Homework 6: Bug Hunt

36-350

Due at 11:59 pm on Thursday, 23 October 2014

In this assignment you will debug each of the functions included in the accompanying R script so that they produce the correct results.

```
source("hw-06-supplement.R")
```

You will replace the above sourcing statement with your corrected code so that when you mark down the blocks below to execute their R code, you will have the desired output.

1. `my.regexpr.1()` should produce full names.

```
my.regexpr.1 ("Cosma Shalizi, Andrew Thomas, Samuel L. Ventura, Bryan Hooi") ## Should be obvious.
my.regexpr.1 (theIncredibles)
## full names: Helen Parr, Lucius Best.
## Optional: Mr. Incredible. Bonus points:
##    add the space back and exclude "Nomanisan Island" from the search.
```

2. `my.regexpr.2()` should extract dates from a string in a known pattern. Test on this sequence to extract the two dates.

```
my.regexpr.2 (terminator2)
```

3. `my.dgamma.log()` generates the log of the probability density of the Gamma distribution. Note that this is exactly equal to the function `dgamma(..., log=TRUE)` so use this to verify your answers.

```
my.dgamma.log (seq(0.2, 5, by=0.2), shape=0.2, rate=5)
## dgamma (seq(0.2, 5, by=0.2), shape=0.2, rate=5, log=TRUE)
my.dgamma.log (seq(0.4, 10, by=0.4), shape=0.2, rate=5)
my.dgamma.log (seq(1, 10, by=0.2), shape=2, rate=6)
```

4. `my.dnorm.log()` generates the log of the probability density of the normal distribution. Note that this is exactly equal to the function `dnorm(..., log=TRUE)` so use this to verify your answers.

```
my.dnorm.log (seq(-3, 3, by=0.2), mean=0, sd=1)
## dnorm (seq(-3, 3, by=0.2), mean=0, sd=1, log=TRUE)
my.dnorm.log (seq(-3, 3, by=0.2), mean=-1, sd=2)
my.dnorm.log (seq(-3, 3, by=0.2), mean=-2, sd=5)
```

5. Find the nearest zero of a function using the Newton-Raphson method [http://en.wikipedia.org/wiki/Newton's_method], coded in `zero.finder()`. HINT: add extra intermediate outputs to diagnose what the function is doing.

```
zero.finder (function(x) {1 - x^2}, 1.5)  # should be 1
zero.finder (function(x) {1 - x^2}, 0.5)  # should be 1
zero.finder (function(x) {x^3 - 7*x^2 + 7*x + 15}, 3.5)  # should be 3
zero.finder (function(x) {x^3 - 7*x^2 + 7*x + 15}, -1.5)  # should be -1
zero.finder (function(x) {x^3 - 7*x^2 + 7*x + 15}, 4.5)  # should be 5
```

6. Find the nearest maximum of a function using the Newton-Raphson method [http://en.wikipedia.org/wiki/Newton's_method], coded in `maximizer()`. Hint: plot these functions using `curve()` to see where they should maximize. If the minimum

```
zero.finder (function(x) {1 - x^2}, 1.5)  # should be 1
zero.finder (function(x) {1 - x^2}, 0.5)  # should be 1
maximizer (function(x) {x^3 - 7*x^2 + 7*x + 15}, 3.5)  # should be ~ 0.569
maximizer (function(x) {x^3 - 7*x^2 + 7*x + 15}, -1.5)  # should be ~ 0.569
maximizer (function(x) {x^3 - 7*x^2 + 7*x + 15}, 4.5)  # should be Inf -- build a check for this
```

7. The Fibonacci sequence is better defined dynamically, and one attempt at this is in `dynamic.fibonacci()`. It should return the numbers corresponding to `1,2,3,5,8,13 ...` for those values, and zero for any disallowed values.

```
dynamic.fibonacci (10)
dynamic.fibonacci (1)
dynamic.fibonacci (0)
dynamic.fibonacci (0.5)
```