Fix my mistakes - sequentially.shortened.sequence function

David Zeleny

I wrote a bit silly R function `sequentially.shortened.sequence`, which creates a sequentially shortened sentence, and prints it on screen. However, I was a bit in a hurry and did not really tidy up my R-code, so it may not work as expected. Would you please fix it up so that it works?

Note that I ask you to fix only the mistakes/mistypes, not to considerably rewrite and modify the script.

```
# Define the function:
sequentially.shortened.sentence (sentence == "No need to have R depresssion today,
things go perfectly smoothly!')
{
  for (i inn nchar (Sentence):1)
    print (substr (sentence, start = 1. stop = i))
}}

# Run the function with default values:
sequential1y.shortened,sentence {}

# Run the function with another sentence:
sequentiallly.sh0rtened.sentenCe (sentencece = 'Here we go! Go - where?'#)
```

When the code is correct, the first run of the function (with default values) will return this:

```
[1] "No need to have R depresssion today, things go perfectly smoothly!"
[1] "No need to have R depresssion today, things go perfectly smoothly"
[1] "No need to have R depresssion today, things go perfectly smoothl"
[1] "No need to have R depresssion today, things go perfectly smooth"
[1] "No need to have R depresssion today, things go perfectly smoot"
....
....
....
[1] "No ne"
[1] "No n"
[1] "No "
[1] "No"
[1] "N"
```

The second run, with alternative text, will return:

```
[1] "Here we go! Go - where?"
[1] "Here we go! Go - where"
[1] "Here we go! Go - wher"
[1] "Here we go! Go - whe"
[1] "Here we go! Go - wh"
....
....
....
[1] "Here "
[1] "Here"
[1] "Her"
[1] "He"
[1] "H"
```

Hints (try first to solve the exercise without using them):

1. Check the script first - is it really a definition of the function which should do all the job? And does it have too many or too few parentheses?
2. The function `nchar` counts the number of characters in the character string (e.g., `nchar ("Hello Doly!")` will return `11`.
3. The function `substr` subsets the character string (e.g., `substr ("Hello Doly!", start = 4, stop = 8)` will return `lo Do`).