A4Q3

Jiaqi Bi

Stratified Unadjusted and Marginal

```
un_est <- function(a, b, c, d) {

  n1= a + c
  n2 = b + d
  m1 = a + b
  m2 = c + d

  p1 = a/(a+c)
  p2 = b/(b+d)
  RD = p1 - p2
  RR = p1/p2
  OR = (p1/(1-p1))/(p2/(1-p2))

  log.RR = log(RR)
  log.OR = log(OR)

  var.RD = (p1 * (1 - p1))/n1 + (p2 * (1-p2))/n1
  var.logRR = 1/a - 1/n1 + 1/b + 1/n2
  var.logOR = 1/a + 1/b + 1/c + 1/d

  CI.RD = c(RD - 1.96 * sqrt(var.RD), RD + 1.96 * sqrt(var.RD))
  CI.logRR = c(log.RR - 1.96 * sqrt(var.logRR),
               log.RR + 1.96 * sqrt(var.logRR))
  CI.logOR = c(log.OR - 1.96 * sqrt(var.logOR),
               log.OR + 1.96 * sqrt(var.logOR))
  CI.RR = exp(CI.logRR)
  CI.OR = exp(CI.logOR)


  return(list(
    RD=RD,
    RR=RR,
    OR=OR,
    log.RR=log.RR,
    log.OR=log.OR,
    var.RD=var.RD,
    var.logRR=var.logRR,
    var.logOR=var.logOR,
    CI.RD=CI.RD,
    CI.RR=CI.RR,
    CI.OR=CI.OR
  ))
}

## Stratum 1
un_est(18, 162, 25, 252)
## Stratum 2
un_est(12, 26, 123, 431)
## Stratum 3
un_est(27, 121, 104, 475)
## Stratum 4
un_est(7, 21, 3, 25)
## Stratum 5
un_est(14, 353, 7, 359)
## Marginal
a <- c(18, 12, 27, 7, 14)
b <- c(162, 26, 121, 21, 353)
c <- c(25, 123, 104, 3, 7)
d <- c(252, 431, 475, 25, 359)
a <- sum(a)
b <- sum(b)
c <- sum(c)
d <- sum(d)
un_est(a, b, c, d)
```

Stratified-Adjusted Estimator MH Estimates

```
voltage <- array(
  c(18, 162, 25, 252,
    12, 26, 123, 431,
    27, 121, 104, 475,
    7, 21, 3, 25,
    14, 353, 7, 359),
  dim = c(2, 2, 5),
  dimnames = list(response = c("Case", "Control"),
               treatment = c("<100m", ">100m"),
               strata = c("Study 1", "Study 2", "Study 3", "Study 4", "Study 5"))
)

ak <- voltage[1,1,]
bk <- voltage[1,2,]
ck <- voltage[2,1,]
dk <- voltage[2,2,]
strata.spe.data <- data.frame(ak, bk, ck, dk)
MH.adj_est <- function(ak, bk, ck, dk) {

  ## Pre-check on input validation
  suppressWarnings(
   if(is.vector(any(ak, bk, ck, dk)) == FALSE) {
    stop("Input invalid")
  }
  )

  ## Sample size calculation
  Nk = ak + bk + ck + dk
  n1k = ak + ck
  n2k = bk + dk
  m1k = ak + bk
  m2k = ck + dk

  ## Calculate MH OR
  nume.OR <- sum((ak * dk)/Nk)
  deno.OR <- sum((bk * ck)/Nk)
  OR.mh <- nume.OR/deno.OR

  ## Calculate MH RR
  nume.RR <- sum((ak * n2k)/Nk)
  deno.RR <- sum((ak * n1k)/Nk)
  RR.mh <- nume.RR/deno.RR

  ## Stratified-adjusted MH test and CI
  chi.square.MH <- ((sum(ak-n1k * (m1k/Nk)))^2)/(sum((m1k * m2k * n1k * n2k)/(Nk^2 * (Nk
  #### RR CI
  log.RR = log(RR.mh)
  var.logRR.mh = (log.RR^2)/(chi.square.MH^2)
  CI.logRR.MH = c(log.RR - 1.96 * sqrt(var.logRR.mh),
                  log.RR + 1.96 * sqrt(var.logRR.mh))
  CI.RR = exp(CI.logRR.MH)
  #### OR CI
  log.OR = log(OR.mh)
  var.logOR.mh = (log.OR^2)/(chi.square.MH^2)
  CI.logOR.mh = c(log.OR - 1.96 * sqrt(var.logOR.mh),
                  log.OR + 1.96 * sqrt(var.logOR.mh))
  CI.OR = exp(CI.logOR.mh)

  ## Output
  return(list(
    OR.mh = OR.mh,
    RR.mh = RR.mh,
    var.RR = var.logRR.mh,
    var.OR = var.logOR.mh,
    CI.RR = CI.RR,
    CI.OR = CI.OR
  ))
}

MH.adj_est(ak, bk, ck, dk)
```

Conclusion