Stage 1 modeling simulation to assess coverage

Austin Schumacher

2021-06-25

Contents

Three categories (i.e. two causes of death + alive)

Data generating mechanism

For 100,000 women, the number of births were generated from a Poisson distribution with rate of 3. We then generate deaths for these children using the following process.

Let $i = 1, \ldots, N$ index births, let $t = 0, \ldots, T = 59$ index time in months. Divide the 60 months into $J = 6$ age groups indexed by $j[t]$ such that

$$j[t] = \begin{cases} 1 \text{ if } t = 0 \\ 2 \text{ if } 1 \leq t \leq 11 \\ 3 \text{ if } 12 \leq t \leq 23 \\ 4 \text{ if } 24 \leq t \leq 35 \\ 5 \text{ if } 36 \leq t \leq 47 \\ 6 \text{ if } 48 \leq t \leq 59 \end{cases}$$

Let there be $M = 3$ categories, indexed by $m \in \{1, 2, 3\}$, for a Multinomial distribution. Category $m = 3$ will be "alive" and categories $m = 1$ and $m = 2$ will correspond to "death from cause m." For each individual, we will draw from a Multinomial distribution starting at time $t = 0$ until the first draw in one of the non-alive categories. We will have two scenarios with different probabilities. In scenario one, we will have different probabilities for each age group, $\pi_{j[t],m}$, that are constant across clusters. For scenario two, we will have different probabilities for each age group that are cluster-specific. We will specify regression parameters for a base-line category multinomial logistic regression model to generate the data. For $c = 1, \ldots, N_c$, let $c[i]$ index the cluster for individual i. Also, the time of death for individual i will be T_i and let $t = 1, \ldots, T_i$ index months. Define $\boldsymbol{y_{it}} = (y_{it1}, \ldots, y_{itM})$ and $\boldsymbol{\pi_{c[i],t}} = (\pi_{c[i],t1}, \ldots, \pi_{c[i],tM})$. We will generate deaths as

$$\begin{aligned} \boldsymbol{y_{it}} | \boldsymbol{\pi_{c[i]t}} &\sim \text{Multinomial}(1; \boldsymbol{\pi_{c[i]t}}) \\ \pi_{c[i]tm} &= \frac{\exp(\beta_{j[t],m} + \epsilon_{c[i]})}{1 + \sum_{m=1}^{2} \exp(\beta_{j[t],m} + \epsilon_{c[i]})} \text{ for } m \in \{1, 2\} \end{aligned}$$

We will set the $\beta_{j[t],m}$ parameters such that the age-group-specific monthly probabilities $(\text{expit}(\beta_{1,m}), \ldots, \text{expit}(\beta_{6,m})) = \delta_m \times (0.04, 0.005, 0.004, 0.003, 0.002, 0.001)$, with $\delta_m = \begin{cases} 0.3 \text{ if } m = 1 \\ 0.7 \text{ if } m = 2 \end{cases}$. We can think of δ_m as the overall cause-fraction (not age-specific) for cause m.

For scenario 1, we will set all $\epsilon_c = 0$. For scenario 2, we will let $\epsilon_c \overset{iid}{\sim} N(0, 0.3)$.

Cluster sampling coverage simulation

After we have this population of children, we will perform $S = 100$ simulations where we perform a two stage cluster sampling design, fit a baseline category multinomial logistic regression model to estimate monthly probabilities of death, calculate a 95% CI for estimated ${}_5q_0^m$ based on our design-based asymptotic variance estimator and based on a jackknife estimator, and assess whether the 95% confidence intervals cover the true values of ${}_5q_0^m$ from the population.

At the first stage of our two-stage cluster sampling design, n_c clusters were randomly selected from the N_c available. At the second stage, suppose cluster c is selected, then n_W women were randomly selected from the N_{Wc} total women within the selected cluster. The resulting sampling weight for a woman in cluster c is

$$w_{Ec} = \frac{N_c}{n_c} \times \frac{N_{Wc}}{n_W}$$

We set $N_c = 500$, $n_c = 25$, and set $n_W \in (10, 20, 30)$. Note that N_{Wc} is random due to the random allocation of the 100,000 women to the 500 clusters.

For all models, we will organize the data such that each observation is a person-month, so individual i will contribute a sequence of vectors of length 3: $T_i - 1$ vectors (0, 0, 1) followed by a single (1, 0, 0) or (0, 1, 0) for month T_i—except if no death was observed, then it will be all (0, 0, 1) vectors—and each of these 0s and 1s will be its own row in the data set with person-month indicator variables defined corresponding to cause 1, cause 2, and alive.

To estimate cause-specific probabilites of death, we will then fit a baseline category multinomial logistic regression model with J age-group-specific intercepts and no overall intercept, accounting for the survey design using the `surveyVGAM` package in R. This will give us 12 estimated regression coefficients, $\hat{\beta}_{jm}$ parameters, $m = 1, 2$. Then each monthy probability of dying is calculated as

$$\lambda_m(t) = \frac{\exp(\hat{\beta}_{j[t],m})}{1 + \sum_{m'=1}^{2} \exp(\hat{\beta}_{j[t],m'})}$$

for $t = 0, \ldots, 59$. Then, we calculate $\widehat{{}_5q_0^m}$ using these cause-specific hazards as

$$\widehat{{}_5q_0^m} = \sum_{t=0}^{59} \left[\lambda_m(t) \prod_{t'=0}^{t-1} (1 - \sum_{m=1}^{2} \lambda_m(t')) \right]$$

We will then extract the covariance matrix and use a simulation-based method to calculate an asymptotic design-based variance estimate. To do this, we will simulate $B = 10000$ draws from the multivariate normal distribution using the 12 estimated regression coefficients and the 12 by 12 design-based covariance matrix. For each of these draws, we will calculate $\widehat{{}_5q_0^m}$, and then we will calculate the asymptotic design-based variance estimates as the varaince of these draws. We will use this to calculate 95% CIs.

We will also calculate a jackknife variance estimate of $\widehat{{}_5q_0^m}$, separately for each of the $m = 1, 2$ causes, as

$$V^m_{JACK} = \frac{n_c - 1}{n_c} \sum_{c=1}^{n_c} (\widehat{{}_5q_0^m(c)} - \widehat{{}_5q_0^m})^2$$

where $\widehat{{}_5q_0^m(c)}$ is the estimate based on all the data while holding out cluster c, and use this to calculate a 95% CIs for comparison.

```r
# function to calculate 5q0^c from multinomial model
get_5q0_multi <- function(beta, n) {
    ## For testing
    # beta <- coef(mod.multi.pop)
    # n <- c(1, 11, 12, 12, 12, 12)

    betas_of_interest <- beta[seq(1,length(beta), by = 2)]
    betas_other <- beta[seq(2,length(beta), by = 2)]

    betas_of_interest_monthly <- rep(betas_of_interest, times = n)
    betas_other_monthly <- rep(betas_other, times = n)
    one_plus_sum_exp_betas_monthly <- 1 + exp(betas_of_interest_monthly) + exp(betas_other_monthly)
    lambda_of_interest_monthly <- exp(betas_of_interest_monthly) / one_plus_sum_exp_betas_monthly
    lambda_other_monthly <- exp(betas_other_monthly) / one_plus_sum_exp_betas_monthly
    lambda_monthly <- lambda_of_interest_monthly + lambda_other_monthly
    terms_of_interest <- rep(NA, sum(n))
    terms_of_interest[1] <- lambda_of_interest_monthly[1]
    terms_other <- rep(NA, sum(n))
    terms_other[1] <- lambda_other_monthly[1]
    for (i in 2:sum(n)) {
        terms_of_interest[i] <- lambda_of_interest_monthly[i] * prod(1-lambda_monthly[1:(i-1)])
        terms_other[i] <- lambda_other_monthly[i] * prod(1-lambda_monthly[1:(i-1)])
    }

    phi_of_interest <- sum(terms_of_interest)
    phi_other <- sum(terms_other)

    return(c(phi_of_interest, phi_other))
}

# set seed
set.seed(96)

# parameters
nsim <- 100 # number of simulations for each run
J <- 6 # number of age groups
ns <- c(1, 11, 12, 12, 12, 12) # number of months per age group
T <- sum(ns) # total months
Nw <- 100000 # number of women
Nc <- 500 # number of clusters
methods <- c("Design-based", "Jackknife") # methods for calculating CIs

## looping parameters
# ncs <- c(15, 25) # number of clusters to sample
ncs <- 25
nws <- seq(10, 30, 10)
# nws <- 15
scenarios <- c("Cluster constant",
               # "Cluster-specific, low var",
               "Cluster-specific, high var")
cluster_sigmas <- c(0,
                    # 0.1,
                    0.3)

# results storage
results <- expand_grid(scenario = scenarios,
                       nc = ncs,
                       nw = nws,
                       cause = c("cause of interest", "cause other"),
                       method = methods)
results %<>% mutate(coverage = NA)
elapsed.times <- expand_grid(scenario = scenarios,
                             nc = ncs,
                             nw = nws)
```

```r
elapsed.times %<>% mutate(time_in_seconds = NA)

# starting loop through different simulation parameters
for (scenario_number in 1:length(scenarios)) {
    # testing
    # scenario_number <- 2

    scenario <- scenarios[scenario_number]
    # cat(paste0("\n Starting \n \t Scenario: ", scenario, ";\n"))
    for (ii in 1:length(ncs)) {
        # testing
        # ii <- 1

        nc <- ncs[ii]
        # cat(paste0("\t\t Number of clusters: ", nc, ";\n"))
        for (jj in 1:length(nws)) {
            # testing
            # jj <- 1

            nw <- nws[jj]
            # cat(paste0("\t\t\t Number of women: ", nw, ";\n"))
            start.time <- proc.time()

            birth_lambda <- 3
            nbirths <- rpois(Nw, birth_lambda)
            clusterid <- sort(sample(1:Nc, Nw, replace = TRUE))
            Nwc <- as.vector(table(clusterid))
            weights <- (Nc/nc) * (Nwc/nw)
            clusterprobs <- 1/weights

            # create data set for population where each row is a birth
            # and simulate deaths
            dat <- tibble(motherid = 1:Nw,
                          clusterid = clusterid) %>%
                group_by(clusterid) %>%
                mutate(withinclustermotherid = sequence(n()))
            dat <- dat[rep(1:Nw, times = nbirths),]
            weights.dat <- tibble(clusterid = 1:Nc,
                                  weight = weights,
                                  prob = clusterprobs)
            dat %<>% left_join(weights.dat, by = "clusterid") %>%
                group_by(motherid) %>%
                mutate(birthid = sequence(n())) %>%
                ungroup()  %>%
                mutate(id = 1:n(),
                       clustermotherbirthid = paste(clusterid, motherid, birthid, sep = "_"),
                       clustermotherid = paste(clusterid, withinclustermotherid, sep = "_"))

            # simulate deaths
            N <- nrow(dat)

            probs <- c(0.04, 0.005, 0.004, 0.003, 0.002, 0.001)
            deltas <- c(0.3, 0.7)
            probs_mat <- probs%*%t(deltas)
            probs_mat <- cbind(probs_mat, apply(probs_mat, 1, function(x) {1 - sum(x)}))
            betas <- logitlink(probs_mat[,1:2])
            betas_vec <- as.vector(t(betas))
            if (grepl("Cluster-specific", scenario)) {
                epsilons <- rnorm(Nc, 0, cluster_sigmas[scenario_number])
            } else {
                epsilons <- rep(0, Nc)
            }

            ytmp <- c()
            for (ccc in 1:Nc) {
                number_of_births <- nrow(dat %>% filter(clusterid == ccc))
                tmp.probs <- expit(betas + epsilons[ccc])
                tmp.probs <- cbind(tmp.probs, apply(tmp.probs, 1, function(x) {1 - sum(x)}))
                tmp.pis <- tmp.probs[rep(1:J, times = ns),]
                tmp.pis <- rbind(tmp.pis, c(0, 0, 1))
                ytmp <- c(ytmp, as.vector(rMultinom(tmp.pis, number_of_births)))
            }

            dat <- dat[rep(1:N, each = T + 1), ]
            dat$t <- rep((0:T), N)
            dat$a <- rep(rep(1:(J + 1), times = c(ns, 1)), N)
            dat$cause1 <- as.numeric(ytmp == 1)
            dat$cause_other <- as.numeric(ytmp == 2)
            dat$alive <- as.numeric(ytmp == 3)
            dat %<>% group_by(id) %>%
                filter(pmax(cumsum(cumsum(cause1 == 1)),
                            cumsum(cumsum(cause_other == 1))) <= 1L) %>%
                ungroup() %>%
                filter(a != 7)

            # get 5q0s
            dat_ind <- dat %>% group_by(id) %>%
                slice_tail(n = 1)
            true5q0 <- c(mean(dat_ind$cause1), mean(dat_ind$cause_other))

            # results storage
            coverage <- matrix(NA, nrow = nsim, ncol = 2)
            coverage.jack <- matrix(NA, nrow = nsim, ncol = 2)

            # cat(paste0("\t\t\t\t Starting sims: 1... "))
            # start simulations
            for (s in 1:nsim) {
                # testing
                # s <- 1
                # if (s %% 20 == 0) cat(paste0(s, "... "))

                # sample selection
                sampled_clusters <- sample(1:Nc, nc, replace = FALSE)
                sampled_women <- vector(mode = "list", length = nc)
                for (c in 1:length(sampled_clusters)) {
                    sampled_women[[c]] <- sample(1:Nwc[sampled_clusters[c]], nw, replace = FALSE)
```

```
}
sampled_ids <- tibble(clusterid = rep(sampled_clusters, each = nw),
                      withinclustermotherid = unlist(sampled_women)) %>%
    mutate(clustermotherid = paste(clusterid, withinclustermotherid, sep = "_"))
dat.sampled <- dat %>% filter(clustermotherid %in% sampled_ids$clustermotherid)

# compactify
dat.sampled.comp <- dat.sampled[, c("clusterid", "weight", "a",
                                    "alive", "cause1", "cause_other")]
formula <- as.formula(". ~ clusterid + a + weight")
dat.sampled.comp <- aggregate(formula, data = dat.sampled.comp, FUN = sum, drop = TRUE)

# survey design
my.svydesign <- survey::svydesign(ids = ~ clusterid,
                                  strata = NULL, weights = ~ weight,
                                  data = dat.sampled.comp)

# survey multinom
mod.multi <- svy_vglm(cbind(cause1, cause_other, alive) ~ -1 + factor(a),
                      design = my.svydesign, rescale = TRUE,
                      family = multinomial)

betahats <- coef(mod.multi)
est5q0s <- get_5q0_multi(betahats, ns)

V <- stats::vcov(mod.multi)
betasim <- rmvnorm(10000, mean = betahats, sigma = V)
q5.sim <- t(apply(betasim, 1, function(x) get_5q0_multi(x, ns)))
bounds <- apply(q5.sim, 2, quantile, c(0.025, 0.975))

coverage[s, ] <- c(bounds[1, 1] < true5q0[1] & bounds[2, 1] > true5q0[1],
                   bounds[1, 2] < true5q0[2] & bounds[2, 2] > true5q0[2])

## jackknife estimator
jack5q0s <- matrix(NA, nrow = nc, ncol = 2)
clusts <- unique(dat.sampled.comp$clusterid)
for (cc in 1:nc) {
    clust <- clusts[cc]
    tmp <- dat.sampled.comp[dat.sampled.comp$clusterid != clust,]

    my.svydesign.tmp <- survey::svydesign(ids = ~ clusterid,
                                          strata = NULL, weights = ~ weight,
                                          data = tmp)

    ## binomial model
    mod.multi.tmp <- svy_vglm(cbind(cause1, cause_other, alive) ~ -1 + factor(a),
                              design = my.svydesign.tmp, rescale = TRUE,
                              family = multinomial)
    betas.tmp <- coef(mod.multi.tmp)

    # estimates
    jack5q0s[cc,] <-get_5q0_multi(betas.tmp, ns)
}
```

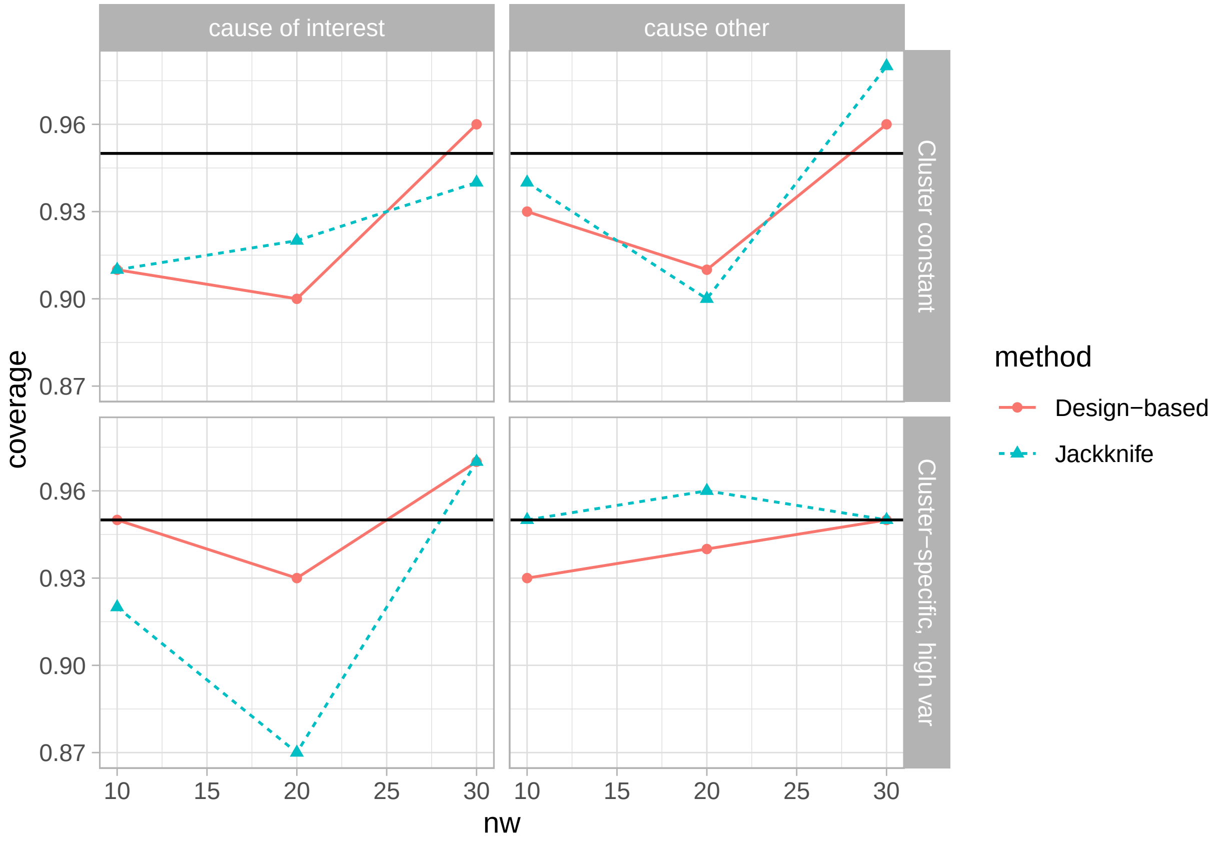
```
                Vjack <- rep(NA, 2)
                Vjack[1] <- ((nc - 1)/nc) * sum((jack5q0s[, 1] - est5q0s[1])^2)
                Vjack[2] <- ((nc - 1)/nc) * sum((jack5q0s[, 2] - est5q0s[2])^2)
                bounds.jack <- matrix(NA, nrow = 2, ncol = 2)
                bounds.jack[, 1] <- rep(est5q0s[1], 2) + rep(Vjack[1]^0.5, 2)*qnorm(c(0.025, 0.975))
                bounds.jack[, 2] <- rep(est5q0s[2], 2) + rep(Vjack[2]^0.5, 2)*qnorm(c(0.025, 0.975))
                coverage.jack[s, ] <- c(bounds.jack[1, 1] < true5q0[1] & bounds.jack[2, 1] > true5q0[1]
                                        bounds.jack[1, 2] < true5q0[2] & bounds.jack[2, 2] > true5q0[2]
            }

            # asymptotic design based coverage
            design.cover <- apply(coverage, 2, mean)

            # jackknife coverage
            jack.cover <- apply(coverage.jack, 2, mean)

            # store results
            results$coverage[(results$scenario == scenario) & (results$nc == nc) & (results$nw == nw) &
            results$coverage[(results$scenario == scenario) & (results$nc == nc) & (results$nw == nw) &
            results$coverage[(results$scenario == scenario) & (results$nc == nc) & (results$nw == nw) &
            results$coverage[(results$scenario == scenario) & (results$nc == nc) & (results$nw == nw) &
            stop.time <- proc.time()
            elapsed.time <- stop.time[3] - start.time[3]
            elapsed.times$time_in_seconds[(elapsed.times$scenario == scenario) & (elapsed.times$nc == n
            rm(list = c("dat", "dat.sampled", "dat.sampled.comp", "my.svydesign", "my.svydesign.tmp", "
        }
    }
}
```

Plot results

```
elapsed.times
```

```
## # A tibble: 6 x 4
##   scenario                      nc    nw time_in_seconds
##   <chr>                      <dbl> <dbl>           <dbl>
## 1 Cluster constant              25    10            850.
## 2 Cluster constant              25    20            838.
## 3 Cluster constant              25    30            812.
## 4 Cluster-specific, high var    25    10            814.
## 5 Cluster-specific, high var    25    20            834.
## 6 Cluster-specific, high var    25    30            826.
```

```
results.plot <- results %>% ggplot(aes(x = nw, y = coverage, col = method, lty = method, pch = method))
    geom_line() + geom_point() +
    facet_grid(scenario ~ cause) +
    geom_hline(yintercept = 0.95, col = "black", size = 0.5) +
    theme_light()
results.plot
```


cause of interest
cause other
Cluster constant
Cluster−specific, high var
coverage
0.96
0.93
0.90
0.87
10
15
20
25
30
nw
method
Design−based
Jackknife