Hailstone Sequence

The purpose of this exercise is to write the user-defined **R** function `hailstoneseq.m` that takes as an input a positive integer N, generates values in the recurrence relation $f(k)$, where

$$f(k) = \left\{ \begin{array}{ll} k & \text{if } k = 1 \\ 3f(k-1)+1 & \text{if } k \text{ is odd} \\ \frac{f(k-1)}{2} & \text{if } f(k-1) \text{ is even} \end{array} \right\}$$

until $f(k) = 1$, and plots and returns the vector $(f(1), f(2), \ldots, f(k))$.

In words, here's how the sequence is generated:

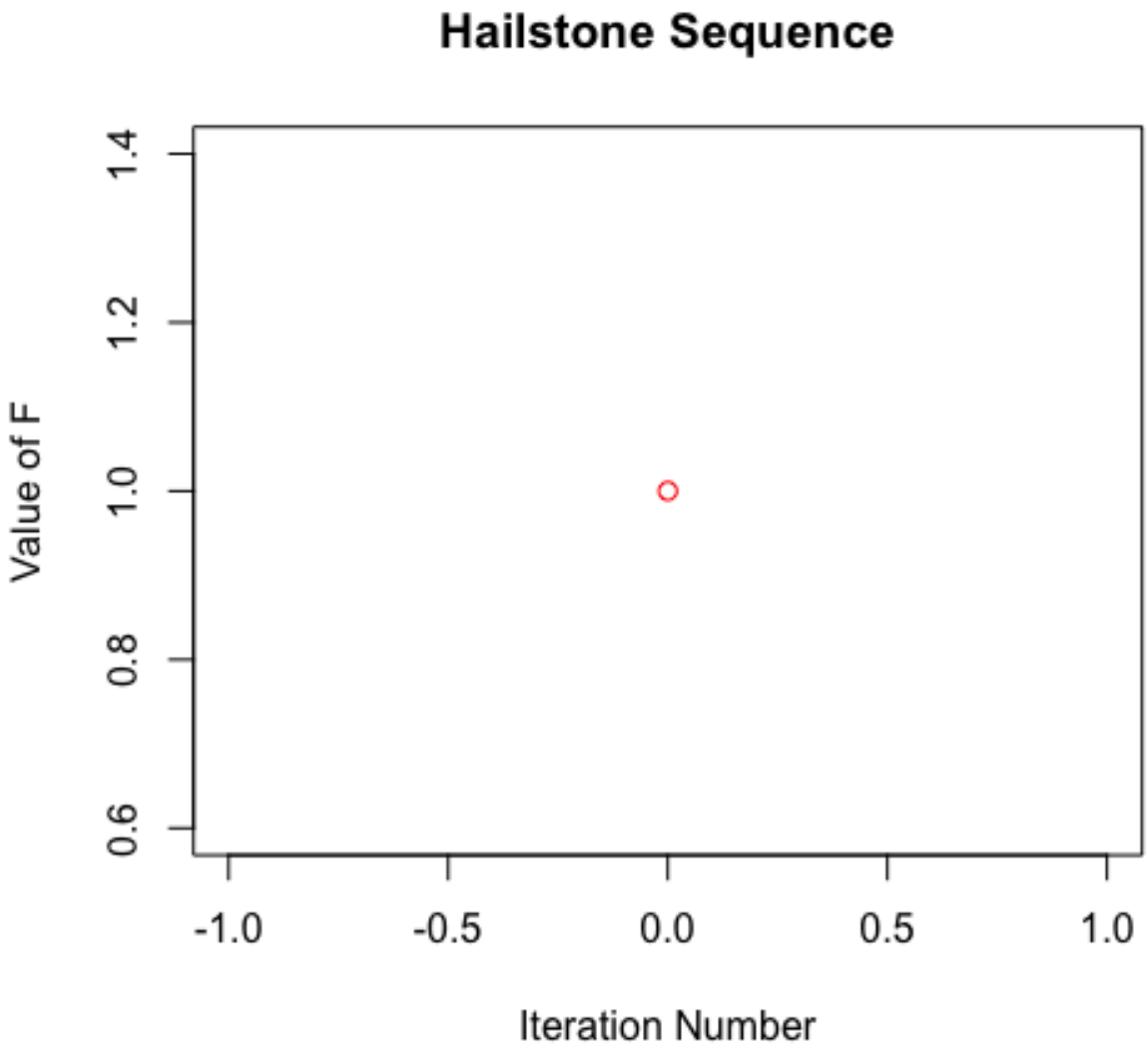
> Take any natural number N. If N is even, divide it by 2 to get $N/2$. If N is odd, multiply it by 3 and add 1. Repeat the process until the number equals 1.

The sequence that is generated is known as the "Hailstone Sequence," and the "Collatz Conjecture" is that no matter what N is, eventually, you will end up with 1. It has not been proven for all positive integers N. The "Collatz Graph" pictured on the last page shows all starting values N that reach 1 in twenty iterations or fewer.

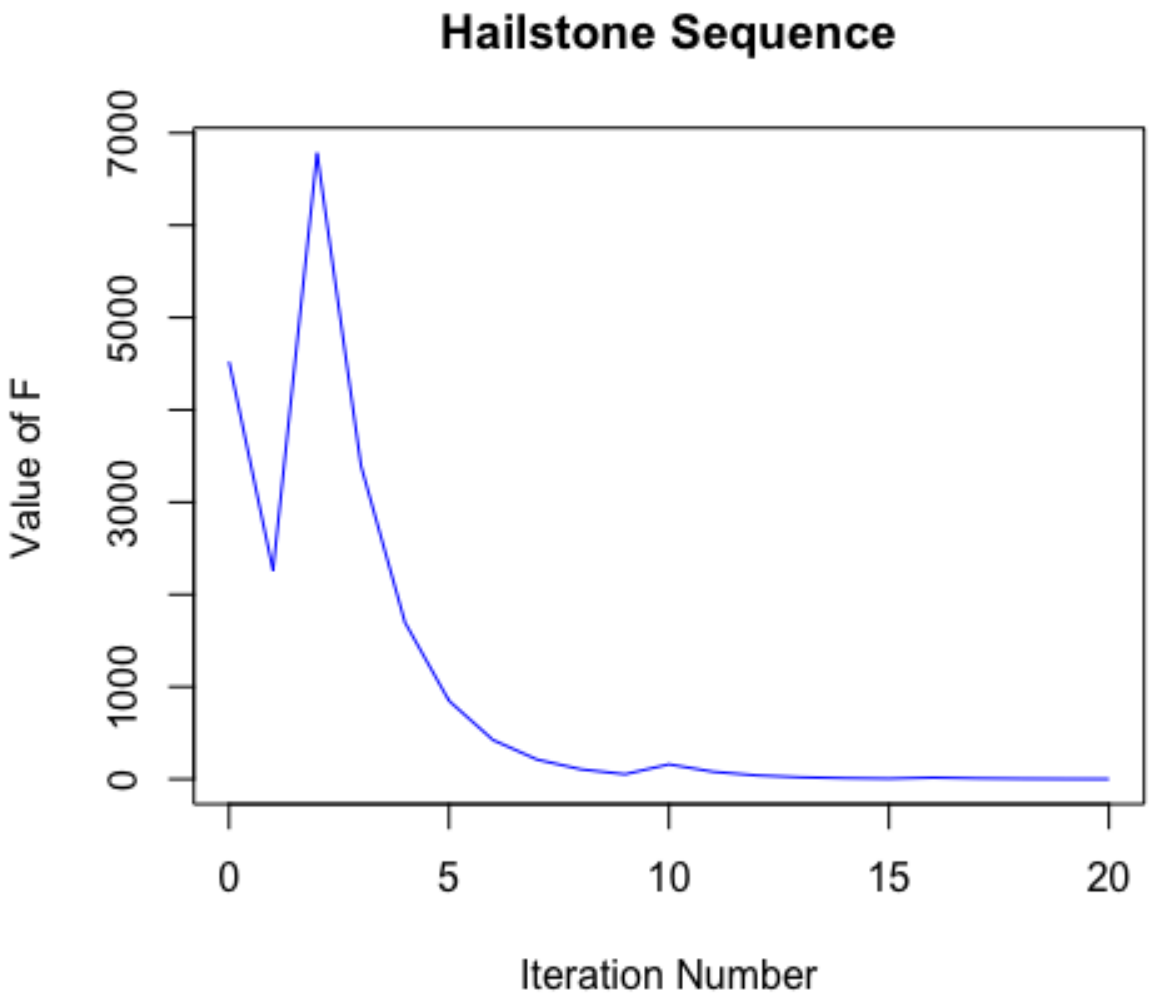
The function should give an error message and return NA when the user inadvertently enters a value of N that is not a positive integer. The plot of the hailstone sequence should begin at 0, not 1. Note that there is a special case when $n = 1$.

Here are several examples and their associated plots.

```
> hailstone(0)
Error, N must be a natural number.
[1] NA
> hailstone(5.6)
Error, N must be a natural number.
[1] NA
> hailstone(-3)
Error, N must be a natural number.
[1] NA
> hailstone(-8.1)
Error, N must be a natural number.
[1] NA
> hailstone(-1)
Error, N must be a natural number.
[1] NA
> hailstone(1)
[1] 1
```


Hailstone Sequence
Value of F
1.4
1.2
1.0
0.8
0.6
-1.0
-0.5
0.0
0.5
1.0
Iteration Number

```
> hailstone(4522)
 [1] 4522 2261 6784 3392 1696  848  424  212  106   53  160   80   40
[14]   20   10    5   16    8    4    2    1
```



```
> hailstone(2018)
  [1] 2018 1009 3028 1514  757 2272 1136  568  284  142   71  214  107
 [14]  322  161  484  242  121  364  182   91  274  137  412  206  103
 [27]  310  155  466  233  700  350  175  526  263  790  395 1186  593
 [40] 1780  890  445 1336  668  334  167  502  251  754  377 1132  566
 [53]  283  850  425 1276  638  319  958  479 1438  719 2158 1079 3238
 [66] 1619 4858 2429 7288 3644 1822  911 2734 1367 4102 2051 6154 3077
 [79] 9232 4616 2308 1154  577 1732  866  433 1300  650  325  976  488
 [92]  244  122   61  184   92   46   23   70   35  106   53  160   80
[105]   40   20   10    5   16    8    4    2    1
```