Week 4 Assignment

Dan Brooks

February 14, 2016

This may be a little bit of over kill, but I wanted to try and extract the first names and then last names separately to practice my extracting skill. I then concatenated the two strings together to get the first name last name order that the assignment calls for.

```
library(stringr)
name <- c("Moe Szyzlak", "Burns, C. Montgomery", "Rev. Timothy Lovejoy", "Ned Flanders", "Simpson, Home
name
```

```
## [1] "Moe Szyzlak"          "Burns, C. Montgomery" "Rev. Timothy Lovejoy"
## [4] "Ned Flanders"         "Simpson, Homer"       "Dr. Julius Hibbert"
```

```
first_name <- unlist(str_extract_all(toupper(name),
"M..[[:space:]]|R..\\.[[:space:]][[:alpha:]]{7}|NED|D.\\.[[:space:]][[:alpha:]]{6}|C[[:punct:]][[:space
first_name
```

```
## [1] "MOE "          "C. MONTGOMERY" "REV. TIMOTHY"  "NED"
## [5] "HOMER"         "DR. JULIUS"
```

```
last_name <- unlist(str_extract_all(toupper(name),
"SZ[[:alpha:]]{5}|B[N-U]{4}|LO.+|F[[:alpha:]]{7}|SI.{5}|HI.{5}"))
last_name
```

```
## [1] "SZYZLAK"  "BURNS"    "LOVEJOY"  "FLANDERS" "SIMPSON"  "HIBBERT"
```

```
full_name <- str_c(first_name, last_name, sep = " ")
full_name
```

```
## [1] "MOE  SZYZLAK"         "C. MONTGOMERY BURNS"  "REV. TIMOTHY LOVEJOY"
## [4] "NED FLANDERS"         "HOMER SIMPSON"        "DR. JULIUS HIBBERT"
```

```
char_vector <- str_detect(toupper(full_name), "[[:alpha:]]{2,3}[[:punct:]]")
char_vector
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```
##I am assuming that a second name is anybody that has more than two parts to their name.
##Homer Simpson only has one name, but Dr. Julius Hibbert has more than one.

second_name <- str_detect(toupper(full_name), "[[:alpha:]]{1,3}[[:punct:]][[:space:]][[:alpha:]]")
second_name
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE  TRUE
```

```
vector <- c("<title>+++BREAKING NEWS+++</title>")
HTMLtag <- unlist(str_extract_all(vector, "<.+>"))
HTMLtag
```

```
## [1] "<title>+++BREAKING NEWS+++</title>"
```

```
##This fails because the (.) means that it can be any character. The (+) sign
##after that means that the wildcard can be matched one or more times. That
##means that and character can be macthed any amount of times. That will give
##you the entire enxpression. The <> characters really do not matter in this case.
```

```
HTMLtagfix <- unlist(str_extract_all(vector,"<[[:alpha:]]+>"))
HTMLtagfix
```

```
## [1] "<title>"
```

```
binomial_theorem <- c("(5-3)^2=5^2-2*5*3+3^2")
extract <- unlist(str_extract_all(binomial_theorem, "[^0-9=+*()]+"))
extract
```

```
## [1] "-" "^" "^" "-" "^"
```

```
##This fails because there is no \\ before the special characters.
##^ and (-) need to be \\^ and \\- for the expression to work.
```

```
extract_expression <- unlist(str_extract_all(binomial_theorem, "[\\^\\-[0-9]=+*()]"))
extract_expression
```

```
##  [1] "(" "5" "-" "3" ")" "^" "2" "=" "5" "^" "2" "-" "2" "*" "5" "*" "3"
## [18] "+" "3" "^" "2"
```

```
Code <- c('clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdij7Lj8kpf03AT5Id
numbers <- unlist(str_extract_all(Code, "[0-9]"))
numbers
```

```
##  [1] "1" "0" "8" "7" "7" "9" "2" "8" "5" "5" "0" "7" "8" "0" "3" "5" "3"
## [18] "0" "7" "5" "5" "3" "3" "6" "4" "1" "1" "6" "2" "4" "9" "0" "5" "6"
## [35] "5" "1" "7" "2" "4" "6" "3" "9" "5" "8" "9" "6" "5" "9" "4" "9" "0"
## [52] "5" "4" "5"
```

```
lower_case <- unlist(str_extract_all(Code, "[a-z]"))
lower_case
```

```
##   [1] "c" "l" "c" "o" "p" "o" "w" "z" "m" "s" "t" "c" "d" "w" "n" "k" "i"
##  [18] "g" "v" "d" "i" "c" "p" "u" "g" "g" "v" "h" "r" "y" "n" "j" "u" "w"
##  [35] "c" "z" "i" "h" "q" "r" "f" "p" "x" "s" "j" "d" "w" "p" "n" "a" "n"
##  [52] "w" "o" "w" "i" "s" "d" "i" "j" "j" "k" "p" "f" "d" "r" "c" "o" "c"
##  [69] "b" "t" "y" "c" "z" "j" "a" "t" "a" "o" "o" "t" "j" "t" "j" "n" "e"
##  [86] "c" "f" "e" "x" "r" "w" "w" "w" "o" "j" "i" "g" "d" "v" "r" "f" "r"
## [103] "b" "z" "b" "k" "n" "b" "h" "z" "g" "v" "i" "z" "c" "r" "o" "p" "w"
## [120] "g" "n" "b" "q" "o" "f" "a" "o" "t" "f" "b" "w" "m" "k" "t" "s" "z"
## [137] "q" "e" "f" "y" "n" "d" "t" "k" "c" "f" "g" "m" "c" "g" "x" "o" "n"
## [154] "h" "k" "g" "r"
```

```
upper_case <- unlist(str_extract_all(Code, "[A-Z\\!]"))
upper_case
```

```
##  [1] "C" "O" "N" "G" "R" "A" "T" "U" "L" "A" "T" "I" "O" "N" "S" "Y" "O"
## [18] "U" "A" "R" "E" "A" "S" "U" "P" "E" "R" "N" "E" "R" "D" "!"
```

I decided to start off by parsing out the code by upper case, lower case and numbers. I figured that would bea good way to start. It turns out that the answer just happens to be in all captial letters. Makes the answer pretty much stand right out. The answer is: CONGRATULATIONS YOU ARE A SUPER NERD!