Exercise: file manipulation

Peter Meißner

28 Februar 2016

```
library(magrittr)
library(stringr)
```

```
if( grepl("file_manipulation_exercise",basename(getwd())) ){
  list.files(pattern = "txt$")  %>%
    file.remove()

  list.files(pattern = "^\\d.*\\d$") %>%
    file.remove()

  file.remove("mylogfile")
}
```

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

1) Logging the System time

a) write a loop that

- sleeps 1 second
- writes a text-file with name equal to the current time

```
for(i in 1:10){
  Sys.sleep(1)

  Sys.time() %>%
    as.character() %>%
    str_replace_all(":", "_") %>%
    writeLines( "blah", .)
}

list.files(pattern = "^\\d")
```

```
##  [1] "2016-03-03 02_35_43" "2016-03-03 02_35_44" "2016-03-03 02_35_45"
##  [4] "2016-03-03 02_35_46" "2016-03-03 02_35_47" "2016-03-03 02_35_48"
##  [7] "2016-03-03 02_35_49" "2016-03-03 02_35_50" "2016-03-03 02_35_51"
## [10] "2016-03-03 02_35_52"
```

b) write a function that

- writes a file with some content of your choosing
- but only if a file with the same name does not exist or is older than 20 seconds

example of a function writing a file

```
  write_punny_file <- function(name="not_a_clever_name.txt"){
    timediff <-
      as.numeric(Sys.time()) -
    as.numeric(file.info(name)$mtime)

    if( !file.exists(name) | timediff > 20 | is.na(timediff > 20) ){
      writeLines("content", name)
      return(TRUE)
    }else{
      return(FALSE)
    }
  }

write_punny_file()
```

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

```
list.files(pattern = "^not_")
```

```
## [1] "not_a_clever_name.txt"
```

```
Sys.time()
```

```
## [1] "2016-03-03 02:35:52 CET"
```

```
file.info("not_a_clever_name.txt")$mtime
```

```
## [1] "2016-03-03 02:35:52 CET"
```

c) write a loop that

- sleeps for a second
- appends the current time to a file

```
for( i in 1:10){
  write(
    as.character(Sys.time()),
    "mylogfile", append = TRUE)
  Sys.sleep(0.3)
}

readLines("mylogfile")
```

```
##  [1] "2016-03-03 02:35:52" "2016-03-03 02:35:53" "2016-03-03 02:35:53"
##  [4] "2016-03-03 02:35:53" "2016-03-03 02:35:53" "2016-03-03 02:35:54"
##  [7] "2016-03-03 02:35:54" "2016-03-03 02:35:54" "2016-03-03 02:35:55"
## [10] "2016-03-03 02:35:55"
```

2) Pasting file names

a) generate a data.frame

- with two variables
- each having three values
- build a double loop (one over variable 1, one over variable 2)
- for each iteration a file should be written to disk
- the file name should entail the value of variable one and two

example of a double loop

```
  for( i in 1:3){
    for( j in 1:3){
      cat("- i=", i, "j=", j, "\n")
    }
  }
```

solution

```
var1 <- letters[1:3]
var2 <- LETTERS[7:9]
df  <- data.frame(var1, var2)

  for( i in seq_along(var1)){
    for( j in seq_along(var2)){
      txt   <- paste("var1 =", df$var1[i], "\nvar2 =", df$var2[j])
      fname <- paste0(df$var1[i], "_", df$var2[j], ".txt")
      writeLines(txt, fname)
    }
  }

list.files(pattern = "txt$")
```

```
##  [1] "a_G.txt"               "a_H.txt"
##  [3] "a_I.txt"               "b_G.txt"
##  [5] "b_H.txt"               "b_I.txt"
##  [7] "c_G.txt"               "c_H.txt"
##  [9] "c_I.txt"               "not_a_clever_name.txt"
```

```
readLines("a_G.txt")
```

```
## [1] "var1 = a " "var2 = G"
```

```
readLines("c_I.txt")
```

```
## [1] "var1 = c " "var2 = I"
```

b) use expand.grid() on the data.frame created before

- than use apply() to apply the paste function with option collapse="_" on each row
- use the resulting strings to create the same file as in ex-2.a

```r
df_expanded <- expand.grid(df)
df_expanded
```

```
##   var1 var2
## 1    a    G
## 2    b    G
## 3    c    G
## 4    a    H
## 5    b    H
## 6    c    H
## 7    a    I
## 8    b    I
## 9    c    I
```

```r
df_expanded$fnames <-
  df_expanded %>%
  apply(1, paste0, collapse="_") %>%
  paste0(".txt")

for( i in seq_along(df_expanded[,1]) ){
  with(
    df_expanded,
    writeLines(
      paste(".var1 =", var1[i], "\n.var2 =", var2[i]),
      fnames[i],
    )
  )
}

list.files(pattern = "txt$")
```

```
##  [1] "a_G.txt"               "a_H.txt"
##  [3] "a_I.txt"               "b_G.txt"
##  [5] "b_H.txt"               "b_I.txt"
##  [7] "c_G.txt"               "c_H.txt"
##  [9] "c_I.txt"               "not_a_clever_name.txt"
```

```r
readLines("a_G.txt")
```

```
## [1] ".var1 = a " ".var2 = G"
```

```r
readLines("c_I.txt")
```

```
## [1] ".var1 = c " ".var2 = I"
```

3) Reading files

a) use list.files() to get a vector of file names in the current folder

```
list.files()
```

```
##  [1] "2016-03-03 02_35_43"             "2016-03-03 02_35_44"
##  [3] "2016-03-03 02_35_45"             "2016-03-03 02_35_46"
##  [5] "2016-03-03 02_35_47"             "2016-03-03 02_35_48"
##  [7] "2016-03-03 02_35_49"             "2016-03-03 02_35_50"
##  [9] "2016-03-03 02_35_51"             "2016-03-03 02_35_52"
## [11] "a_G.txt"                         "a_H.txt"
## [13] "a_I.txt"                         "b_G.txt"
## [15] "b_H.txt"                         "b_I.txt"
## [17] "c_G.txt"                         "c_H.txt"
## [19] "c_I.txt"                         "file_manipulation_exercise.html"
## [21] "file_manipulation_exercise.pdf"  "file_manipulation_exercise.R"
## [23] "file_manipulation_exercise.Rmd"  "mylogfile"
## [25] "not_a_clever_name.txt"
```

b) use list.files() in combination with option pattern

- to only get a subset of files
- use three different subsets

```
list.files(pattern = "^\\d")
```

```
##  [1] "2016-03-03 02_35_43" "2016-03-03 02_35_44" "2016-03-03 02_35_45"
##  [4] "2016-03-03 02_35_46" "2016-03-03 02_35_47" "2016-03-03 02_35_48"
##  [7] "2016-03-03 02_35_49" "2016-03-03 02_35_50" "2016-03-03 02_35_51"
## [10] "2016-03-03 02_35_52"
```

```
list.files(pattern = "txt$")
```

```
##  [1] "a_G.txt"               "a_H.txt"
##  [3] "a_I.txt"               "b_G.txt"
##  [5] "b_H.txt"               "b_I.txt"
##  [7] "c_G.txt"               "c_H.txt"
##  [9] "c_I.txt"               "not_a_clever_name.txt"
```

```
list.files(pattern = "manipulation")
```

```
## [1] "file_manipulation_exercise.html" "file_manipulation_exercise.pdf"
## [3] "file_manipulation_exercise.R"    "file_manipulation_exercise.Rmd"
```

```
list.files(pattern = "manipulation", full.names = TRUE, recursive = TRUE)
```

```
## [1] "./file_manipulation_exercise.html" "./file_manipulation_exercise.pdf"
## [3] "./file_manipulation_exercise.R"    "./file_manipulation_exercise.Rmd"
```

c) make an empy list

- use list.files() to get a vector of file names
- use file.info() to get further information on the files (mtime)
- read in the content of the files
  - only for those files that are newest according to mtime

```
fnames <- list.files(full.names = TRUE, recursive = TRUE)

info   <- file.info(fnames)
mtime  <- info$mtime

newest <- fnames[order(mtime, decreasing = TRUE)]

texts  <- lapply(newest, readLines, warn=FALSE)
```