# *R* computing for Business Data Analytics

Homework 2

Due date: 23:59PM November 6 (Sunday), 2016

Please e-mail your homework (.pdf) and the associated *R* code (.*R*) to hchuang.om@gmail.com.

The email title must be R_HW2_GroupName. Each group submits ONLY ONE copy.

NO late homework will be accepted.

**Q1.** (15%) The Fibonacci sequence is famous in mathematics. The sequence is defined as

$$F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2} \text{ for } n > 2$$

(a) Write a *while*( ) loop to find the first Fibonacci number $k$ > greater than 100.

(b) For the number $F_n$=$k$ in (a), what is the index $n$?

(c) Use *for*( ) to write a function *print.Fib* that takes an integer $k$ as its input and prints **ALL** Fibonacci numbers <= $k$. Test $k$=100 and show me the results.

**Q2.** (10%) Write a function *second.smallest*( ) that takes a vector x as its input. The function will return the number that is the second smallest (第二小) inside x.

Show me the results of *second.small*(x=c(2, 8, 8, 2, 5, 2, 5, 2)).

**Q3.** (10%) Use *while*( ) and/or *if…else* to write a function *f.exist* that takes an *integer* z and a *vector* x as its inputs. The function *f.exist* will return TRUE only if z is inside x.

Test f.exist(z=10, x=c(1:10)) and f.exist(z=10, x=c(9, 3, 1)). Show the answers.

**Q4.** (10%) Use *while*( ) and/or *if…else* to write a function *f.divide* that takes an *integer* z as its input. The function *f.divide* will return how many divisors (除數) z has (other than 1 & z itself).

Test *f.divide*(100) and show me the results.

**Q5.** (10%) Write a function *UNIQUE*( ) that takes a vector x as its input. The function will return a new vector with all unique (獨特的) numbers in x with duplicated (重複) elements removed.

Do NOT use *unique*( ) in *R*. Show me the results of UNIQUE(x=c(2, 8, 8, 2, 5, 2, 5, 2)).

**Q6.** (20%) The Babylonian method (巴比倫法) is famous for getting the square root (平方根) of any number. Suppose we have a positive number *S*, the Babylonian method suggests that

$$x_{n+1} = 0.5(x_n + S / x_n)$$

$x_n$ is your current guess of $\sqrt{S}$ and $x_{n+1}$ is your next guess of $\sqrt{S}$ . You will STOP searching only if | $x_{n+1}$ - $x_n$ | < tolerance.

Now, set *S*=125348, your initial guess $x_0$=600, and tolerance=1e-5.

Use *while*( ) to implement the algorithm and show me the square root ( $\sqrt{S}$ ) you find. The answer should be 354.0452.

How about S=9527, initial guess $x_0$=87, and tolerance=1e-5? What's the answer?

How about S=5566, initial guess $x_0$=78, and tolerance=1e-5? What's the answer?

**Q7.** (25%) Write a function *BesselI_Gen* that has five arguments (a, v, z, max, tolerance). The function will compute the **generalized** modified Bessel function of the first kind:

$$I_a^{\nu}(z) = \sum_{m=0}^{\infty} \frac{1}{\left[\Gamma(m+a+1)m!\right]^{\nu}} \left(\frac{z}{2}\right)^{2m+a} \quad \text{where } \Gamma(\bullet) \text{ is the } gamma(\ ) \text{ in } R.$$

As infinite sum is NOT possible in computers, please mimic the example in lecture 2 to compute the value of the function. Also, the *BesselI_Gen* function will continue to add up numbers only if

$$m < \max \ \& \ \left| I_a^{\nu}(z)_{m-1} - I_a^{\nu}(z)_m \right| > \text{tolerance}$$

Once you finish coding, run *BesselI_Gen*(5, 1, 10, 1000, 1e-5) in *R*. What is the value? Then run besselI(10, 5) (a built-in function for the modified Bessel function of the first kind) in *R*. Are the two values identical?