Lab1

Soyeon Lee

8/31/2020

1. Compute the sum of the sequence 100 to 1000, going up by a constant value of 100 (100,200,300,400,500,600,700,800,900,1

```r
# sum of the numbers in a sequence from 100 to 1000 with an interval of 100
sum(seq(100,1000,100))
```

```
## [1] 5500
```

```r
#a series of numbers using the seq() are assigned to a
#the_sum is initialized
#at each iteration, the_sum is updated
#the final value for the_sum

a<-seq(100,1000,100)
the_sum<-0

for(i in a){
  the_sum<-the_sum+i
}
the_sum
```

```
## [1] 5500
```

2. Compute the sum of these numbers (1,3,2,4,3,5,4,3,4,5,6,5,6,7,6,5,6,5,4,3,4,5)

```r
#adding all the listed numbers
sum(c(1,3,2,4,3,5,4,3,4,5,6,5,6,7,6,5,6,5,4,3,4,5))
```

```
## [1] 96
```

3. Write a custom sequence generator function using a for loop that generates a sequence from a starting integer value to an ending integer value in steps of 1. Demonstrate that it can produce the sequence 1 to 10.

```r
#assign function() to the seq_generator function
#conditional for loop prints out a sequence of numbers from 1 to 10
#the loop continues as i is updated
seq_generator<-function(first,last){
  for(i in first:last){
    print(i)
```

```
      i<-i+1
  }
}
seq_generator (first=1,last=10)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```
#assign function()to a new function seq_gen
#creates a null vector called sequence set at null
#grows vector content as j goes through start:end
seq_gen<-function(start,end){
  sequence<-c()

  for(j in start:end){
    sequence[j]<-j
  }
  return(sequence)
}

seq_gen(1,10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

4. Write a custom function to implement the following general equation to find the sum of any constant
   series: Gauss formula (#deleted the formula because the m-dash character and three dots tried to mess
   me up) Demonstrate that your function correctly produces the sum for the series below: seq(10,100,10)
   #> [1] 10 20 30 40 50 60 70 80 90 100

```
#Creates a function called gauss_meth
#calculate two components of the formula separately
#multiply the components and return value
gauss_meth<-function(from,to,by){
  num_nums<-(to-from)/by +1
  sum_unit<-(from+to)/2
  temp<-num_nums*sum_unit
  return(temp)
}
gauss_meth(from=10,to=100,by=10)
```

```
## [1] 550
```

```
#second option
#gauss's formula is calculated as the total_sum
#the total_sum is the value being returned
gauss_meth<-function(from,to,by){
  total_sum<-((to-from)/by +1)*((from+to)/2)
  return(total_sum)
}
gauss_meth(from=10,to=100,by=10)
```

[1] 550

Demonstrate that your function correctly produces the sum for the series below:

seq(10,100,10) #> [1] 10 20 30 40 50 60 70 80 90 100

5. Write a custom function that generates a constant series between any start and end values, with any constant, and finds the sum. Have your function output both the sequence and the sum. For this problem, feel free to use the existing seq() and sum() functions in your custom function. Demonstrate the function correctly prints out the above sequence (10 to 100 in steps of 10), and its sum.

```
#the function is assigned to the new function
#the seq() function outcome is assigned to a new variable
#the outcome of sum(sequence)is assigned to the_sum variable
#the outcome of sequence and the_sum are assigned to a new variable my_output
#return output

seq_n_sum<-function(first,last,steps){
  sequence<-seq(first,last,steps)
  the_sum<-sum(sequence)

  my_output<-list(sequence,the_sum)
  return(my_output)
}
seq_n_sum(10,100,10)
```

[[1]]
[1] 10 20 30 40 50 60 70 80 90 100
##
[[2]]
[1] 550

```
#second option using the previous function, gauss_meth()
#seq()function is assigned to the_seq variable
#gauss_meth()function is assigned to the_sum variable
#the list()with the_seq and the_sum as arguments are assigned to the_outcome
#return the_outcome

seq_sum<-function(from,to,by){

  the_outcome<-list(the_seq<-seq(from,to,by),
                    the_sum<-gauss_meth(from,to,by))
  return(the_outcome)
}
seq_sum(10,100,10)
```

```
## [[1]]
##  [1]  10  20  30  40  50  60  70  80  90 100
##
## [[2]]
## [1] 550
```

6. Use the sum() and the length() functions to calculate the mean (average) of the numbers x = c(1,2,3,4,5).

```r
#create a vector x
#initialize the_sum and the_length
#at each iteration, the_sum and the_length are updated
#the average is calculated using the final sum and the length

x<-c(1,2,3,4,5)
the_sum<-0
the_length<-0

for(i in x){
  the_sum<-the_sum+i
  the_length<-the_length+1
  average<-the_sum/the_length
}
print(average)
```

```
## [1] 3
```

```r
#second option
#a combined set of numbers are put into a
#the sum is assigned to s, and the length is assigned to l
#s is divided by l to obtain average

a<-(c(1,2,3,4,5))

s<-sum(a)
l<-length(a)
average<-s/l

print(average)
```

```
## [1] 3
```