Advent of Code Day 3

Dr Jeremiah MF Kelly

2024-12-02

Training

will use regex to select all 'mul(d,d)' where d? is is some number

```
mul <- function(a,b){a*b}
tmp <- read.table('test.txt')
tmp
```

```
##                                                                         V1
## 1 xmul(2,4)%&mul[3,7]!@^do_not_mul(5,5)+mul(32,64]then(mul(11,8)mul(8,5))
```

```
line <- unlist(str_extract_all(tmp, 'mul\\(\\d+\\,\\d+\\)'))
# line
# print(length(line))
out = 0
for(ii in 1:length(line)){
 out <- out + eval(str2expression(line[ii]))
}
out
```

```
## [1] 161
```

```
lines <- readLines('input_03.txt')
# length(lines)
out = 0
pattern = 'mul\\(\\d+\\,\\d+\\)'
for(jj in 1:length(lines)){
  line <- lines[jj]
  line <- unlist(str_extract_all(line,pattern))
  for(ii in 1:length(line)){
     out <- out + eval(str2expression(line[ii]))
    }
}
out
```

```
## [1] 155955228
```

There are two new instructions you'll need to handle:

```
The do() instruction enables future mul instructions.
The don't() instruction disables future mul instructions.
```

We define *do* and *don't* to redefine *mul* and return zero

```
do <- function(){
  mul <<- function(a,b){return(a*b)}
  0
```

```r
  }


dont <- function(){
  mul <<- function(a,b){return(0)}
  0
  }

mul <- function(a,b){a*b}
# set counter
out = 0
pattern = "don\\'t\\(\\)|do\\(\\)|mul\\(\\d+\\,\\d+\\)"

tmp <- read.table('test_1.txt')
line <- unlist(
  str_extract_all(tmp,pattern)
)

for(ii in 1:length(line)){
  if(line[ii] == "don't()"){
    line[ii] = "dont()"
  }
  out <- out + eval(str2expression(line[ii]))
}
out
```

```
## [1] 48
```

No part 2

```r
lines <- readLines('input_03.txt')
out = 0
for(jj in 1:length(lines)){
  line <- lines[jj]
  line <- unlist(str_extract_all(line,pattern))
  for(ii in 1:length(line)){
    if(line[ii] == "don't()"){
      line[ii] = "dont()"
    }
    out <- out + eval(str2expression(line[ii]))
  }
}
out
```

```
## [1] 100189366
```