**# Homework4**

流行病學與生物統計計算

Homework 6

學號：b07401048

系級：醫學五

姓名：賴柏瑞

**#Q1 : exact 95% CI**

**# Pl**

*# exact 95% CI*

*# Pl : P(X = 0) + P(X = 1) + ...+ P(X = 19) = 0.975*

*# step1 : find initial value*

p <- **seq**(0, 1, by = 0.001)

**fp** <- function(p) {

y <- -0.975

for (k in 0:19) {

y <- y + **choose**(100, k) \* p^k \* (1 - p)^(100 - k)

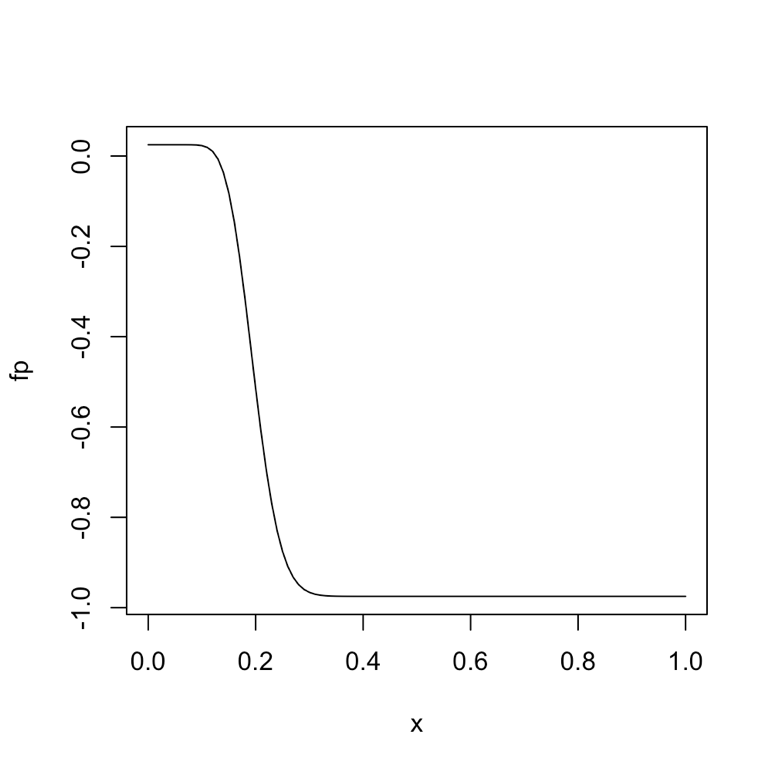
}

return(y)

}

**png**(filename = "hwk6.2.png", width = 1000, height = 1000, res = 200)

**plot**(fp)

**** **dev.off**()

*# step 2 : find f(p), f'(p)*

**ftn** <- function(p) {

y <- -0.975

dydp <- 0

for (k in 0:19) {

y <- y + **choose**(100, k) \* (p^k) \* ((1 - p)^(100 - k))

dydp <- dydp + **choose**(100, k) \* ((k \* p^(k - 1) \* (1 - p)^(100 - k))

- (p^k \* (100 - k) \* (1 - p)^(99 - k)))

}

return(**c**(y, dydp))

}

*# step 3 : Newton-Raphson function*

**root** <- function(ftn, x0, tol, max\_iter) {

x <- x0

y <- ftn(x)

iter <- 0

while ((**abs**(y[1]) > tol) && (iter < max\_iter)) {

x <- x - y[1] / y[2]

y <- ftn(x)

iter <- iter + 1

**cat**("at iteration", iter, "value of x is", x, "\n")

}

if (**abs**(y[1] > tol)) {

**cat**("algorithm failed to converge\n")

return(NULL)

} else {

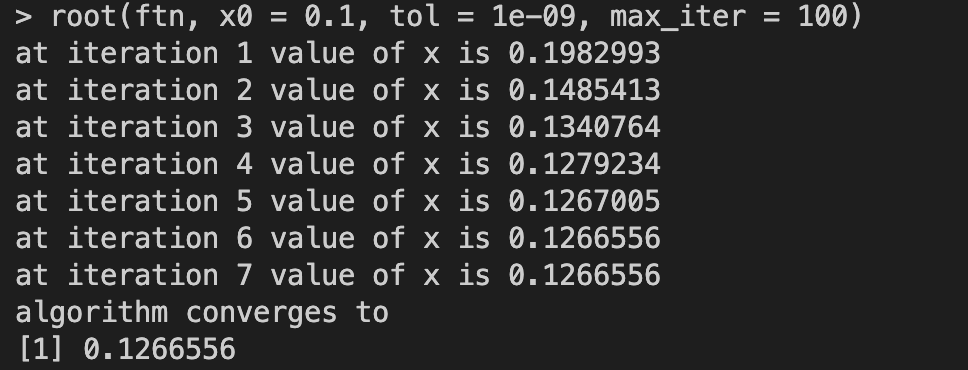
**cat**("algorithm converges to \n")

return(x)

}

}

root(ftn, x0 = 0.1, tol = 1e-09, max\_iter = 100)

 *# Pl = 0.1266556*

**# Pu**

*# Pu : P(X = 0) + P(X = 1) + ...+ P(X = 20) = 0.025*

*# find initial value*

p <- **seq**(0, 1, by = 0.001)

**fp** <- function(p) {

y <- -0.025

for (k in 0:20) {

y <- y + **choose**(100, k) \* p^k \* (1 - p)^(100 - k)

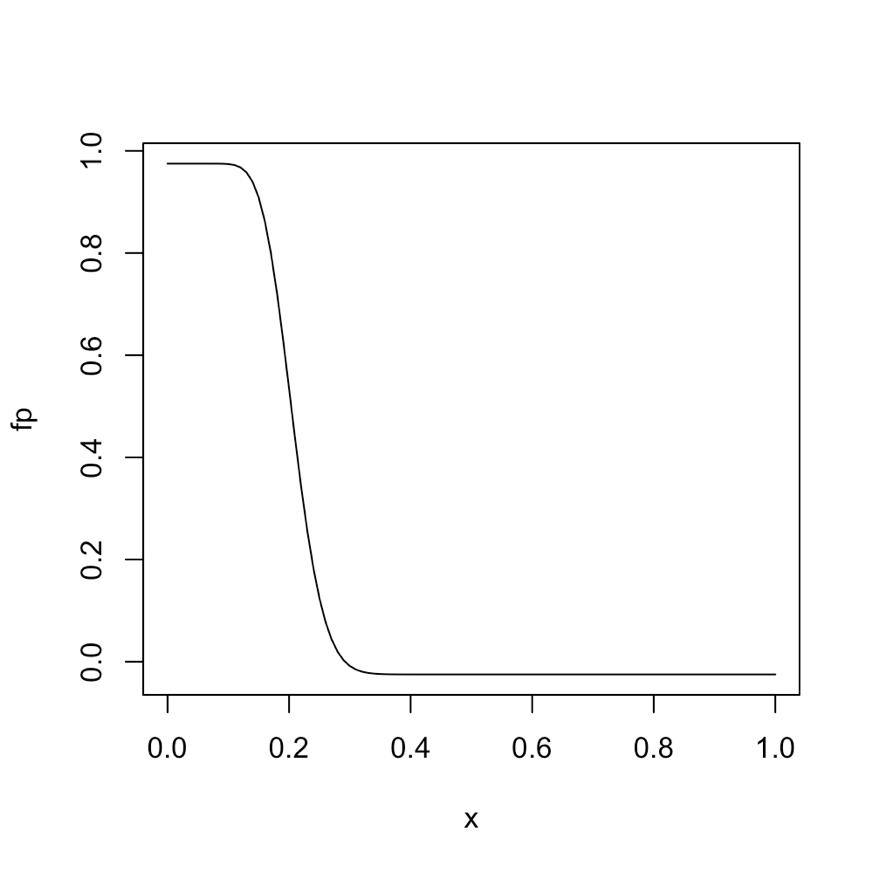
}

return(y)

}

**png**(filename = "hwk6.2.png", width = 1000, height = 1000, res = 200)

**plot**(fp)

**** **dev.off**()

*# step2 : find f(p), f'(p)*

**ftn** <- function(p) {

y <- -0.025

dydp <- 0

for (k in 0:20) {

y <- y + **choose**(100, k) \* (p^k) \* ((1 - p)^(100 - k))

dydp <- dydp + **choose**(100, k) \* ((k \* p^(k - 1) \* (1 - p)^(100 - k))

- (p^k \* (100 - k) \* (1 - p)^(99 - k)))

}

return(**c**(y, dydp))

}

*# step3 : Newton-Raphson function*

**root** <- function(ftn, x0, tol, max\_iter) {

x <- x0

y <- ftn(x)

iter <- 0

while ((**abs**(y[1]) > tol) && (iter < max\_iter)) {

x <- x - y[1] / y[2]

y <- ftn(x)

iter <- iter + 1

**cat**("at iteration", iter, "value of x is", x, "\n")

}

if (**abs**(y[1] > tol)) {

**cat**("algorithm failed to converge\n")

return(NULL)

} else {

**cat**("algorithm converges to \n")

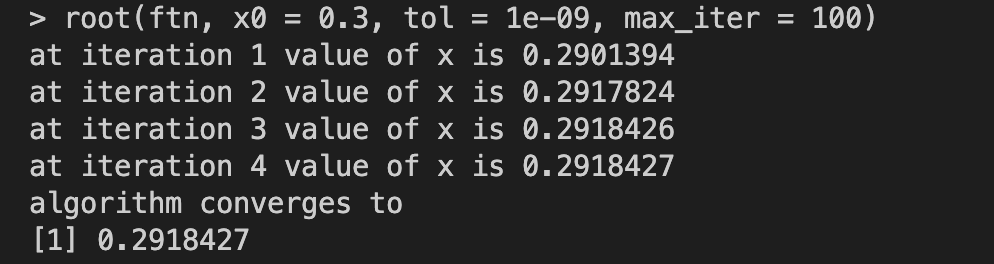
return(x)

}

}

root(ftn, x0 = 0.3, tol = 1e-09, max\_iter = 100)

*# Pu = 0.2918427*

**** *# exact 95%CI : 0.1266556-0.2918427*

**#Q1 : asymptotic 95% CI**

*# asymptotic 95% CI*

n <- 100

phat <- (20 / n)

pl <- phat - (**qnorm**(0.975) \* **sqrt**(((phat) \* (1 - phat)) / n))

pu <- phat + (**qnorm**(0.975) \* **sqrt**(((phat) \* (1 - phat)) / n))

(ci <- **c**(pl, pu))

*# asymptotic 95%CI : 0.1216014-0.2783986*