Lab 1 - Computational Statistics (732A89)

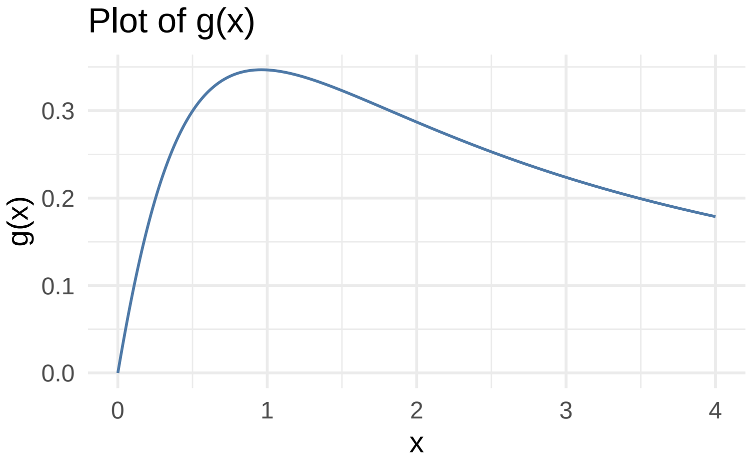
Helena Llorens Lluís (hllor282), Yi Yang (yiyan338)

QUESTION 1: Maximization of a function in one variable

The function

$$g(x) = \frac{log(x+1)}{x^{3/2}+1}$$

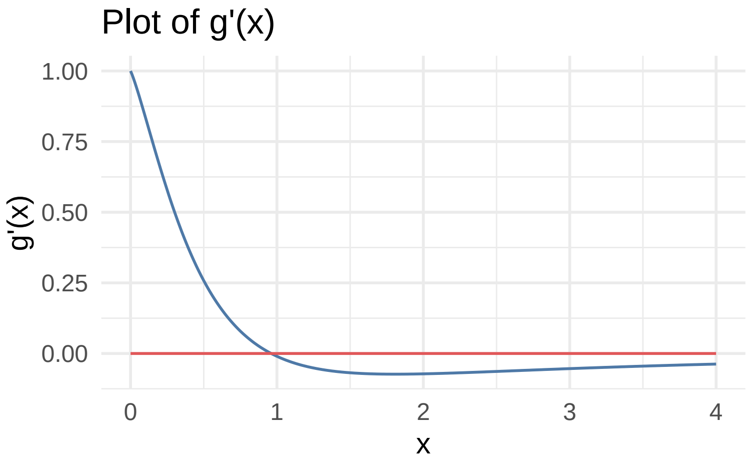
is represented in the interval $[0, 4]$ on the following plot. The maximum of this function seems to be at $x = 1$.



Then, we calculate

$$g'(x) = \frac{\frac{1}{x+1}(x^{3/2}+1) - log(x+1)\frac{3}{2}x^{1/2}}{(x^{3/2}+1)^2}$$

In the interval $[0, 4]$, $g'(x)$ is plotted as follows.



We have written a function for the bisection method for finding a local maximum for g, that takes as input $g'(x)$, the initial values for the interval a and b and a tolerance for a stopping criteria (set at 0.0001 as default).

We try this function for different intervals and observe the results. First, with $a = 0$ and $b = 10$, we get that the local maximum is at

```
##          x         it 
##  0.9610748 17.0000000
```

The bisection method converged reliably for the interval $[0, 10]$ because $g'(x)$ changes sign within this range. The method is robust but slower due to repeatedly halving the interval.

If we set the initial interval as $[2, 10]$, we get the following error:

```
## Error in bisection(f = dg, a = 2, b = 10): Signs of f(a) and f(b) have to differ
```

The algorithm requires a sign change within the interval to ensure convergence.

On the other hand, we have written a function for the secant method that takes as input $g'(x)$, a pair of starting values and a number of iterations (set to 1000 as default) and tolerance (set to 0.0001) as stopping criteria.

If we take as initial values $x_0 = 0$ and $x_1 = 1$, we get the following result.

```
##         x        it 
## 0.9610605 4.0000000
```

With starting points $x_0 = 0$ and $x_1 = 1$, the secant method converged rapidly to the correct maximum. This shows its efficiency when given good initial guesses.

However, if we set the starting values at $x_0 = 0$ and $x_1 = 2$, the result is as follows.

```
## Error in secant(f = dg, x0 = 0, x1 = 2): Division by zero detected. Method stopped.
```

In this case, the result diverged due to poor initial guesses and division by very small values in the denominator of the update formula.

Discussion

If we are aiming for robustness, the bisection method should be used. The method is guaranteed to converge if the sign change condition for $g'(x)$ is met within the interval. However, it may require more iterations compared to the secant method.

On the other hand, the secant method should be used for faster convergence, but only if reliable initial guesses are available.

For this function, Given the behavior of $g'(x)$, the bisection method is safer because it guarantees convergence when the sign change condition is met. However, the secant method is faster and works well if good initial guesses (close to the root) are provided.

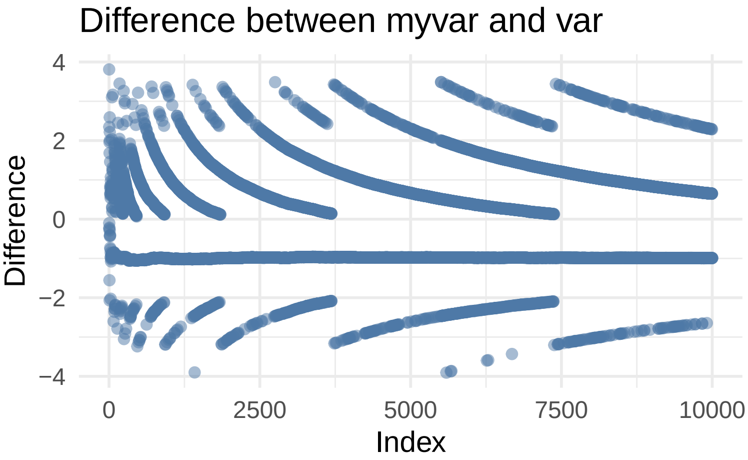
Question 2: Computer arthmetrics(variance)

We have written a function `myvar` for estimating the variance using the formula:

$$\mathrm{Var}(\vec{x}) = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right)$$

Then, using function `rnorm` to generate 10000 random numbers with mean 10^8 and variance 1.

For each subset , we compute the difference$Y_i = \mathrm{myvar}(X_i) - \mathrm{var}(X_i)$, and the plot of dependence Y_i on i shows as follows:



Conclusion

$$Y_i = \mathrm{myvar}(X_i) - \mathrm{var}(X_i)$$

varies from 3.67043 to -3.82699.The difference between `myvar` and `var` is relatively big. `myvar` doesn't perform well when the values are big.

Reasons

Computer arithmetics has limitations. Computers use floating-point numbers to represent real nembers.The precision of floating-point numbers is insufficient to represent differences within a small range.

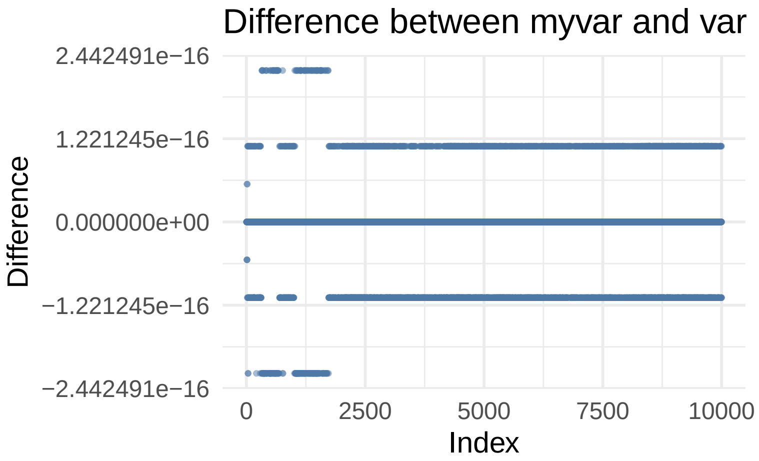
In this case, the values are very large(the mean of value is 10^8) and the variance is small(1),when calculating $\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2$,it may lead to computational errors.

Improvement of the variance estimator

To avoid calculating $\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2$, we use the following formula instead:

$$\operatorname{var}(x) = \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)$$

Then we compute the difference $Y_i = \operatorname{myvar}(X_i) - \operatorname{var}(X_i)$, and the plot of dependence Y_i on i shows as follows:



The improved variance estimator works better.It gives the same results as `var()`, the `improved_Yi` is relatively small and can be ignored.

Appendix

Question 1

```r
library(ggplot2)
# g(x) function
g <- function(x) {
  log(x + 1) / (x^(3/2) + 1)
}

#plot of g(x)
x <- seq(0, 4, by = 0.01)
y <- sapply(x, g) # apply g(x) to the vector x
data <- data.frame(x_vals = x, y_vals = y)
ggplot(data, aes(x = x_vals, y = y_vals)) +
  geom_line(col = "#4E79A7") +
  labs(title = "Plot of g(x)", x = "x", y = "g(x)") +
  theme_minimal()

# g'(x) function
dg <- function(x){
  a <- (1/(x+1)*(x^(3/2) + 1) - log(x+1)*((3/2)*x^(1/2)))
  b <- (x^(3/2) + 1)^2
  return(a/b)
}

# plot of g'(x)
x_dg <- seq(0, 4, by = 0.01)
y_dg <- sapply(x_dg, dg) # apply g'(x) to the vector x_dg
data_dg <- data.frame(x_vals = x_dg, y_vals = y_dg)
ggplot(data_dg, aes(x = x_vals, y = y_vals)) +
  geom_line(col = "#4E79A7") +
  geom_line(y = 0, col = "#E15759") +
  labs(title = "Plot of g'(x)", x = "x", y = "g'(x)") +
  theme_minimal()

# bisection algorithm function
bisection <- function(f, a, b, maxit = 1000, tol = 0.0001){

  # check that the assumptions of the algorithm are met
  if(sign(f(a) * f(b)) != sign(-1)){
    stop("Signs of f(a) and f(b) have to differ")
  }

  for(it in 1:maxit){
    # find middle point
    c <- (a + b)/2

    # if stoping critera is met, return the solution
    if((b-a)/2 < tol){
      return(c(x = c, it = round(it)))
    }

    # update interval
    if(sign(f(c)) == sign(f(a))){
      a <- c
    } else{
      b <- c
    }
  }

  warning("Maximum number of iterations reached without convergence.")
  return(c(x = (a + b)/2), it = round(it))
}

bisection(f = dg, a = 0, b = 10)
bisection(f = dg, a = 2, b = 10)

secant <- function(f, x0, x1, maxit = 1000, tol = 0.0001){
  for(it in 1:maxit){

    # check that the division can be calculated
    denom <- (f(x1)-f(x0))/(x1-x0)
    if(denom == 0){
```

```r
      stop("Division by zero detected. Method stopped.")
    }

    # update x2
    x2 <- x1 - f(x1)/denom

    # if stopping criteria is met, return the solution
    if(abs(x2 - x1) < tol*abs(x2)){
      return(c(x = x2, it = round(it)))
    }

    x0 <- x1
    x1 <- x2
  }

  warning("Maximum number of iterations reached without convergence.")
  return(c(x = x2, it = round(it)))
}

secant(f = dg, x0 = 0, x1 = 1)
secant(f = dg, x0 = 0, x1 = 2)
```

Question 2

```r
#a.myvar
myvar <- function(x){
  n <- length(x)
  sum_x2 <- sum(x^2)
  sum_x_squared <- ((sum(x))^2)/n
  return((sum_x2-sum_x_squared)/(n-1))
}
#b.generate random numbers
x_random <- rnorm(10000,10^8,1)
#calculate Yi
Yi <- numeric(length (x_random))

for (i in 1:length(x_random) ) {
  Xi <- x_random[1:i]
  Yi[i] <- myvar(Xi)-var(Xi)
}
#plot
i <- c(1:length(x_random))
# plot(i,Yi,type="p",col="blue",
#      main = "Difference between myvar and var",
#      xlab = "i",ylab = "Yi")
# grid()
plot_data <- cbind(i, Yi)
ggplot(plot_data, aes(x = i, y = Yi)) +
  geom_point(col = "#4E79A7",alpha = 0.5) +
  labs(title = "Difference between myvar and var", x = "Index", y = "Difference") +
  theme_minimal()
```

```r
#d.improved variance estimator

improved_myvar <- function(x){

  mean_x <- mean(x)
  sum_squared_difference <- sum((x-mean_x)^2)
  n <- length(x)
  return(sum_squared_difference/(n-1))

}
improved_Yi <- numeric(length (x_random))

for (i in 1:length(x_random) ) {
  Xi <- x_random[1:i]
  improved_Yi[i] <- improved_myvar(Xi)-var(Xi)
}
#plot
i <- c(1:length(x_random))
# plot(i,improved_Yi,type="p",col="blue",
#      main = "Difference between myvar and var",
#      xlab = "i",ylab = "improved_Yi")
# grid()

plot_data <- cbind(i, improved_Yi)
ggplot(plot_data, aes(x = i, y = improved_Yi)) +
  geom_point(col = "#4E79A7",alpha = 0.5) +
  labs(title = "Difference between myvar and var", x = "Index", y = "Difference") +
  theme_minimal()
```