

Individual-level discrete time competing risks survival modeling

Austin Schumacher

2021-04-13

Simulation

Multinomial

3 categories

We will first simulate n draws from a multinomial distribution with 3 categories

```
library(svyVGAM)
library(SUMMER)
set.seed(98125)

# sim setup and storage of results
nsim <- 100
p.res <- matrix(NA, nrow = nsim, ncol = 2)
coverage.res <- matrix(NA, nrow = nsim, ncol = 2)
cov.res <- array(NA, dim = c(2, 2, nsim))

# parameters
n <- 1000
alpha <- c(-2, -1)
denom_p <- sum(exp(alpha)) + 1
p_cause <- c(exp(alpha)/denom_p)
p <- c(p_cause, 1 - sum(p_cause))

for (s in 1:nsim) {
  # cat(paste0("Starting sim ", s, "....\n"))
  # simulate data
  y <- rmultinom(n, 1, p)
  dat <- data.frame(obs = 1:n,
                    cause1 = y[1,],
                    cause_other = y[2,],
                    alive = y[3,])

  # fit model
  my.svydesign <- survey::svydesign(ids = ~ obs,
                                  data = dat)

  mult.mod <- svy_vglm(cbind(cause1, cause_other, alive) ~ 1,
                      family = multinomial,
                      design = my.svydesign)
  denom <- 1 + sum(exp(mult.mod$coef))
```

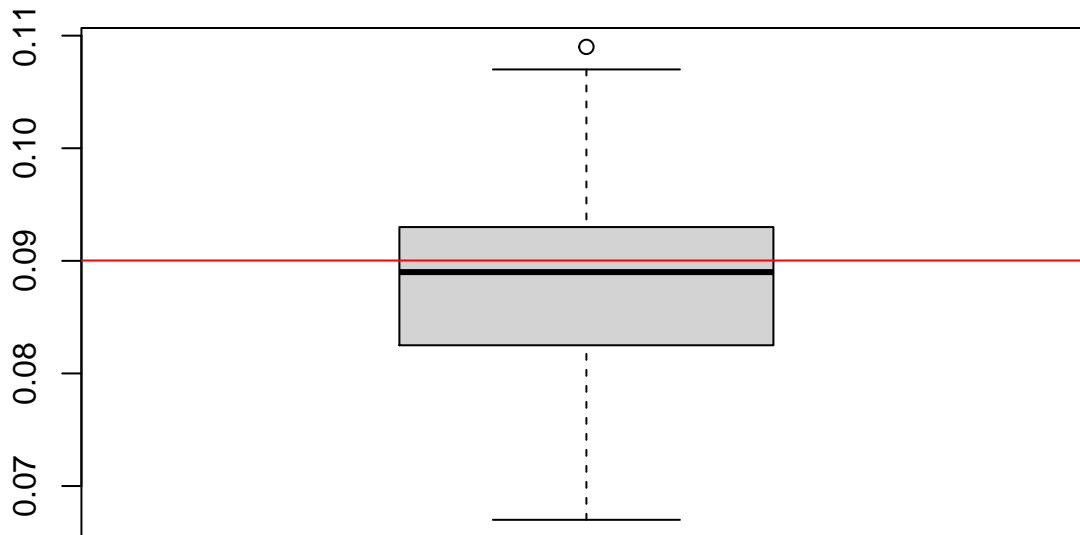
```

p.res[s, ] <- exp(mult.mod$coef)/denom
cis <- confint(mult.mod)
coverage.res[s, 1] <- alpha[1] > cis[1, 1] & alpha[1] < cis[1, 2]
coverage.res[s, 2] <- alpha[2] > cis[2, 1] & alpha[2] < cis[2, 2]
cov.res[,s] <- mult.mod$var
}

boxplot(p.res[, 1], main = "Posterior medians of p1", ylim = range(c(p.res[, 1], p[1])))
abline(h = p[1], col = "red")

```

Posterior medians of p1

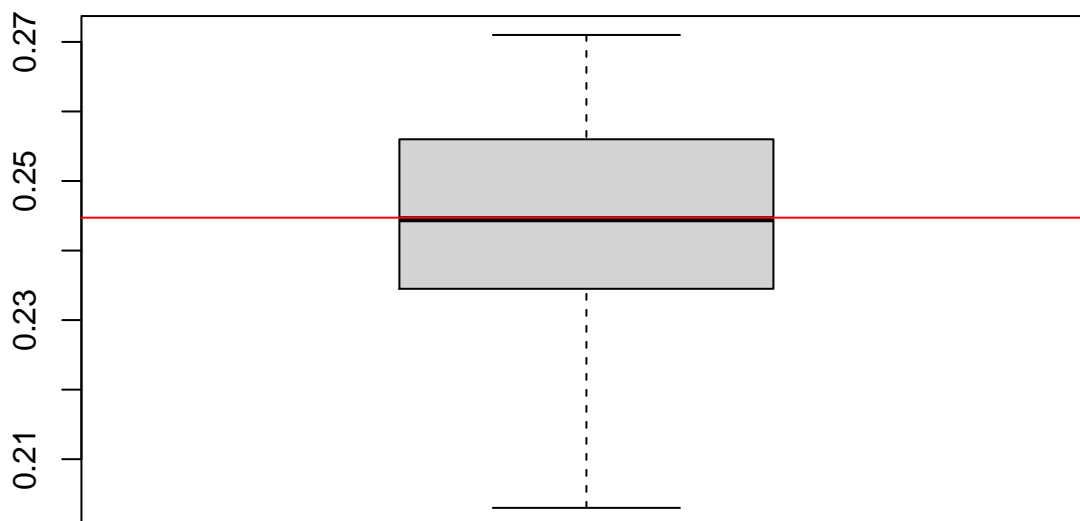


```

boxplot(p.res[, 2], main = "Posterior medians of p2", ylim = range(c(p.res[, 2], p[2])))
abline(h = p[2], col = "red")

```

Posterior medians of p2



```

apply(coverage.res, 2, mean)

```

```
## [1] 0.96 0.94
apply(cov.res, c(1, 2), mean)

##           [,1]      [,2]
## [1,] 0.012952300 0.001500962
## [2,] 0.001500962 0.005609496
```

Individual-level SRS

3 categories (1 cause, all other causes, and alive)

We will simulate n individuals over 60 months, where in each month for all alive individuals, we apply the multinomial probabilities of dying from cause 1, dying from all other causes, or remaining alive.

Thus, we have the discrete-time survival data generating mechanism as follows:

```
# sim setup and storage of results
nsim <- 100
p.res <- matrix(NA, nrow = nsim, ncol = 2)
coverage.res <- matrix(NA, nrow = nsim, ncol = 2)
cov.res <- array(NA, dim = c(2, 2, nsim))

# parameters
n <- 1000
alpha <- c(-4, -3)
denom_p <- sum(exp(alpha)) + 1
p_cause <- c(exp(alpha)/denom_p)
p <- c(p_cause, 1 - sum(p_cause))
p_cum <- cumsum(p)

for (s in 1:nsim) {
  # simulate data
  y <- vector(mode = "list", length = n)
  for(i in 1:n) {
    y[[i]] <- matrix(NA, ncol = 5, nrow = 60)
    y[[i]][, 4] <- i # person id
    y[[i]][, 5] <- 1:60 # month indicator
    # cat(paste("\nStarting person", i, "\n"))
    t <- 1
    death <- 0
    while(death == 0) {
      # if (t %% 10 == 0) cat(paste("\t month", t, "\n"))
      rand <- runif(1, 0, 1)
      if (rand < p_cum[1]) {
        y[[i]][t, 1:3] <- c(1, 0, 0)
        death <- 1
      } else if (rand >= p_cum[1] & rand < p_cum[2]) {
        y[[i]][t, 1:3] <- c(0, 1, 0)
        death <- 1
      } else {
        y[[i]][t, 1:3] <- c(0, 0, 1)
      }
      t <- t + 1
      if (t == 60) death <- 1
    }
  }
}
```

```

    }
  }

  ymat <- do.call(rbind, y)
  ymat <- ymat[complete.cases(ymat), ]

  # create data frame
  dat <- data.frame(id = ymat[, 4],
                    time = ymat[, 5],
                    cause1 = ymat[, 1],
                    cause_other = ymat[, 2],
                    alive = ymat[, 3])

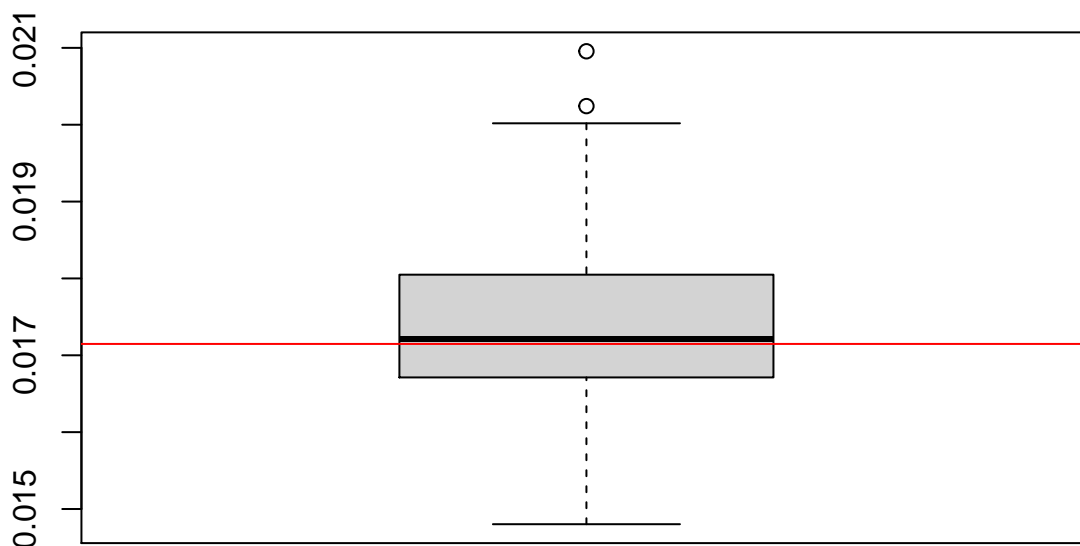
  # fit model
  my.svydesign <- survey::svydesign(ids = ~ id,
                                data = dat)

  mult.mod <- svy_vglm(cbind(cause1, cause_other, alive) ~ 1,
                      family = multinomial,
                      design = my.svydesign)
  denom <- 1 + sum(exp(mult.mod$coef))
  p.res[s, ] <- exp(mult.mod$coef)/denom
  cis <- confint(mult.mod)
  coverage.res[s, 1] <- alpha[1] > cis[1, 1] & alpha[1] < cis[1, 2]
  coverage.res[s, 2] <- alpha[2] > cis[2, 1] & alpha[2] < cis[2, 2]
  cov.res[,s] <- mult.mod$var
}

# results
boxplot(p.res[, 1], main = "Posterior medians of p1", ylim = range(c(p.res[, 1], p[1])))
abline(h = p[1], col = "red")

```

Posterior medians of p1

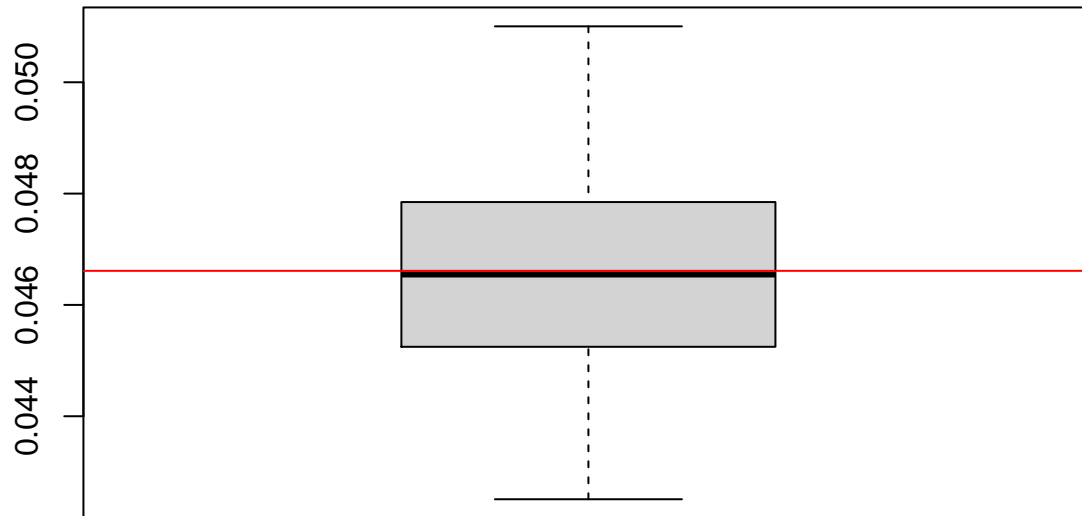


```

boxplot(p.res[, 2], main = "Posterior medians of p2", ylim = range(c(p.res[, 2], p[2])))
abline(h = p[2], col = "red")

```

Posterior medians of p2



```
apply(coverage.res, 2, mean)
```

```
## [1] 0.94 0.95
```

```
apply(cov.res, c(1, 2), mean)
```

```
##           [,1]      [,2]  
## [1,] 3.847064e-03 6.719748e-05  
## [2,] 6.719748e-05 1.468054e-03
```

Modeling BGD data