Functions for a personal R-package

Yunting Chiu

9/18/2020

```
library(ggplot2)
library(tibble)
```

function 1

```
myseq_n <- function(x, n){
  stopifnot(length(x) == 3 & is.numeric(x)) # error check x
  stopifnot(n > 0 & as.integer(n)) # error check n

  nums <- vector(mode = "integer", length = n) # create nums's type
  for (i in seq_along(nums)) {
    if (i <= 3) {
      nums[i] <- x[i]
    }else{
      nums[i] <- nums[i - 1] + (nums[i - 3] - nums[i - 2]) / i
    }
  }
  return(nums[n])
} # created myseq_n function

# Test function
myseq_n(x = c(2, 3, 3), n = 3)
```

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

```
myseq_n(x = c(2, 4, 3), n = 4)
```

```
## [1] 2.5
```

```
myseq_n(x = c(2, 4, 3), n = 5)
```

```
## [1] 2.7
```

```
myseq_n(x = c(2, 4, 3), n = 6)
```

```
## [1] 2.783333
```

```
myseq_n(x = c(2, 4, 3), n = 7)
```

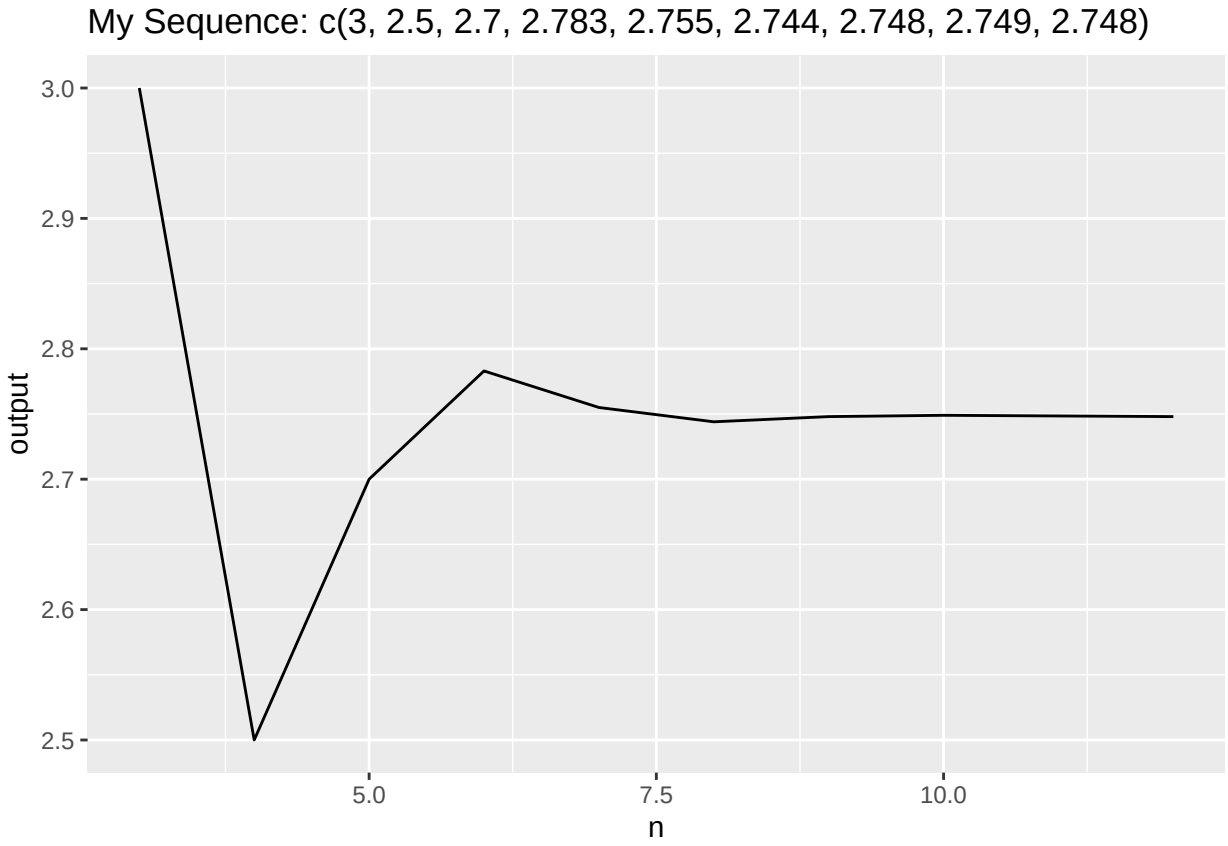
```
## [1] 2.754762
```

```
#calcn(x = c(2, 4, 3), n = 7L) # correct x and n type
#calcn(x = c(2, 4), n = 7) # length of x < 3
#calcn(x = c(2, 4, 3, 5), n = 6) # length of x > 3
```

```
#calcn(x = c(2, 4, 3), n = 0) # n <= 0
#calcn(x = c(2, 4, 3), n = 5.53) # n is a float(dbl)
```

function 2

```
numSeqPlot <- function(nums){
  stopifnot(ncol(nums) == 4 & as_tibble(nums)) # error check the length and df type
  stopifnot(all(nums[[4]] > 0) & is.numeric(nums[[1]]) & is.numeric(nums[[2]]),
            is.numeric(nums[[3]]) & as.integer(nums[[4]]))
  # error check the fourth column is a positive integer, and all numbers are numeric

  df <- tibble(n = 0, output = 0) # build a blank data frame
  nums <- tibble(nums) # named nums as tibble
  for (i in 1:nrow(nums)) {
    x <- c(nums[[i, 1]], nums[[i, 2]], nums[[i, 3]]) # extract x, y, z
    n <- nums[[i, 4]] # extract n
    myseq_n(x, n) -> df[i, 2] # the second column
    n -> df[i, 1] #the first column

  }

  df[ ,2] <- round(df[ ,2],digits = 3) # set the rounding numbers will store in ggtitle later
  graphic <- df %>% # make a graphic
    ggplot2::ggplot(mapping = ggplot2::aes(x = n, y = output)) +
    ggplot2::geom_line() +
    ggplot2::labs(title = paste("My Sequence:", df[ ,2]))

  return(graphic) # created numSeqPlot
}

# testing the function
my_data <- tribble(
  ~x, ~y, ~z, ~n,
  2,4,3,3,
  2,4,3,4,
  2,4,3,5,
  2,4,3,6,
  2,4,3,7,
  2,4,3,8,
  2,4,3,9,
  2,4,3,10,
  2,4,3,12)

numSeqPlot(my_data)
```


My Sequence: c(3, 2.5, 2.7, 2.783, 2.755, 2.744, 2.748, 2.749, 2.748)
output
3.0
2.9
2.8
2.7
2.6
2.5
5.0
7.5
10.0
n