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

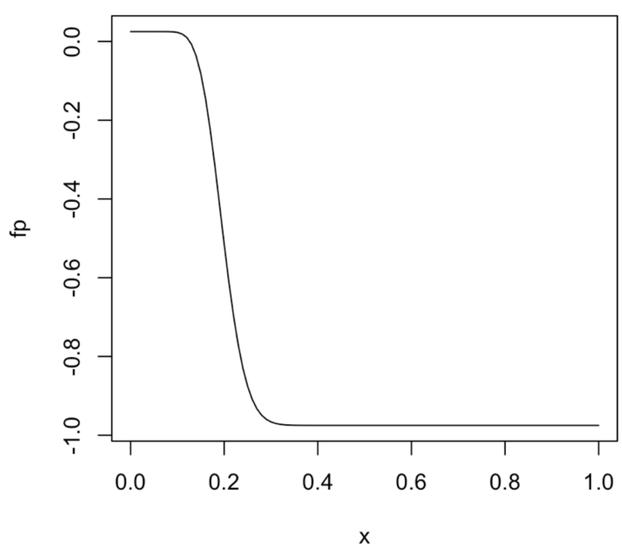
Homework4

#Q1 : exact 95% CI

Pl

```r
# 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()
```



```r
# 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))
}
```

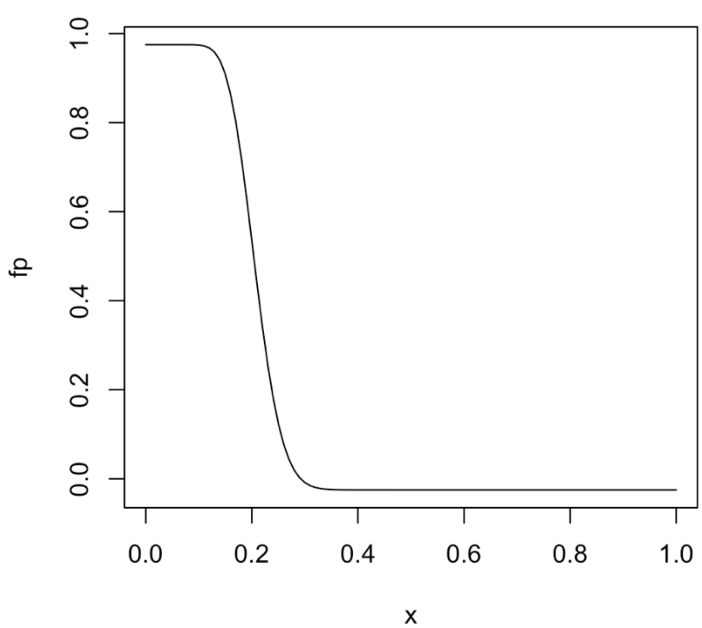
```r
# step 3 :  Newton-Raphson function
root <- function(ftn, x0, tol, max_iter) {
    x <- x0
    y <- ftn(x)
    iter <- 0
    while ((abs(y[1]) > tol) && (iter < max_iter)) {
      x <- x - y[1] / y[2]
      y <- ftn(x)
      iter <- iter + 1
      cat("at iteration", iter, "value of x is", x, "\n")
    }
    if (abs(y[1] > tol)) {
      cat("algorithm failed to converge\n")
      return(NULL)
    } else {
      cat("algorithm converges to \n")
      return(x)
    }
  }
```

```r
root(ftn, x0 = 0.1, tol = 1e-09, max_iter = 100)
   # Pl = 0.1266556
```

```
> root(ftn, x0 = 0.1, tol = 1e-09, max_iter = 100)
at iteration 1 value of x is 0.1982993
at iteration 2 value of x is 0.1485413
at iteration 3 value of x is 0.1340764
at iteration 4 value of x is 0.1279234
at iteration 5 value of x is 0.1267005
at iteration 6 value of x is 0.1266556
at iteration 7 value of x is 0.1266556
algorithm converges to
[1] 0.1266556
```

Pu

```r
# 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()
```



```r
# 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))
}
```

```r
# 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)
      }
   }
```

```r
root(ftn, x0 = 0.3, tol = 1e-09, max_iter = 100)
  # Pu = 0.2918427

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

```
> root(ftn, x0 = 0.3, tol = 1e-09, max_iter = 100)
at iteration 1 value of x is 0.2901394
at iteration 2 value of x is 0.2917824
at iteration 3 value of x is 0.2918426
at iteration 4 value of x is 0.2918427
algorithm converges to
[1] 0.2918427
```

#Q1 : asymptotic 95% CI

```r
# 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
```