
    vr <- v[1, 1] + 100 * v[2, 2] + 10000 * v[3, 3] + 20 * v[1, 2] +
          200 * v[1, 3] + 2000 * v[2, 3]
    list(Contrast=effect, SE=sqrt(vr))
  }
  print(contrvgam(f))
}
}

```

Constant treatment effect

beta = -0.704

Time-varying treatment effect

```

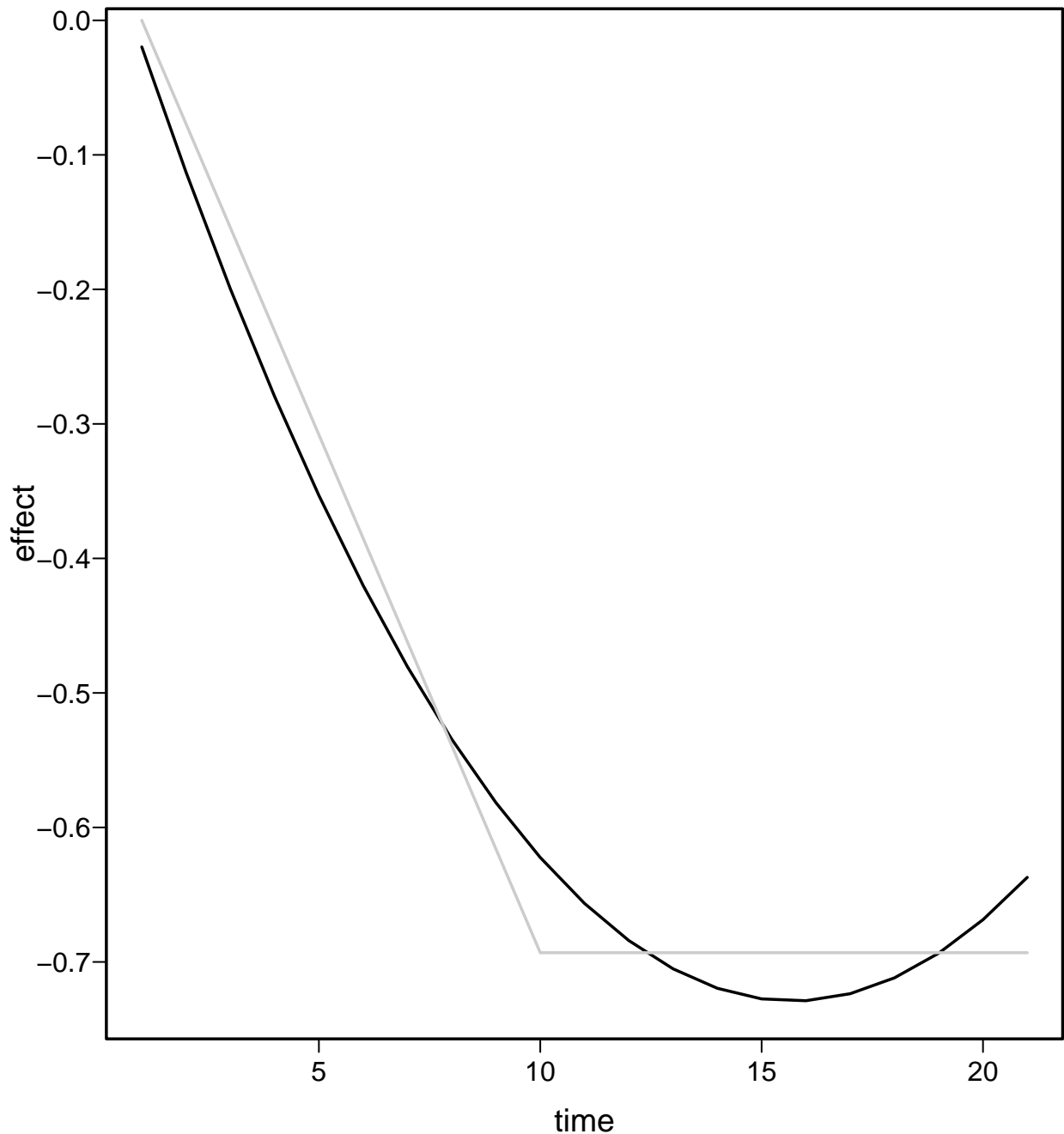
$Contrast
  group2
-0.6223537

```

```

$SE
[1] 0.03897312

```



The partial PO Markov model was able to recover the treatment effect large simulated data, as the fitted quadratic effect was not far from the true linear spline time-related treatment effect shown in gray above.

Approximating the Standard Error of the Treatment Effect

Let's see if we can satisfactorily approximate the standard error of the treatment effect. This would speed up simulations in the future. We model on the scale of the log of the $\hat{\beta}$ standard error. Do this separately for model 1 (one treatment parameter) and model 2 (three treatment parameters).

```
# Try seeing how predictable is the standard error of the main contrast, for varying true effects and s
donev <- TRUE
set.seed(1)
ors <- seq(0.1, 1, by=0.1)
```

```

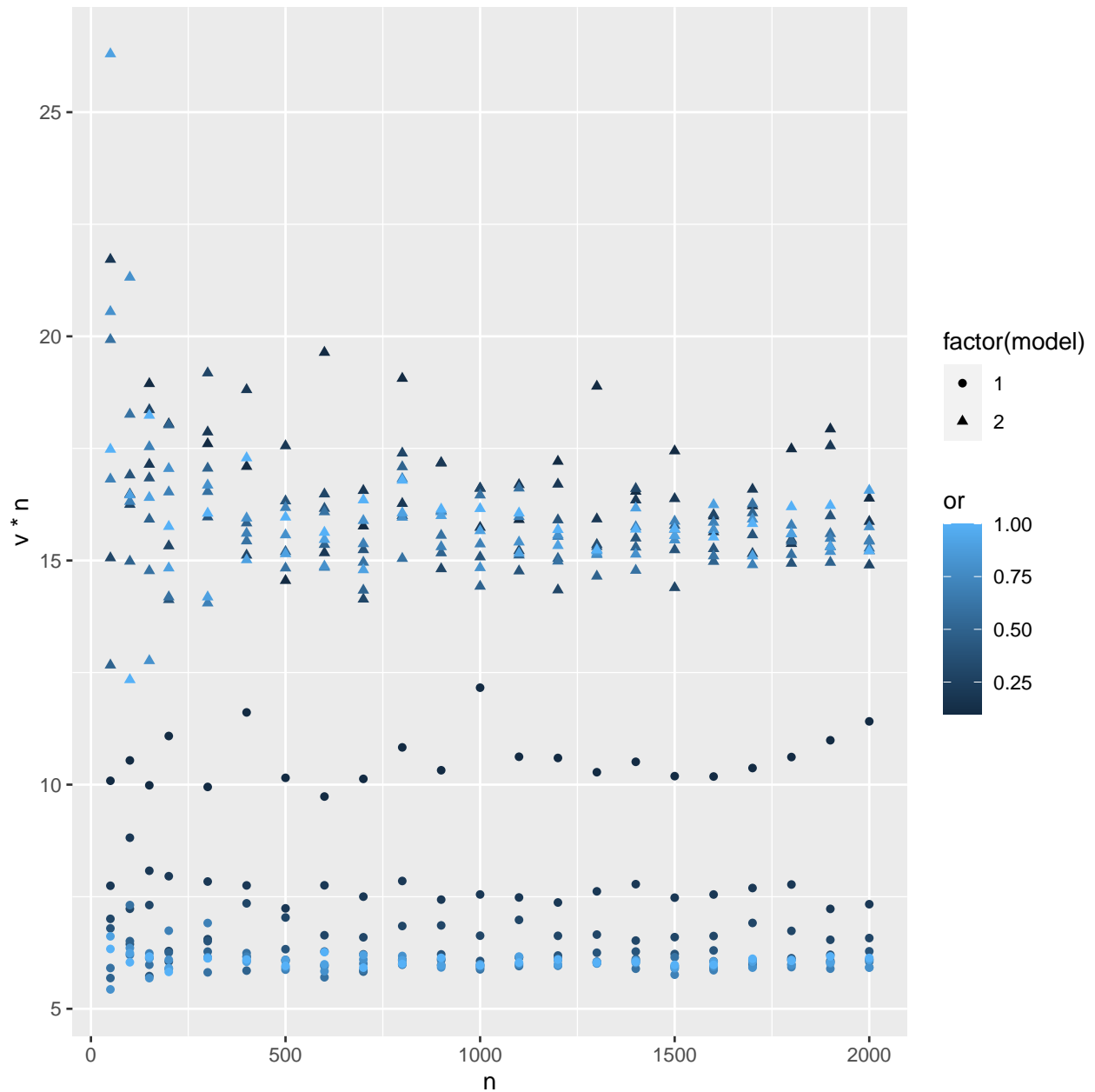
if(donev) w <- readRDS('vn.rds') else {
  w <- expand.grid(or=ors, n=c(50, 100, 150, 200, seq(300, 2000, by=100)), model=1:2)
  v <- numeric(nrow(w))
  for(i in 1 : nrow(w)) {
    n <- w[i, 'n']
    parameter <- log(w[i, 'or'])
    m <- w[i, 'model']
    g <- if(m == 1) g1 else g2
    s1 <- simMarkovOrd(n=n / 2, y=0:2, 1:21, initial=1, X=c(group=1),
                      absorb=c(0, 2), intercepts=ints, g=g, parameter=0)
    s2 <- simMarkovOrd(n=n / 2, y=0:2, 1:21, initial=1, X=c(group=2),
                      absorb=c(0, 2), intercepts=ints, g=g, parameter=parameter)

    s <- rbind(s1, s2)
    s$group <- as.factor(s$group)
    form <- y ~ pol(time, 2) + group
    if(m == 2) form <- y ~ pol(time, 2) * group
    f <- vgam(form, cumulative(parallel=FALSE ~ pol(time, 2), reverse=TRUE), data=s)
    v[i] <- if(m == 1) vcov(f)['group2', 'group2'] else contrvgam(f)$SE ^ 2
  }
  w$v <- v
  saveRDS(w, 'vn.rds')
}
ratio <- subset(w, model == 2)$v / subset(w, model == 1)$v
pr('Median model 2 : model 1 ratio of variances of treatment effect', inline=median(ratio))

Median model 2 : model 1 ratio of variances of treatment effect 2.521503

w <- subset(w, v < 1) # remove outliers from vgam problems
v <- w$v
ggplot(w, aes(x=n, y=v * n, shape=factor(model), color=or)) + geom_point()

```



Judging by the median variance ratio and the graph above, the 3-parameter treatment contrast is less than half as efficient as the 1-parameter contrast assuming constant treatment effect. Judging by the flatness of the above trends in the variance of $\hat{\beta}$ multiplied by the sample size, the dominant form of the variance is $\frac{k}{n}$.

```
# Fit various models to log(v)
h <- function(form) {
  z <- NULL
  for(type in c('ols', 'robust')) {
    h <- switch(type,
      robust = MASS::rlm(form, data=dat),
      ols     = ols(form, data=dat))
    r <- resid(h)
    ssr <- var(fitted(h))
    r2 <- ssr / (ssr + var(r))
  }
}
```

```

z <- rbind(z, data.frame(type=type, model=as.character(form)[3],
                        stat=c('Mean', 'Median', 'P(|error| > 0.1)'),
                        error=c(mean(abs(r)), Median=median(abs(r)),
                                mean(abs(r) > 0.1)),
                        r2=r2,
                        check.names=FALSE))
}
z
}

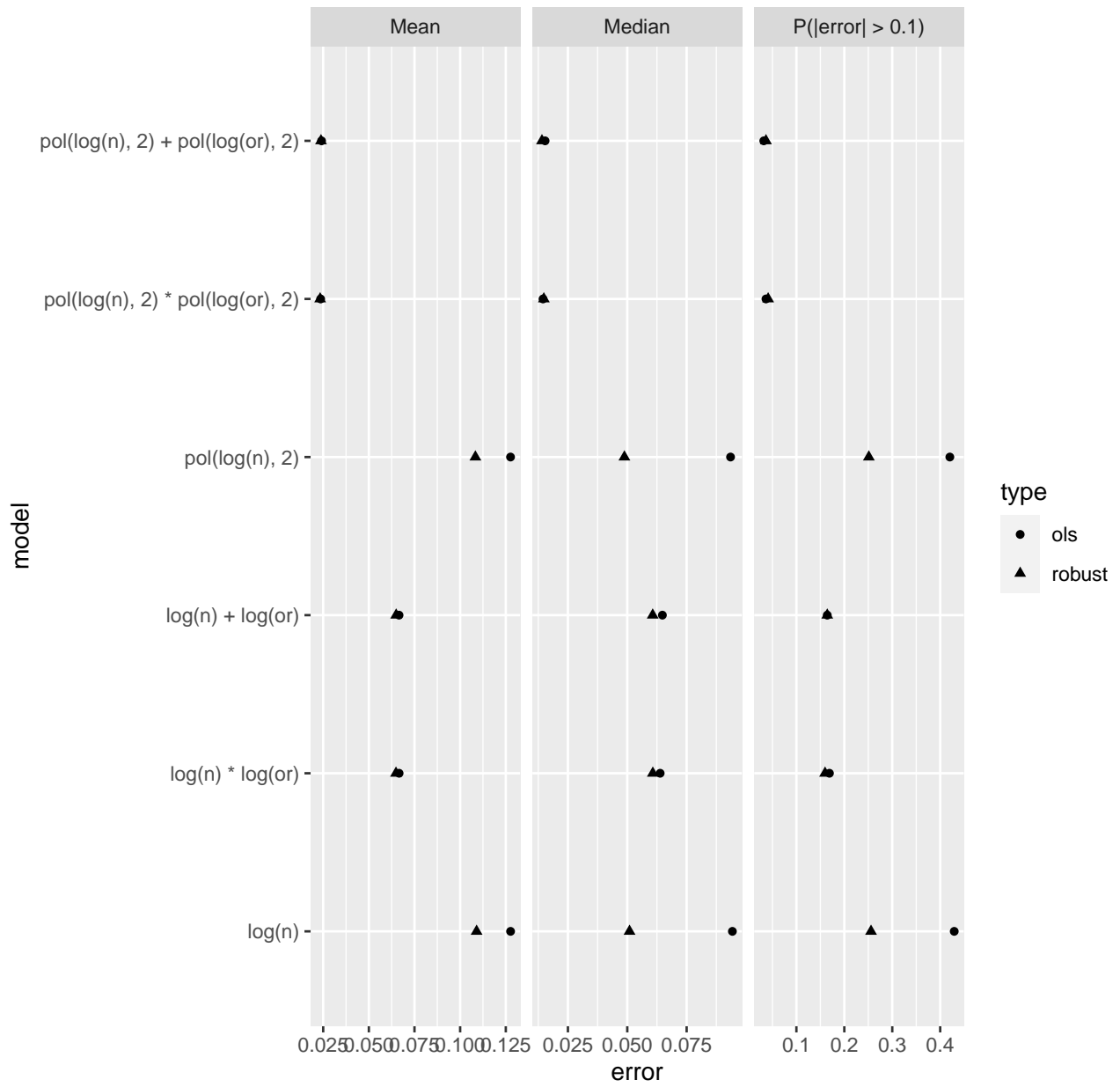
for(wg in 1 : 2) {
  cat('\nModel', wg, '\n\n')
  dat <- subset(w, model == wg)
  z <- rbind(
    h(log(v) ~ log(n)),
    h(log(v) ~ pol(log(n), 2)),
    h(log(v) ~ log(n) + log(or)),
    h(log(v) ~ log(n) * log(or)),
    h(log(v) ~ pol(log(n), 2) + pol(log(or), 2)),
    h(log(v) ~ pol(log(n), 2) * pol(log(or), 2)) )

  print(ggplot(z, aes(x=error, y=model, shape=type)) + geom_point() +
        facet_wrap(~ stat, scales='free_x') + labs(caption=paste('Model', wg)))
}

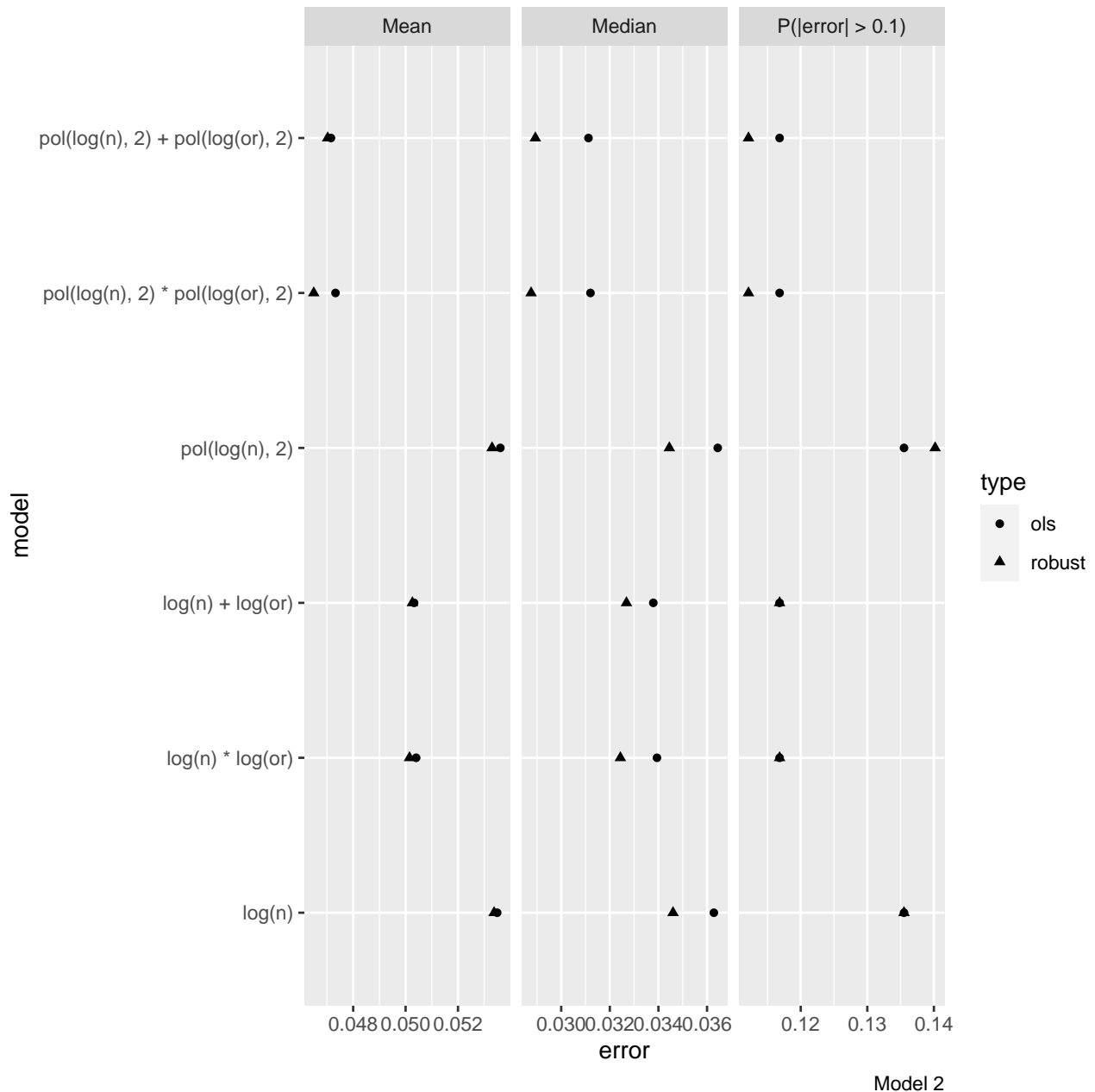
```

Model 1

Model 2



Model 1



The full models (two quadratics) without interaction estimated using robust regression best predicts the log of the standard error of the treatment log odds ratio. These model have $R^2 > 0.99$ and less than 0.12 chance of making an error larger than 0.1 on the log scale (which is a fold change error of 0.9). The median absolute log error is < 0.03 (fold change error of 0.97 in estimating the variance). Write a function for using these approximations later.

```
for(wg in 1 : 2) {
  dat <- subset(w, model == wg)
  approxvar <- MASS::rlm(log(v) ~ pol(log(n), 2) + pol(log(or), 2), data=dat)

  # Derive a function to compute standard error from n and or
  b <- coef(approxvar)
  if(wg == 1) beta.approxvar1 <- b else beta.approxvar2 <- b
  approxse <- function(n, or, n1=NULL, n2=NULL) {
```

```

b <- beta.approxvar
if(length(n1)) n <- 4. / ((1. / n1) + (1. / n2))
x <- log(n)
y <- log(or)
xb <- b[1] + b[2] * x + b[3] * x ^ 2 + b[4] * y + b[5] * y ^ 2
sqrt(exp(xb))
}
}

```

Simulation

For each true OR we simulate 1000 clinical trials. ORs will vary from 0.2, 0.3, ..., 1.0. Markov ordinal analysis is done after 74, 108, 142, and the 178 patients have completed follow-up. Frequentist power is computed at $\alpha = 0.05$.

For each trial the treatments are randomly assigned with probability 1/2 each. The high-level `estSeqMarkovOrd` function calls the `simMarkovOrd` function to simulate each trial. Because the contrast of interest needs to take into account a treatment \times time interaction, the `groupContrast` argument must be specified below.

Note: `dosim` calls `estSeqMarkovOrd` which fits data using the `rms` package `lrm` function when the model is a proportional odds model. Since there is non-proportional odds, the `VGAM` package's `vgam` or `vglm` function is used to fit the partial proportional odds model instead. When doing Bayesian analysis, the `rmsb` package `blrm` function will fit a constrained partial PO model.

Partial Proportional Odds Model Simulation

```

ppo <- ~ pol(time, 2) # partial PO model with PO exception for time
ors <- c(0.3, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.8, 1.0)

# Define time to recovery with treatment of death at any time as censored at 21d
tc <- function(y) {
  if(any(y == 2)) return(c(21, 0)) # death
  if(any(y == 0)) return(c(min(which(y == 0)), 1)) # which works since time=1,2,3,...
  c(length(y), 0) # censored; length(y) should be 21
}

dosimit <- function(model, g, formula, file)
  dosim(y=0:2, times=1:21, g=g, initial=c('1'=1.0), # one initial state, sampled w/prob 1.0
        absorb=c(0, 2),
        intercepts=ints,
        ors=ors, formula=formula,
        ppo=ppo, groupContrast=if(model == 2) contrvgam,
        timecriterion=tc, coxzph=FALSE, seed=3,
        nsim=1000, looks=c(74, 108, 142, 178),
        estcut=10, vestcut=1,
        file=file)

pr('Model 1')

```

Model 1

```
sim1 <- dosimit(1, g1, y ~ pol(time, 2) + group, file='simlooks1.rds')
```

Median of 1000 logistic regression coefficients for each OR

	(Intercept):1	(Intercept):2	pol(time, 2)time:1	pol(time, 2)time:2
0.3	9.6519	-4.6701	-0.6601	-0.0039
0.4	9.7387	-4.5732	-0.6784	-0.0003
0.5	9.7268	-4.6103	-0.6719	-0.0023
0.55	9.6718	-4.5580	-0.6682	-0.0099
0.6	9.7693	-4.6049	-0.6777	-0.0081
0.65	9.6578	-4.5740	-0.6644	-0.0073
0.7	9.7861	-4.5306	-0.6811	-0.0175
0.8	9.7074	-4.5441	-0.6710	-0.0163
1	9.8316	-4.5846	-0.6900	-0.0093

	pol(time, 2)time^2:1	pol(time, 2)time^2:2	group2
0.3	0.0135	0.0016	-1.2056
0.4	0.0139	0.0013	-0.9346
0.5	0.0139	0.0016	-0.7088
0.55	0.0135	0.0019	-0.6073
0.6	0.0140	0.0017	-0.5155
0.65	0.0134	0.0016	-0.4226
0.7	0.0141	0.0021	-0.3762
0.8	0.0136	0.0021	-0.2290
1	0.0143	0.0019	0.0124

Standard deviation of simulated group effects, scaled MAD, and square root of median of estimated variance

	OR	look	SD	MAD	SDest
1:	0.30	74	1.834008e+11	0.340	0.307
2:	0.30	108	1.342390e+11	0.282	0.251
3:	0.30	142	2.438357e+09	0.215	0.218
4:	0.30	178	1.910101e+11	0.196	0.194
5:	0.40	74	4.981205e+10	0.326	0.296
6:	0.40	108	8.321299e+10	0.236	0.243
7:	0.40	142	3.321830e+10	0.212	0.211
8:	0.40	178	1.139360e+08	0.181	0.189
9:	0.50	74	9.300620e+09	0.302	0.291
10:	0.50	108	6.178306e+06	0.240	0.240
11:	0.50	142	4.255509e+09	0.212	0.208
12:	0.50	178	1.203888e+11	0.183	0.186
13:	0.55	74	3.325708e+10	0.320	0.289
14:	0.55	108	1.321638e+10	0.247	0.238
15:	0.55	142	7.450744e+09	0.208	0.207
16:	0.55	178	4.842122e+07	0.189	0.185
17:	0.60	74	4.799702e+10	0.314	0.289
18:	0.60	108	3.435603e+11	0.246	0.238
19:	0.60	142	1.124638e+10	0.210	0.207
20:	0.60	178	4.288924e+07	0.184	0.184
21:	0.65	74	5.894516e+10	0.296	0.288
22:	0.65	108	1.158864e+09	0.229	0.237
23:	0.65	142	2.151767e+07	0.193	0.206
24:	0.65	178	1.858170e+10	0.185	0.184
25:	0.70	74	4.057616e+09	0.323	0.289
26:	0.70	108	3.231593e+10	0.245	0.238
27:	0.70	142	3.964475e+10	0.210	0.206
28:	0.70	178	1.813434e+08	0.180	0.184
29:	0.80	74	9.389267e+09	0.310	0.289

```

30: 0.80 108 4.765086e+08 0.241 0.238
31: 0.80 142 7.970000e-01 0.214 0.207
32: 0.80 178 1.910000e-01 0.188 0.185
33: 1.00 74 8.403755e+09 0.291 0.291
34: 1.00 108 2.584757e+07 0.241 0.239
35: 1.00 142 1.803306e+04 0.198 0.208
36: 1.00 178 1.880000e-01 0.187 0.186
      OR look          SD    MAD SDest

```

Standard deviation of simulated group effects and square root of median of estimated variances after re

```

      OR look    SD    MAD SDest
1: 0.30 74 0.317 0.303 0.304
2: 0.30 108 0.266 0.266 0.250
3: 0.30 142 0.220 0.211 0.218
4: 0.30 178 0.194 0.193 0.194
5: 0.40 74 0.312 0.300 0.296
6: 0.40 108 0.236 0.227 0.243
7: 0.40 142 0.207 0.208 0.211
8: 0.40 178 0.180 0.180 0.188
9: 0.50 74 0.298 0.290 0.291
10: 0.50 108 0.247 0.235 0.240
11: 0.50 142 0.212 0.210 0.208
12: 0.50 178 0.187 0.182 0.186
13: 0.55 74 0.303 0.310 0.289
14: 0.55 108 0.241 0.244 0.238
15: 0.55 142 0.214 0.207 0.207
16: 0.55 178 0.186 0.188 0.185
17: 0.60 74 0.291 0.298 0.289
18: 0.60 108 0.237 0.242 0.238
19: 0.60 142 0.209 0.208 0.207
20: 0.60 178 0.181 0.183 0.184
21: 0.65 74 0.292 0.287 0.287
22: 0.65 108 0.234 0.226 0.237
23: 0.65 142 0.204 0.192 0.206
24: 0.65 178 0.185 0.184 0.184
25: 0.70 74 0.310 0.319 0.289
26: 0.70 108 0.250 0.243 0.237
27: 0.70 142 0.210 0.208 0.206
28: 0.70 178 0.185 0.179 0.184
29: 0.80 74 0.303 0.306 0.289
30: 0.80 108 0.241 0.239 0.238
31: 0.80 142 0.215 0.214 0.207
32: 0.80 178 0.191 0.188 0.185
33: 1.00 74 0.307 0.287 0.291
34: 1.00 108 0.248 0.241 0.239
35: 1.00 142 0.211 0.198 0.208
36: 1.00 178 0.188 0.187 0.186
      OR look    SD    MAD SDest

```

```
examSim(sim1, desc='Partial PO Markov model with constant treatment effect')
```

Comparison of OR and HR

OR HR

1: 0.3 0.3
2: 0.4 0.4
3: 0.5 0.5
4: 0.6 0.5
5: 0.6 0.6
6: 0.7 0.6
7: 0.7 0.7
8: 0.8 0.8
9: 1.0 1.0

Power of Cox test at last look for time until Y=1

OR power
1: 0.30 1.00
2: 0.40 1.00
3: 0.50 0.97
4: 0.55 0.92
5: 0.60 0.81
6: 0.65 0.63
7: 0.70 0.53
8: 0.80 0.22
9: 1.00 0.05

Power of Markov proportional odds model test

OR look power
1: 0.30 74 0.99
2: 0.30 108 1.00
3: 0.30 142 1.00
4: 0.30 178 1.00
5: 0.40 74 0.92
6: 0.40 108 0.99
7: 0.40 142 1.00
8: 0.40 178 1.00
9: 0.50 74 0.70
10: 0.50 108 0.86
11: 0.50 142 0.93
12: 0.50 178 0.97
13: 0.55 74 0.57
14: 0.55 108 0.74
15: 0.55 142 0.84
16: 0.55 178 0.92
17: 0.60 74 0.45
18: 0.60 108 0.61
19: 0.60 142 0.72
20: 0.60 178 0.81
21: 0.65 74 0.30
22: 0.65 108 0.42
23: 0.65 142 0.54
24: 0.65 178 0.63
25: 0.70 74 0.27
26: 0.70 108 0.35
27: 0.70 142 0.45
28: 0.70 178 0.53
29: 0.80 74 0.13

30: 0.80 108 0.16
 31: 0.80 142 0.19
 32: 0.80 178 0.22
 33: 1.00 74 0.06
 34: 1.00 108 0.06
 35: 1.00 142 0.05
 36: 1.00 178 0.05
 OR look power

Spearman rho correlation between Markov and Cox model chi-square: 1

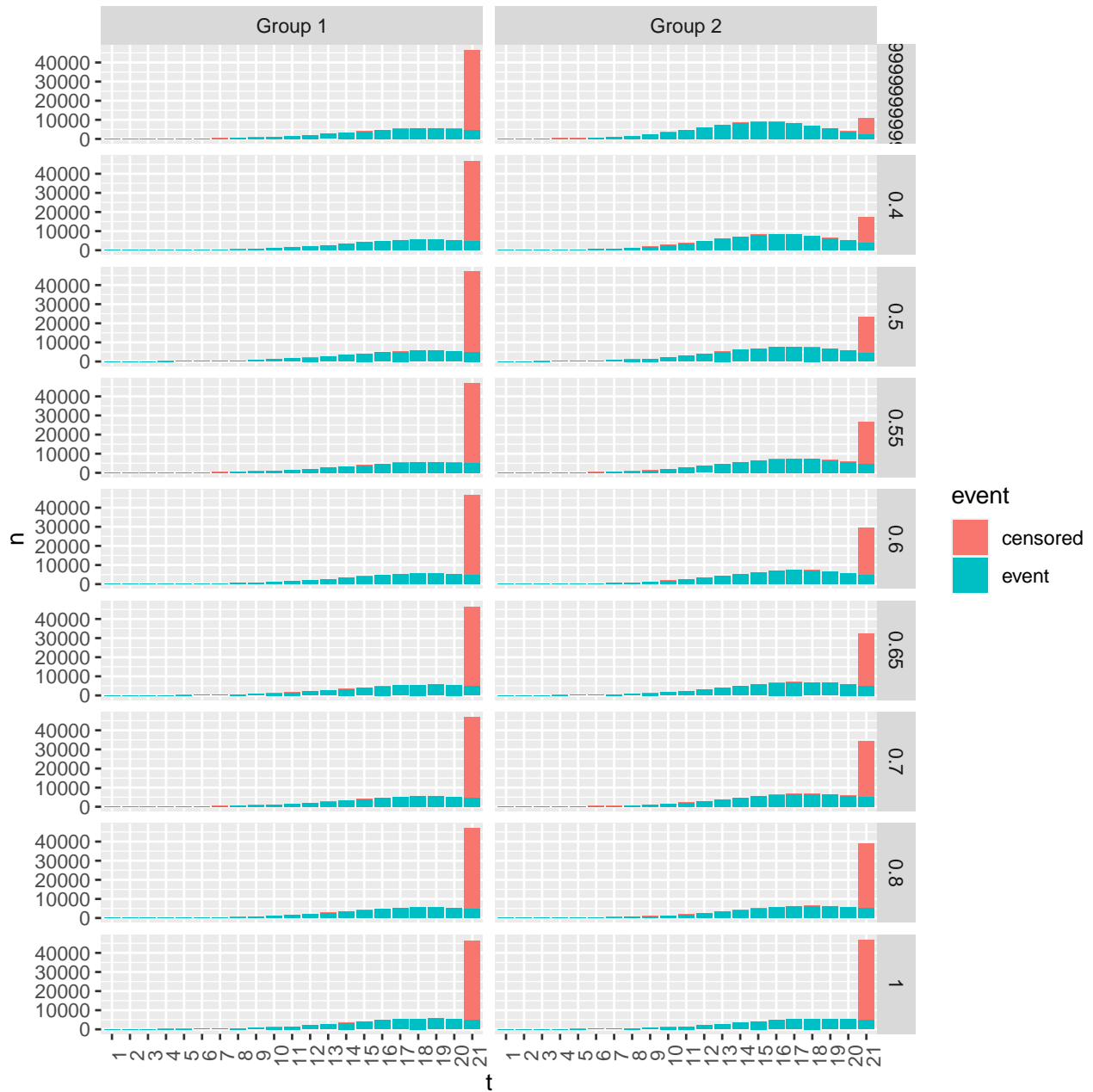


Figure 1: Time to Y=1 by group and OR, over simulations. Partial PO Markov model with constant treatment effect.

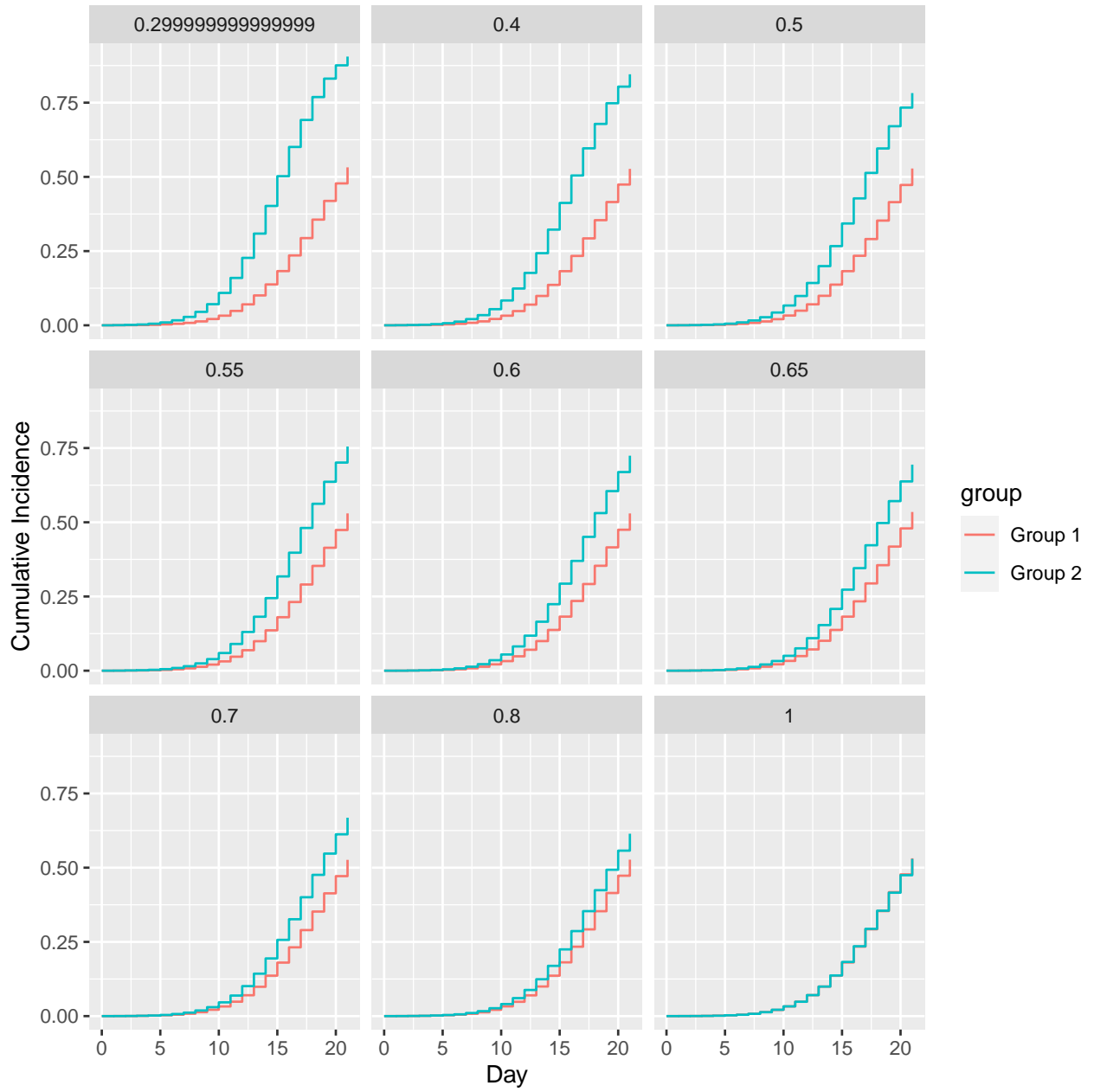


Figure 2: Cumulative incidence of $Y=1$. Partial PO Markov model with constant treatment effect.

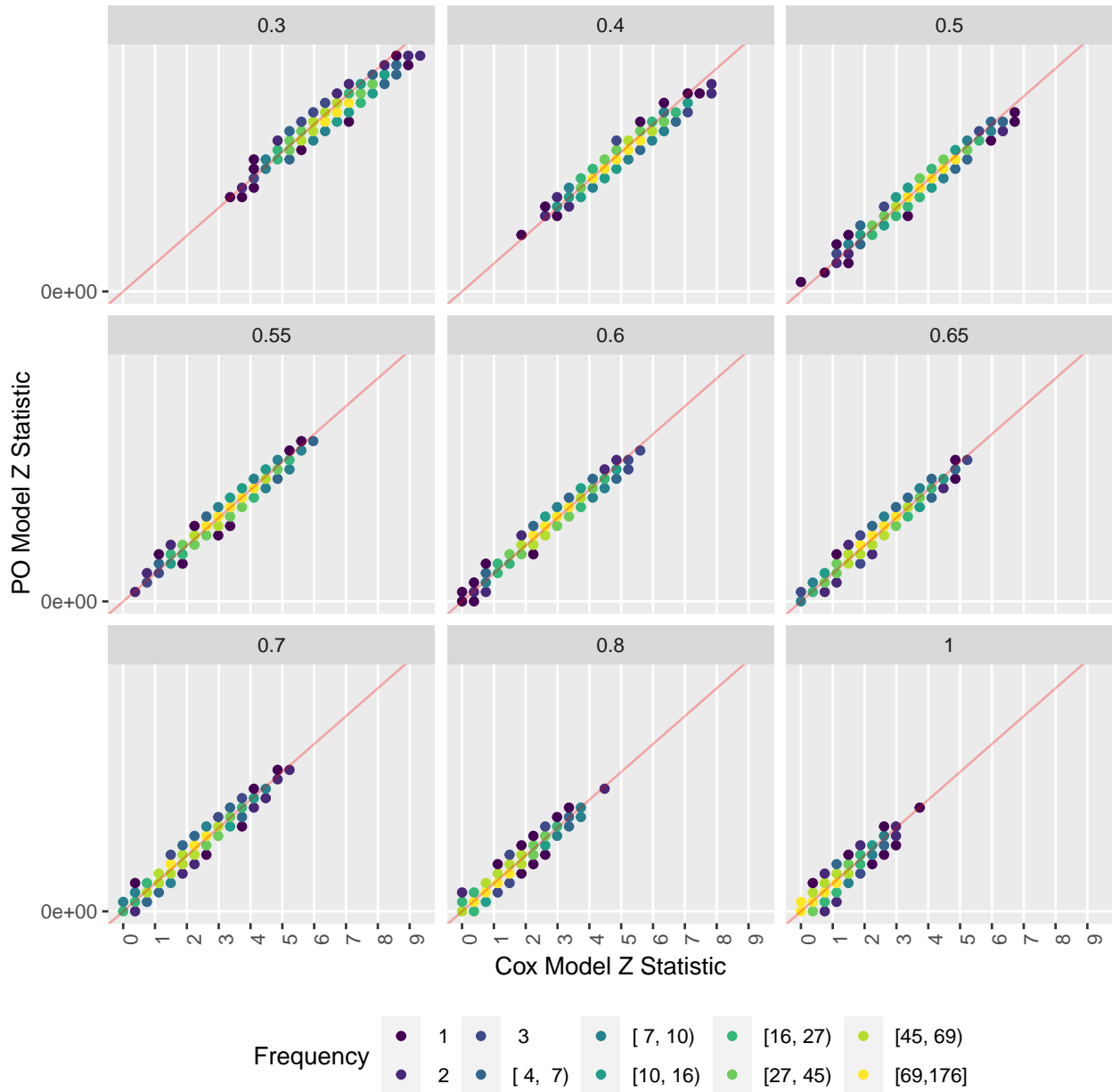


Figure 3: Scatter plot of Cox and PO model Z statistics. Partial PO Markov model with constant treatment effect.

For many of the simulations in this constant treatment effect model the `vgam` function did not converge properly, creating impossibly large maximum likelihood estimates or variance estimates. In the latter part of the output above these outliers have been removed. There you can see the extremely good accuracy of the standard error estimates from the maximum likelihood estimation process.

The transition odds ratios are equal to the hazard ratios in the time-to-recovery analyses and at the last data look the power of the transition model treatment test and the Cox proportional hazards model test are the same. The two χ^2 statistics are virtually the same.

There is approximately 0.9 power to detect An OR or HR of 0.55 and 0.8 power to detect a ratio of 0.6, both at the planned maximum sample size of 178 for two treatment arms. As shown earlier, ORs of 0.55 and 0.6 correspond to 1.7 and 1.4 days, respectively, reduction in mean time unrecovered.

Now look at model 2 in which the treatment effect modeled as a quadratic function in time but the true treatment effect is linear up until day 10, then flat. The transition model treatment contrast is made at 10d using the quadratic time \times treatment interaction.

```
pr('Model 2')
```

Model 2

```
# beta.approxvar <- beta.approxvar2
sim2 <- dosimit(2, g2, y ~ pol(time, 2) * group, file='simflooks2.rds')
```

Median of 1000 logistic regression coefficients for each OR

	(Intercept):1	(Intercept):2	pol(time, 2)time:1	pol(time, 2)time:2
0.3	9.6035	-4.4950	-0.6577	-0.0153
0.4	9.6791	-4.5851	-0.6606	-0.0005
0.5	9.6300	-4.5550	-0.6642	-0.0145
0.55	9.6749	-4.6130	-0.6616	-0.0065
0.6	9.6982	-4.5547	-0.6694	-0.0173
0.65	9.7574	-4.5826	-0.6713	-0.0133
0.7	9.7410	-4.5965	-0.6724	-0.0079
0.8	9.7657	-4.5738	-0.6819	-0.0032
1	9.7764	-4.5928	-0.6863	-0.0062

	pol(time, 2)time^2:1	pol(time, 2)time^2:2	group2	pol(time, 2)time:group2
0.3	0.0133	0.0023	0.0208	-0.1575
0.4	0.0138	0.0016	0.1136	-0.1274
0.5	0.0135	0.0018	0.1344	-0.1074
0.55	0.0134	0.0018	0.0389	-0.0858
0.6	0.0133	0.0021	0.0209	-0.0716
0.65	0.0135	0.0019	0.0234	-0.0586
0.7	0.0138	0.0016	0.0106	-0.0489
0.8	0.0137	0.0013	0.0143	-0.0331
1	0.0143	0.0015	0.0097	-0.0024

	pol(time, 2)time^2:group2
0.3	0.0050
0.4	0.0039
0.5	0.0033
0.55	0.0027
0.6	0.0024
0.65	0.0018
0.7	0.0014
0.8	0.0011
1	0.0003

Standard deviation of simulated group effects, scaled MAD, and square root of median of estimated variance

OR	look	SD	MAD	SDest	
1:	0.30	74	3.731926e+11	0.837	0.509
2:	0.30	108	2.089600e+11	0.606	0.399
3:	0.30	142	2.876523e+11	0.465	0.342
4:	0.30	178	2.670812e+11	0.390	0.303
5:	0.40	74	3.280151e+11	0.607	0.484
6:	0.40	108	6.591189e+10	0.472	0.389
7:	0.40	142	1.509852e+11	0.381	0.335
8:	0.40	178	4.525140e+10	0.311	0.298
9:	0.50	74	1.022247e+11	0.581	0.483

10:	0.50	108	8.570042e+10	0.426	0.389
11:	0.50	142	1.012767e+11	0.343	0.335
12:	0.50	178	1.702883e+10	0.321	0.297
13:	0.55	74	7.216797e+10	0.540	0.479
14:	0.55	108	4.795914e+10	0.409	0.389
15:	0.55	142	3.236497e+10	0.361	0.336
16:	0.55	178	8.530172e+09	0.323	0.299
17:	0.60	74	2.275773e+11	0.545	0.479
18:	0.60	108	1.374987e+11	0.419	0.389
19:	0.60	142	4.001238e+10	0.359	0.336
20:	0.60	178	3.992638e+09	0.309	0.298
21:	0.65	74	1.411390e+11	0.535	0.478
22:	0.65	108	9.644645e+10	0.405	0.388
23:	0.65	142	6.621740e+10	0.346	0.337
24:	0.65	178	1.031091e+10	0.287	0.300
25:	0.70	74	6.702510e+10	0.504	0.475
26:	0.70	108	2.556478e+10	0.390	0.389
27:	0.70	142	3.041722e+09	0.327	0.336
28:	0.70	178	5.884371e+10	0.288	0.299
29:	0.80	74	1.441860e+11	0.491	0.483
30:	0.80	108	8.927405e+10	0.396	0.392
31:	0.80	142	1.758400e+01	0.341	0.338
32:	0.80	178	7.426599e+04	0.320	0.300
33:	1.00	74	2.788613e+10	0.481	0.484
34:	1.00	108	2.033854e+09	0.395	0.394
35:	1.00	142	8.220139e+08	0.354	0.342
36:	1.00	178	2.353363e+06	0.309	0.303
	OR look		SD	MAD	SDest

Standard deviation of simulated group effects and square root of median of estimated variances after re

	OR	look	SD	MAD	SDest
1:	0.30	74	0.512	0.522	0.473
2:	0.30	108	0.412	0.416	0.386
3:	0.30	142	0.349	0.355	0.334
4:	0.30	178	0.303	0.305	0.297
5:	0.40	74	0.507	0.477	0.470
6:	0.40	108	0.418	0.399	0.385
7:	0.40	142	0.354	0.336	0.333
8:	0.40	178	0.307	0.288	0.297
9:	0.50	74	0.498	0.512	0.476
10:	0.50	108	0.400	0.399	0.387
11:	0.50	142	0.337	0.337	0.334
12:	0.50	178	0.299	0.318	0.297
13:	0.55	74	0.520	0.480	0.475
14:	0.55	108	0.410	0.391	0.388
15:	0.55	142	0.354	0.353	0.336
16:	0.55	178	0.306	0.317	0.298
17:	0.60	74	0.541	0.490	0.476
18:	0.60	108	0.422	0.409	0.388
19:	0.60	142	0.356	0.355	0.336
20:	0.60	178	0.314	0.307	0.298
21:	0.65	74	0.493	0.496	0.475
22:	0.65	108	0.396	0.396	0.388

```

23: 0.65 142 0.336 0.339 0.337
24: 0.65 178 0.296 0.286 0.299
25: 0.70 74 0.499 0.484 0.473
26: 0.70 108 0.397 0.386 0.389
27: 0.70 142 0.337 0.324 0.336
28: 0.70 178 0.302 0.288 0.299
29: 0.80 74 0.509 0.484 0.482
30: 0.80 108 0.413 0.390 0.391
31: 0.80 142 0.352 0.340 0.338
32: 0.80 178 0.309 0.319 0.300
33: 1.00 74 0.514 0.475 0.484
34: 1.00 108 0.412 0.395 0.394
35: 1.00 142 0.351 0.354 0.342
36: 1.00 178 0.311 0.309 0.303
      OR look      SD      MAD SDest

```

```
examSim(sim2, desc='Partial PO Markov model with varying treatment effect')
```

Comparison of OR and HR

```

      OR  HR
1: 0.30 0.34
2: 0.40 0.43
3: 0.50 0.52
4: 0.55 0.56
5: 0.60 0.61
6: 0.65 0.64
7: 0.70 0.71
8: 0.80 0.79
9: 1.00 1.00

```

Power of Cox test at last look for time until Y=1

```

      OR power
1: 0.30 1.00
2: 0.40 1.00
3: 0.50 0.93
4: 0.55 0.86
5: 0.60 0.74
6: 0.65 0.62
7: 0.70 0.43
8: 0.80 0.21
9: 1.00 0.05

```

Power of Markov proportional odds model test

```

      OR look power
1: 0.30 74 0.64
2: 0.30 108 0.82
3: 0.30 142 0.91
4: 0.30 178 0.97
5: 0.40 74 0.40
6: 0.40 108 0.55
7: 0.40 142 0.69
8: 0.40 178 0.79
9: 0.50 74 0.24

```

10:	0.50	108	0.35
11:	0.50	142	0.44
12:	0.50	178	0.53
13:	0.55	74	0.21
14:	0.55	108	0.28
15:	0.55	142	0.36
16:	0.55	178	0.44
17:	0.60	74	0.16
18:	0.60	108	0.23
19:	0.60	142	0.30
20:	0.60	178	0.34
21:	0.65	74	0.11
22:	0.65	108	0.18
23:	0.65	142	0.23
24:	0.65	178	0.26
25:	0.70	74	0.10
26:	0.70	108	0.13
27:	0.70	142	0.14
28:	0.70	178	0.17
29:	0.80	74	0.06
30:	0.80	108	0.08
31:	0.80	142	0.09
32:	0.80	178	0.11
33:	1.00	74	0.04
34:	1.00	108	0.05
35:	1.00	142	0.04
36:	1.00	178	0.05

OR look power

Spearman rho correlation between Markov and Cox model chi-square: 0.77

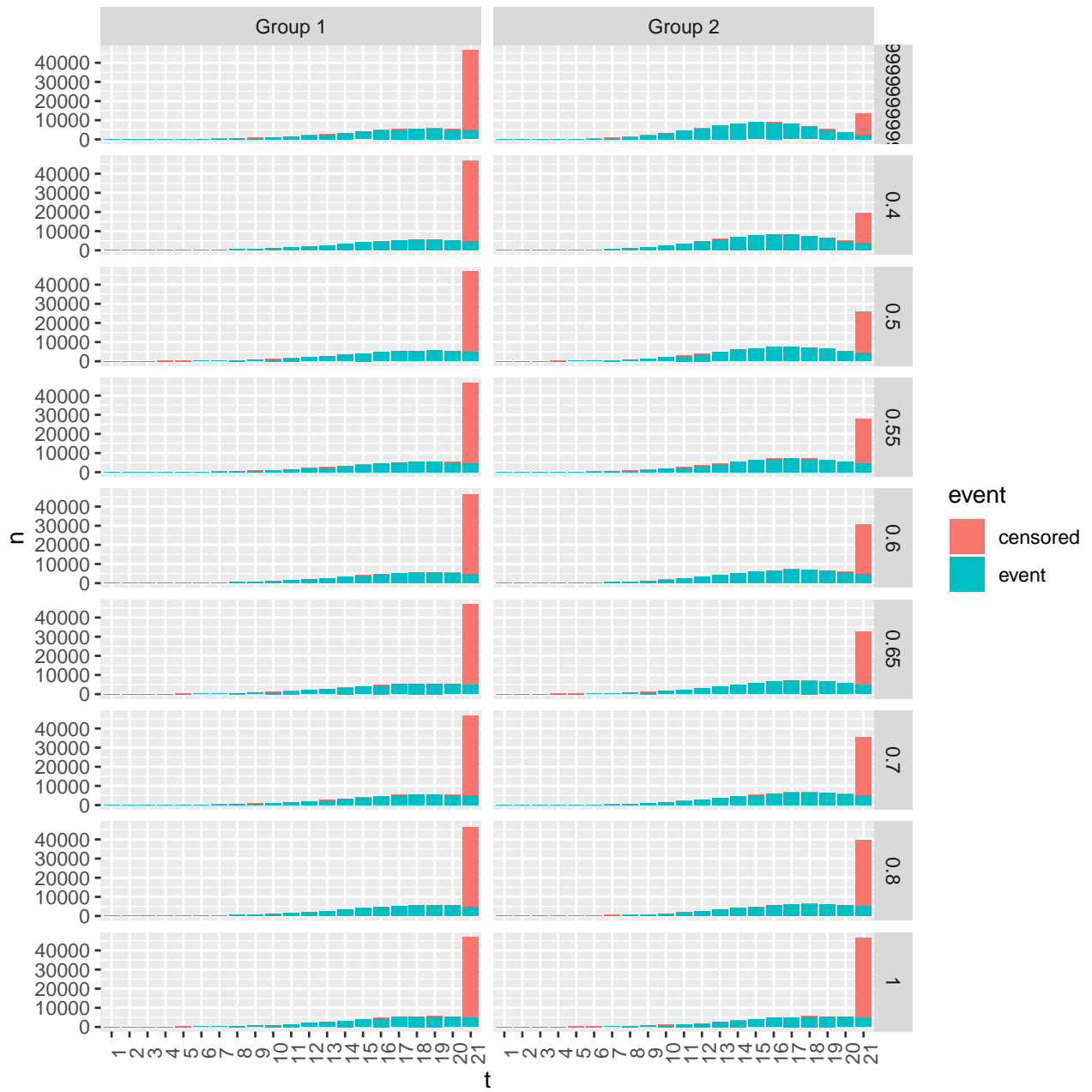


Figure 4: Time to $Y=1$ by group and OR, over simulations. Partial PO Markov model with varying treatment effect.

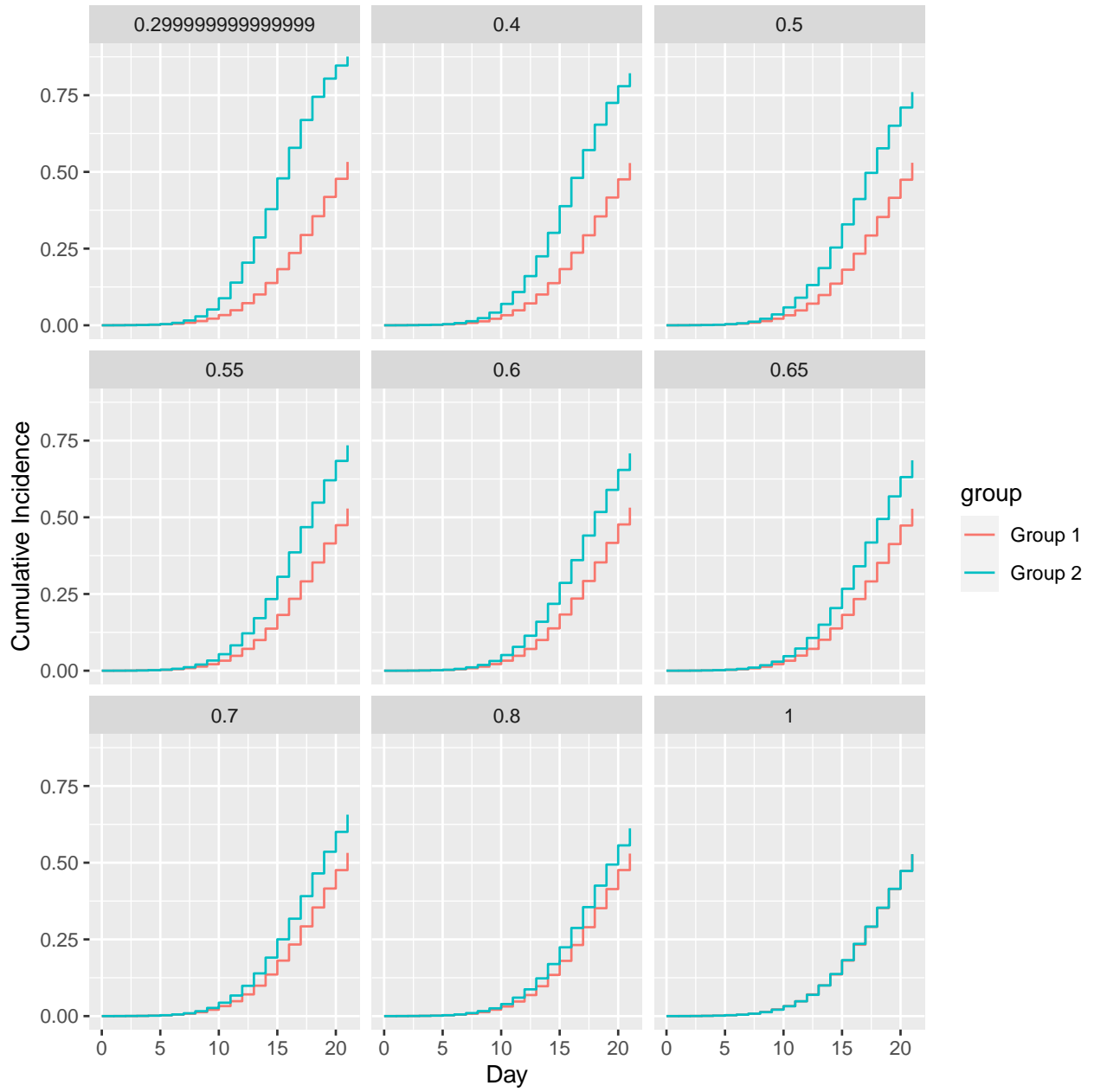


Figure 5: Cumulative incidence of $Y=1$. Partial PO Markov model with varying treatment effect.

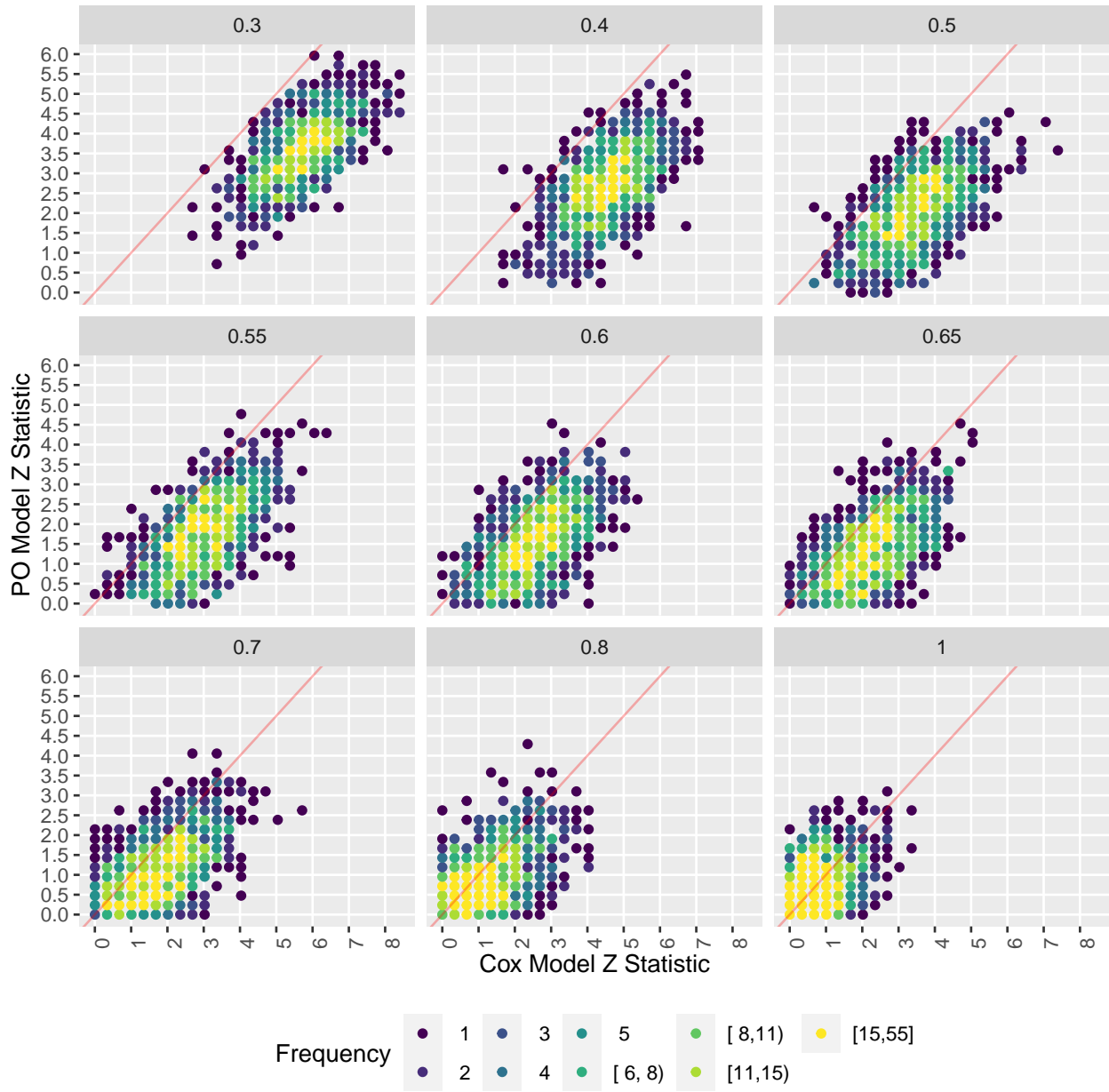


Figure 6: Scatter plot of Cox and PO model Z statistics. Partial PO Markov model with varying treatment effect.

```
knitr::knit_exit()
```