流行病學與生物統計計算 Homework 7
學號：b07401048　系級：醫學五　姓名：賴柏瑞

# Homework10

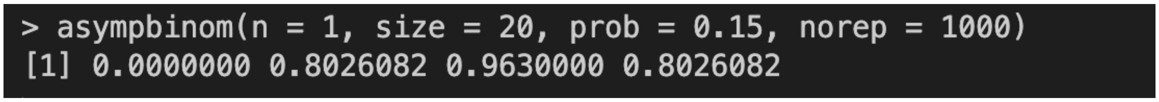
#EX 21

```
# asympotic 95% CI
        asympbinom <- function(n, size, prob, norep) {
            lower95 <- rep(NA, norep)
            upper95 <- rep(NA, norep)

            for (i in 1 : norep) {
                set.seed(i)
                x <- rbinom(n = n, size = size, prob = prob)
                phat <- (x / size)
                lower95[i] <- phat - (qnorm(0.975) * sqrt(((phat) * (1 - phat)) /
n))
                upper95[i] <- phat + (qnorm(0.975) * sqrt(((phat) * (1 - phat)) /
n))
            }
            for (i in 1 : norep) {
                if (lower95[i] <= 0) {
                    lower95[i] <- 0
                }
            }
            lower <- mean(lower95)
            upper <- mean(upper95)
            coverage <- mean((lower95 <= prob) & (upper95 >= prob))
            length <- mean(upper95 - lower95)

            return(c(lower, upper, coverage, length, mean(upper95)))
        }

        asympbinom(n = 1, size = 20, prob = 0.15, norep = 1000)
```



```
> asympbinom(n = 1, size = 20, prob = 0.15, norep = 1000)
[1] 0.0000000 0.8026082 0.9630000 0.8026082
```

Lower bound : 0.00

Upper bound : 0.80

Coverage : 0.96

Length : 0.80

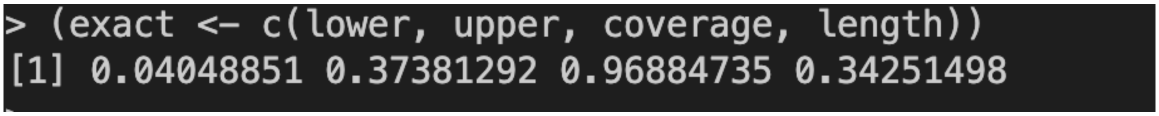
```
# exact 95% CI
        norep <- 1000
        randomnum <- rep(NA, norep)
        for (i in 1 : norep) {
            set.seed(i)
            random <- rbinom(n = 1, size = 20, prob = 0.15)
            randomnum[i] <- random
        }
        randomnum
        upper95 <- rep(NA, norep)
        lower95 <- rep(NA, norep)
        for (i in 1 : norep) {
            r <- randomnum[i]
            ftnupper <- function(p, r) {
                y <- -0.025
                dydp <- 0

                for (k in 0:r) {
                    y <- y + choose(20, k) * (p^k) * ((1 - p)^(20 - k))
                    dydp <- dydp + choose(20, k) * ((k * p ^ (k - 1) * (1 - p) ^
(20 - k)) - (p ^ k * (20 - k) * (1 - p) ^ (19 - k))) #nolint
                }

                return(c(y, dydp))
            }
            ftnlower <- function(p, r) {
                y <- -0.025
                dydp <- 0

                for (k in r:20) {
                    y <- y + choose(20, k) * (p^k) * ((1 - p)^(20 - k))
                    dydp <- dydp + choose(20, k) * ((k * p ^ (k - 1) * (1 - p) ^
(20 - k)) - (p ^ k * (20 - k) * (1 - p)^(20 - k))) #nolint
                }
```

```
      return(c(y, dydp))
    }
      root <- function(ftn, x0, tol, max_iter, random) {
        x <- x0
        y <- ftn(p = x, r = random)
        iter <- 0
        while ((abs(y[1]) > tol) && (iter < max_iter)) {
          x <- x - y[1] / y[2]
          y <- ftn(p = x, r = random)
          iter <- iter + 1
        }
        if (abs(y[1]) > tol) {
          return(NA)
        } else {
          return(x)
        }
      }
      lower95[i] <- root(ftn = ftnlower, x0 = 0.1, tol = 1e-9, max_iter = 1000, random = r)
      upper95[i] <- root(ftn = ftnupper, x0 = 0.3, tol = 1e-9, max_iter = 1000, random = r)
    }
    lower <- mean(lower95, na.rm = TRUE)
    upper <- mean(upper95, na.rm = TRUE)
    coverage <- mean((lower95 <= 0.15) & (upper95 >= 0.15), na.rm = TRUE)
    length <- mean(upper95 - lower95, na.rm = TRUE)
    (exact <- c(lower, upper, coverage, length))
```



```
> (exact <- c(lower, upper, coverage, length))
[1] 0.04048851 0.37381292 0.96884735 0.34251498
```

Lower bound : 0.04

Upper bound : 0.37

Coverage : 0.97

Length : 0.34