Standard Error Calibration

Zachary McCaw

2024-02-24

MCF Standard Error Calibration

Data generating process

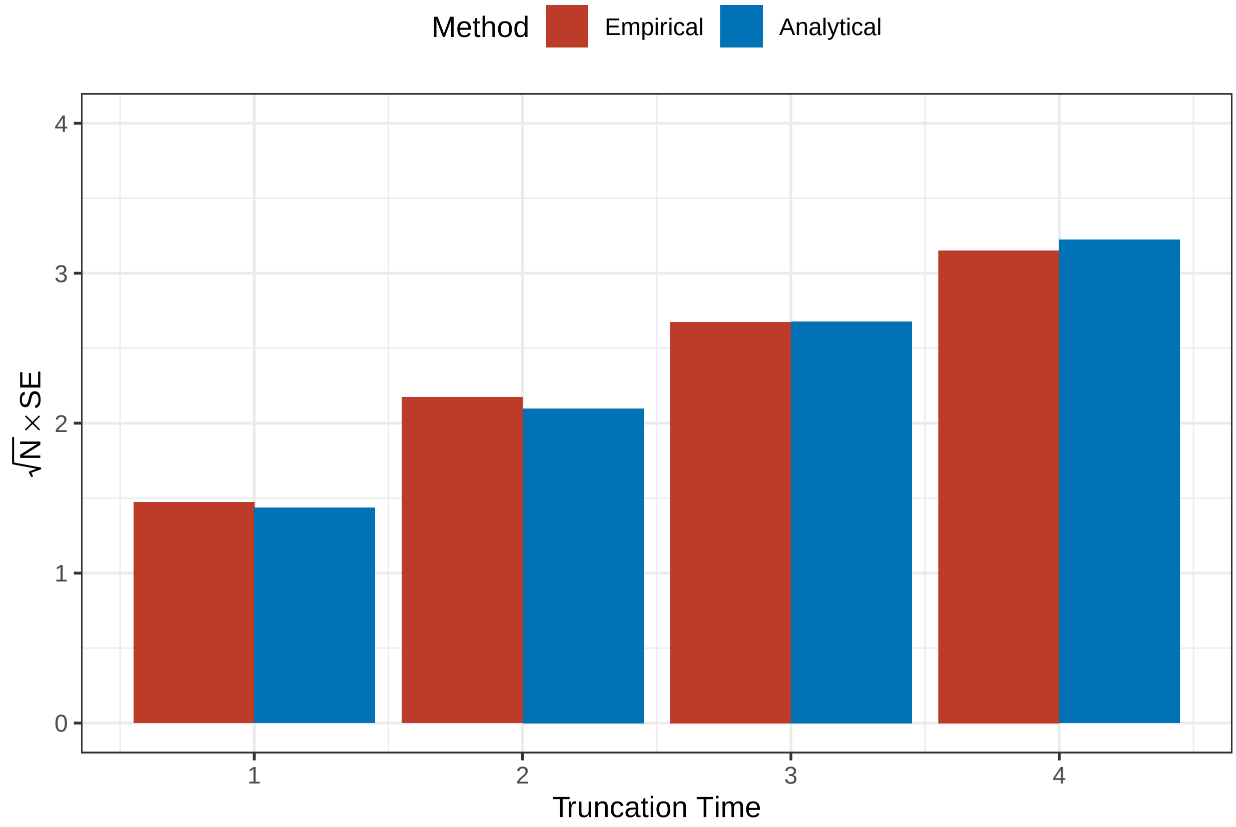
```
N <- 1e3
DGP <- function() {
  out <- GenData(
    base_event_rate = 0.5,
    base_death_rate = 0.2,
    censoring_rate = 0.2,
    tau = 5,
    n = N
  )
  out$weights <- 2
  return(out)
}
```

Simulation functions

```
#' Calculate MCF
GetMCF <- function(df) {
  mcf <- CalcMCF(
    idx = df$idx,
    status = df$status,
    time = df$time,
    weights = df$weights
  )
  return(mcf)
}

#' MCF curve.
Curve <- function(mcf) {
  out <- stats::stepfun(x = mcf$time, y = c(0, mcf$mcf))
  return(out)
}

#' SE curve.
SeCurve <- function(mcf) {
  out <- stats::stepfun(x = mcf$time, y = c(0, mcf$se_mcf))
  return(out)
```

```r
}

#' Simulation loop.
Loop <- function(i) {

  # Generate data.
  data <- DGP()

  # Calculate MCF.
  mcf <- GetMCF(data)
  mcf_fn <- Curve(mcf)
  mcf_se <- SeCurve(mcf)

  # Evaluate.
  taus <- seq(1:4)
  mcf_evals <- sapply(taus, mcf_fn)
  se_evals <- sapply(taus, mcf_se)

  # Output.
  out <- data.frame(
    idx = i,
    tau = taus,
    mcf = mcf_evals,
    ses = se_evals
  )
  return(out)
}

#' Simulation.
Sim <- function(reps = 1e3) {

  sim <- lapply(seq_len(reps), Loop)
  sim <- do.call(rbind, sim)

  results <- sim %>%
    dplyr::group_by(tau) %>%
    dplyr::summarise(
      reps = dplyr::n(),
      empirical_se = stats::sd(mcf),
      analytical_se = sqrt(mean(ses^2))
    )

  return(results)
}
```

```r
sim <- Sim(reps = 500)
```



AUMCF Standard Error Calibration

Data generating process

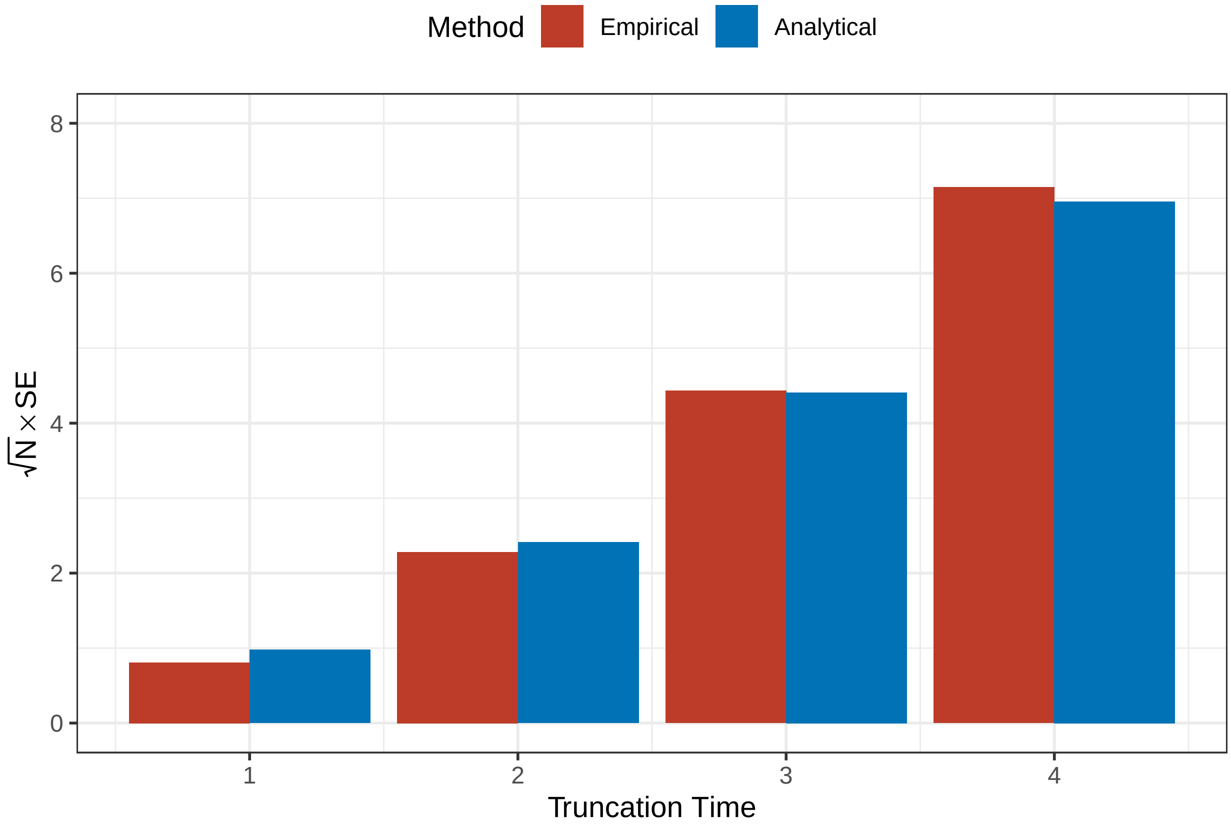
```
N <- 250
DGP <- function() {
  out <- GenData(
    base_event_rate = 0.5,
    base_death_rate = 0.2,
    censoring_rate = 0.2,
    tau = 5,
    n = N
  )
  out$weights <- 2
  return(out)
}
```

Simulation functions

```
#' Calculate MCF
GetAUC <- function(df, tau) {
  auc <- SingleArmAUC(
    data = df,
    weights = df$weights,
```

```r
    tau = tau
  )
  out <- auc@MargAreas
  return(out)
}

#' Simulation loop.
Loop <- function(i) {

  # Generate data.
  data <- DGP()

  # Calculate MCF.
  taus <- c(1, 2, 3, 4)
  out <- lapply(taus, function(t) {GetAUC(data, t)})
  out <- data.frame(do.call(rbind, out))

  # Output.
  out$arm <- NULL
  out$idx <- i
  return(out)
}

#' Simulation.
Sim <- function(reps = 1e3) {

  sim <- lapply(seq_len(reps), Loop)
  sim <- do.call(rbind, sim)

  results <- sim %>%
    dplyr::group_by(tau) %>%
    dplyr::summarise(
      reps = dplyr::n(),
      empirical_se = stats::sd(area),
      analytical_se = sqrt(mean(se^2))
    )

  return(results)
}
```

```r
sim <- Sim(reps = 500)
```


Method
Empirical
Analytical
8
6
4
2
0
$\sqrt{N} \times \mathrm{SE}$
1
2
3
4
Truncation Time