Assignment 10 (R)

Glenn Niles

11-8-2018

Efficiency

a. Install and load the `multcomp` package, to allow testing for statistically significant differences in function timings.

```
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

Load other necessary packages here.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggformula)
```

```
## Loading required package: ggplot2

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```r
library(microbenchmark)
```

In this problem, you will work with the cleaned version of the US News and World Report data on colleges and universities, which you created in Homework 9.

b. Read the data into R.

```r
usnews = read_csv("usnewsR.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   `College Name` = col_character(),
##   State = col_character(),
##   `Student/faculty ratio` = col_double(),
##   Pub_or_Priv = col_character()
## )

## See spec(...) for full column specifications.
```

Use the following code to change spaces in column names into underscores (_). Change the name of the data frame to match what you called the data frame when you read it into R.

```r
names(usnews) = make.names(names(usnews))
```

Write a function using a method of your choice to determine how many schools have a per-student instructional expenditure (`Instructional.expenditure.per.student`) higher than their out-of-state tuition (`Out.of.state.tuition`).
- Your function for this part of the problem should not use control flow. - Functions from `dplyr` may be useful here. - Alternatively, the built-in functions `length` and `which` may be useful.

```
NoControlFlow <- function(usnews) {
  length(which(usnews$Instructional.expenditure.per.student > usnews$Out.of.state.tuition))
}
```

Run the function.

```
NoControlFlow(usnews)
```

```
## [1] 457
```

c. Write a function using control flow to determine how many schools have a per-student instructional expenditure higher than their out-of-state tuition.

```
WithControlFlow <- function(usnews){
  count = 0
  num.rows = dim(usnews)[1]
  for(i in 1:num.rows){
    if( length(which(usnews[i, 34] > usnews[i, 23])))
      count = count + 1
  }
  return(count)
}
```

Run the function and check that you get the same answer as in part b.

```
count = WithControlFlow(usnews)
count
```

```
## [1] 457
```

d. Use `microbenchmark` to compare the running times of the two methods you wrote in parts b and c.

```
timings = microbenchmark(
  NoControlFlow(usnews),
  WithControlFlow(usnews)
)
timings
```

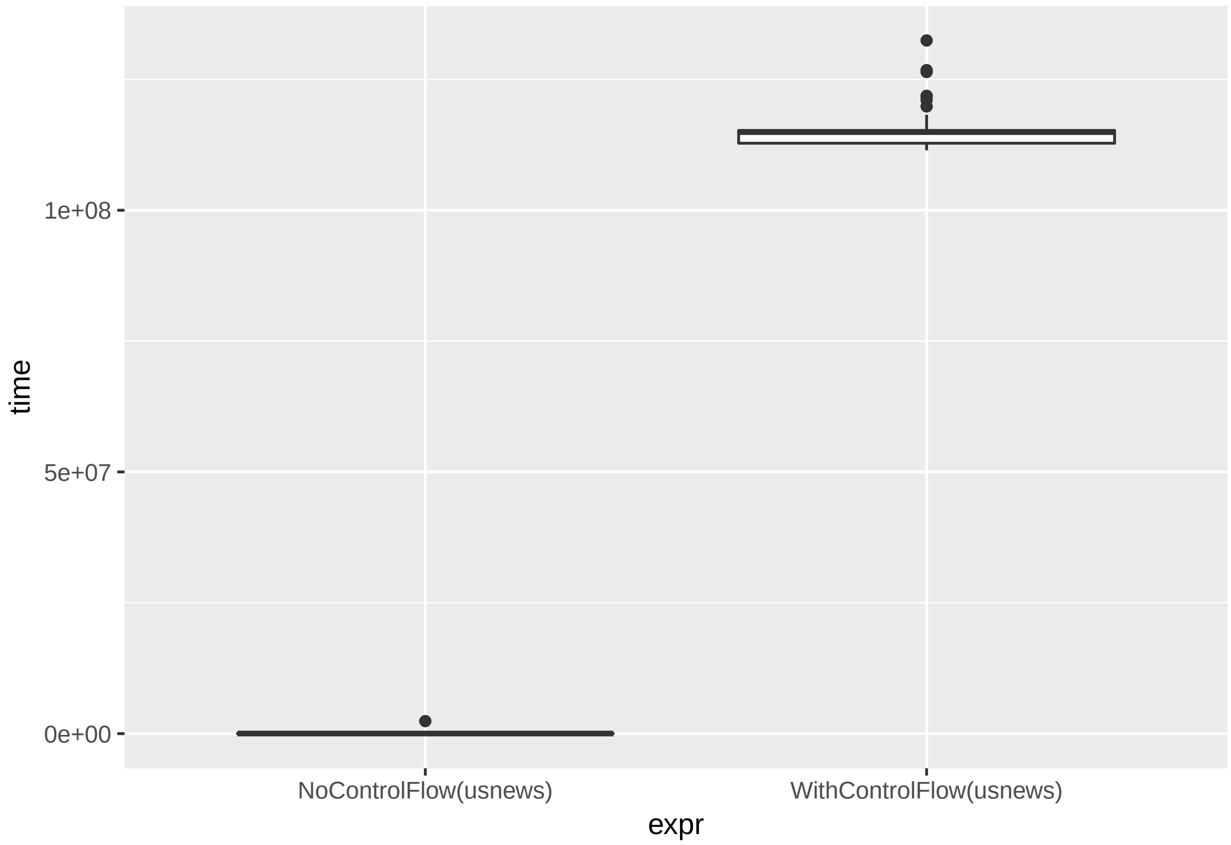
```
## Unit: microseconds
##                     expr        min         lq         mean      median
##    NoControlFlow(usnews)      7.473      9.411     38.07134     10.9325
##  WithControlFlow(usnews) 111447.893 112784.351 115051.29344 114928.6600
##          uq        max neval cld
##      19.927   2387.026   100  a 
##  115250.389 132438.211   100   b
```

- **Write 1-2 sentences** answering the following: Is there a significant difference in the running times of the two methods? If so, which is more efficient?

There is a significant difference in the amount of time it takes to run the two functions. The function without control flow is much, much faster than the function using flow control.

e. Make a boxplot showing the timing comparison of the two methods.

```
gf_boxplot(time ~ expr, data = timings)
```


time
1e+08
5e+07
0e+00
NoControlFlow(usnews)
WithControlFlow(usnews)
expr