Intermediate R Assignment 2

Katie Beidler

January 25, 2015

1. **The sum of a sequence. Write a for loop that will produce a vector "y" that contains the sum of x up to that index of x.**

For each iteration in the for loop, x is being summed up to the next number in the sequence. In other words, for the sequence 1 through 10- 1 is added to 2, making the next number in the sequence 3 wich is added to 3, and the sum is added to 4 and so on to produce the output of the for loop- y.

```
x = 1:10
y = NULL
for(i in 1:length(x)) {
    y [i] = sum(x[1:i])
}
x

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

y

##  [1]  1  3  6 10 15 21 28 36 45 55
```

2. **Modify your for loop so that if the sum is great than 10 the value of y is set to NA**

Now the for loop should spit out NA for the last 6 values of y.

```
x = 1:10
y = NULL
for(i in 1:length(x)) {
    y [i] = sum(x[1:i])
    if(y [i] > 10) {
      print('NA')
    }
    else {
        print (y[i])
      }
}

## [1] 1
## [1] 3
## [1] 6
## [1] 10
## [1] "NA"
## [1] "NA"
## [1] "NA"
## [1] "NA"
## [1] "NA"
## [1] "NA"
```

3. **Place your for loop into a function that accepts as its argument any vector of arbitrary length and it will return y.**

The function, eval_y allows you to enter a vector (x) that contains values for whatever length of x you specify in the function. See the tests for the following input: eval_y (1:15) and eval_y(1:234).

```
eval_y = function(x) {
  y = NULL
for(i in 1:length(x)) {
    y [i] = sum(x[1:i])
    }
    return(y)
  }
## Now to test the function
eval_y(1:15)
```

```
##  [1]   1   3   6  10  15  21  28  36  45  55  66  78  91 105 120
```

```
eval_y(1:234)
```

```
##   [1]     1     3     6    10    15    21    28    36    45    55    66
##  [12]    78    91   105   120   136   153   171   190   210   231   253
##  [23]   276   300   325   351   378   406   435   465   496   528   561
##  [34]   595   630   666   703   741   780   820   861   903   946   990
##  [45]  1035  1081  1128  1176  1225  1275  1326  1378  1431  1485  1540
##  [56]  1596  1653  1711  1770  1830  1891  1953  2016  2080  2145  2211
##  [67]  2278  2346  2415  2485  2556  2628  2701  2775  2850  2926  3003
##  [78]  3081  3160  3240  3321  3403  3486  3570  3655  3741  3828  3916
##  [89]  4005  4095  4186  4278  4371  4465  4560  4656  4753  4851  4950
## [100]  5050  5151  5253  5356  5460  5565  5671  5778  5886  5995  6105
## [111]  6216  6328  6441  6555  6670  6786  6903  7021  7140  7260  7381
## [122]  7503  7626  7750  7875  8001  8128  8256  8385  8515  8646  8778
## [133]  8911  9045  9180  9316  9453  9591  9730  9870 10011 10153 10296
## [144] 10440 10585 10731 10878 11026 11175 11325 11476 11628 11781 11935
## [155] 12090 12246 12403 12561 12720 12880 13041 13203 13366 13530 13695
## [166] 13861 14028 14196 14365 14535 14706 14878 15051 15225 15400 15576
## [177] 15753 15931 16110 16290 16471 16653 16836 17020 17205 17391 17578
## [188] 17766 17955 18145 18336 18528 18721 18915 19110 19306 19503 19701
## [199] 19900 20100 20301 20503 20706 20910 21115 21321 21528 21736 21945
## [210] 22155 22366 22578 22791 23005 23220 23436 23653 23871 24090 24310
## [221] 24531 24753 24976 25200 25425 25651 25878 26106 26335 26565 26796
## [232] 27028 27261 27495
```