PS02

Yingyue Luan

September 10 2017

Problem 1

(a)

```
## save letters in text format

chars <- sample(letters, 1e6, replace = TRUE)

write.table(chars, file = 'tmp1.csv', row.names = FALSE, quote = FALSE,
    col.names = FALSE)

system('ls -l tmp1.csv', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  2000000 Sep 15 12:44 tmp1.csv"

## [1] "-rw-r--r-- 1 paciorek scfstaff 2000000 Sep  8  2017 tmp1.csv"

chars <- paste(chars, collapse = '')

write.table(chars, file = 'tmp2.csv', row.names = FALSE, quote = FALSE,
    col.names = FALSE)

system('ls -l tmp2.csv', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  1000001 Sep 15 12:44 tmp2.csv"

## [1] "-rw-r--r-- 1 paciorek scfstaff 1000001 Sep  8  2017 tmp2.csv"
```

1. Create a sample of $1e^6$ of the 26 lower case letters with replacement
2. Output the sample as data frame with name 'tmp1.csv', no row or column name, no quotes
3. Invoke a system command to list the file in long format and capture the output as r vector

4. Concatenate all letters without space
5. Output the result which shrinks from 2mb to 1mb.

In ascii, 7 bits approximately equal to 1 bytes. Adding the spaces between letters, the text file originally have $2e^6$ characters. When we delete the spaces, the file now has $1e^6$ characters.

```
## save in binary format

nums <- rnorm(1e6)

save(nums, file = 'tmp3.Rda')

system('ls -l tmp3.Rda', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  7678177 Sep 15 12:44 tmp3.Rda"

## [1] "-rw-r--r-- 1 paciorek scfstaff 7678109 Sep  8  2017 tmp3.Rda"
```

1. Random generate of a standard normal distribution with $n = 1e^6$
2. Save the numbers into a Rda file
3. Output the result. The file size is 7mb which shows that numbers are stored as 8 bytes per number in binary format

```
## save in text format

write.table(nums, file = 'tmp4.csv', row.names = FALSE, quote = FALSE,
    col.names = FALSE, sep = ',')

system('ls -l tmp4.csv', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  18160741 Sep 15 12:44 tmp4.csv"

## [1] "-rw-r--r-- 1 paciorek scfstaff 18160350 Sep  8  2017 tmp4.csv"

write.table(round(nums, 2), file = 'tmp5.csv', row.names = FALSE, quote
    = FALSE, col.names = FALSE, sep = ',')

system('ls -l tmp5.csv', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  5377594 Sep 15 12:44 tmp5.csv"

## [1] "-rw-r--r-- 1 paciorek scfstaff 5378678 Sep 8 2017 tmp5.csv"
```

1. Save numbers into a text file
2. Output the csv file in long format which is 18mb. The file size of a text file is approximately three times larger than a binary file. Take the number 1 for example, we convert it from ascii to binary 00110001 which is 31 in hex. If we save in text file, we are actually store the ascii code of "3", "1" and possible spaces between number 1 and other numbers. However in binary file, we are only saving one byte per hex pairs. Therefore, the ascii file might contain three times more bytes than binary file.
3. Save the rounded numbers into a text file
4. Output the new file which is 5mb

(b)

```
chars <- sample(letters, 1e6, replace = TRUE)

chars <- paste(chars, collapse = '')

save(chars, file = 'tmp6.Rda')

save(chars, file = 'tmp8.Rda', ascii = TRUE)

system('ls -l tmp6.Rda', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  635285 Sep 15 12:44 tmp6.Rda"

system('ls -l tmp8.Rda', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  1000070 Sep 15 12:44 tmp8.Rda"
```

1. The codes above generate a smaple of $1e^6$ letteres and save the characters into an Rda binary file. The size of the file is less than 1mb
2. The defaut option for ascii in save function is FALSE which means a binary file is saved. If we put ascii = TRUE, the file size wil change to 1000070 bytes

```
chars <- rep('a', 1e6)

chars <- paste(chars, collapse = '')

save(chars, file = 'tmp7.Rda')
```

```
save(chars, file = 'tmp9.Rda', ascii = TRUE)

system('ls -l tmp7.Rda', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  1056 Sep 15 12:44 tmp7.Rda"

system('ls -l tmp9.Rda', intern = TRUE)

## [1] "-rw-r--r--  1 lunaluan  staff  1000070 Sep 15 12:44 tmp9.Rda"

system('cat tmp7.Rda', intern = TRUE)

## [1] "\037\x8b\b" "h\xc2,\x99"
```

1. Here, if we do the same of change the value of ascii to TRUE in save function, the file size is also 1000070 bytes, meaning the file is saved as a text file of $1e^6$ characters.
2. The binary file of repetitive "a" is significantly smaller than the previous file with the same amount of characters. Instead of returning thousands of binary code for storing all the letters in the previous file, the file of repetitive "a" can be encoded by simply \037\x8b \b" "h\xc2,\x99"

Problem 2

a

```
library(XML)

returnpage = function(name) {
  first = strsplit(name, " ")[[1]][1]
  last = strsplit(name, " ")[[1]][2]

  if (is.na(first) || is.na(last) || first == '' || last == '') {
    return("Please enter valid input")
  } else {
    url = paste0("https://scholar.google.com/citations?view_op=
       search_authors&mauthors=", first, "+", last,"&hl=en&oi=ao")
    html = readLines(url)
    links = getHTMLLinks(html)
    targetlinks = links[grep('user=', links)[1]]
    if (is.na(targetlinks) == TRUE) {
```

```
      return("Your search didn't match any user profiles.")
    } else {
      id = gsub("(.*user=)","",targetlinks)
      id = gsub("(&.*)","",id)
      pageurl = paste0("https://scholar.google.com/citations?user=",id
          ,"&hl=en&oi=ao")
      citationpage = readLines(pageurl)
      return(c(id,citationpage))
    }
  }
}
```

```
> returnpage("Geoffrey Hinton")
[1] "JicYPdAAAAAJ"

    [2] "<!doctype html><head><meta http-equiv=\"Content-Type\" content
    =\"text/html;charset=ISO-8859-1\"><meta http-equiv=\"X-UA-Compatible
    \" content=\"IE=Edge\"><meta name=\"referrer\" content=\"always\"><
    meta name=\"viewport\" content=\"width=device-width,initial-scale=1,
    minimum-scale=1,maximum-scale=2\"><style>@viewport{width:device-
    width;min-zoom:1;max-zoom:2;}</style><meta name=\"format-detection\"
     content=\"telephone=no\"><style>html,body,form,table,div,h1,h2,h3,
    h4,h5,h6,img,ol,ul,li,button{margin:0;padding:0;border:0;}table{
    border-collapse:collapse;border-width:0;empty-cells:show;}#gs_top{
    position:relative;min-width:964px;-webkit-tap-highlight-color:rgba
    (0,0,0,0);}#gs_top>*:not(#x){-webkit-tap-highlight-color:rgba
    (204,204,204,.5);}.gs_el_ph #gs_top,.gs_el_ta #gs_top{min-width:300
    px;}#gs_top.gs_nscl{position:fixed;width:100%;}body,td,input{font-
    size:13px;font-family:Arial,sans-serif;line-height:1.24}body{
    background:#fff;color:#222;-webkit-text-size-adjust:100%;-moz-text-
    size-adjust:n... <truncated>
```

b

```
createdataframe = function(name){

  citationpage = returnpage(name)[2]
  page = htmlParse(citationpage)
  docs = getNodeSet(page, "//a[@class='gsc_a_at']")
  title = sapply(docs, xmlValue)

  divs = getNodeSet(page, "//div[@class='gs_gray']")
  len = length(sapply(divs, xmlValue))
```

```
  author = sapply(divs, xmlValue)[seq(1,len,2)]

  journal = sapply(divs, xmlValue)[seq(2,len,2)]
  a = strsplit(journal, ",(?=[^,]+$)", perl=TRUE)
  for (i in 1:length(a)) {
     a[[i]] = a[[i]][-length(a[[i]])]
     i=i+1
  }
  a=c(list(a[1:length(a)]), recursive = TRUE)

  table = getNodeSet(page, "//table")[[2]]
  x = readHTMLTable(table)

  citedby = x[2]
  year = x[3]

  data.frame("title" = c(title), "author" = c(author), "journal" = c(a)
      , "citedby" = c(citedby), "year" = c(year))

}


createdataframe("Geoffrey Hinton")
```

```
## Warning in readLines(url): incomplete final line found on 'https://scholar.google.com/citations?view_op=
## Warning in readLines(pageurl): incomplete final line found on 'https://scholar.google.com/citations?user
```

```
##
```

```
    title
## 1                                                              Learning
    representations by back-propagating errors
## 2                                                        Learning
    internal representations by error-propagation
## 3                                                        Learning
    internal representations by error propagation
## 4

    Parallel distributed processing
## 5                                              Imagenet
    classification with deep convolutional neural networks
## 6                                                                    A
    fast learning algorithm for deep belief nets
## 7

    Parallel distributed processing
```

```
## 8                                                          Reducing the
    dimensionality of data with neural networks
## 9

    Deep learning
## 10

    Adaptive mixtures of local experts
## 11                                          Dropout: a simple way to
    prevent neural networks from overfitting.
## 12                                                                   A
    learning algorithm for Boltzmann machines
## 13

    Visualizing data using t-SNE
## 14 Deep neural networks for acoustic modeling in speech recognition:
     The shared views of four research groups
## 15                                           Training products of
    experts by minimizing contrastive divergence
## 16                          A view of the EM algorithm that
    justifies incremental, sparse, and other variants
## 17                                                        Phoneme
    recognition using time-delay neural networks
## 18                                 Improving neural networks by
    preventing co-adaptation of feature detectors
## 19                                              Rectified linear
    units improve restricted boltzmann machines
## 20

    Connectionist learning procedures
##
    author
## 1                                  DE Rumelhart, GE Hinton, RJ
    Williams
## 2                                  DE Rumelhart, GE Hinton, RJ
    Williams
## 3                                  DE Rumelhart, GE Hinton, RJ
    Williams
## 4                       DE Rumelhart, JL McClelland, PDP Research
    Group
## 5                                  A Krizhevsky, I Sutskever, GE
    Hinton
## 6                                          GE Hinton, S Osindero, YW
    Teh
## 7                       JL McClelland, DE Rumelhart, PDP Research
```

```
   Group
## 8                                                GE Hinton, RR
   Salakhutdinov
## 9                                                Y LeCun, Y Bengio, G
   Hinton
## 10                              RA Jacobs, MI Jordan, SJ Nowlan, GE
   Hinton
## 11  N Srivastava, GE Hinton, A Krizhevsky, I Sutskever, R
   Salakhutdinov
## 12                                       DH Ackley, GE Hinton, TJ
   Sejnowski
## 13                                                L van der Maaten, G
   Hinton
## 14  G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior,
   ...
## 15                                                                GE
   Hinton
## 16                                                       RM Neal, GE
   Hinton
## 17                     A Waibel, T Hanazawa, G Hinton, K Shikano, KJ
   Lang
## 18 GE Hinton, N Srivastava, A Krizhevsky, I Sutskever, RR
   Salakhutdinov
## 19                                                        V Nair, GE
   Hinton
## 20                                                                GE
   Hinton
##

   journal
## 1                                                                 Nature
    323, 533-536
## 2      Parallel Distributed Processing: Explorations in the
   Microstructure of ...
## 3                                       CALIFORNIA UNIV SAN DIEGO LA
   JOLLA INST FOR
## 4                                                                   MIT
    press 1, 184
## 5                     Advances in neural information processing
   systems, 1097-1105
## 6                                                Neural computation 18
   (7), 1527-1554
## 7

   MIT press
```

```
## 8                                                                 science 313
    (5786), 504-507
## 9                                                                  Nature 521
    (7553), 436-444
## 10                                                          Neural computation
     3 (1), 79-87
## 11                          Journal of machine learning research 15
    (1), 1929-1958
## 12                                                       Cognitive science 9
     (1), 147-169
## 13                         Journal of Machine Learning Research 9 (
    Nov), 2579-2605
## 14                                    IEEE Signal Processing Magazine
    29 (6), 82-97
## 15                                             Neural computation 14
    (8), 1771-1800
## 16                                            Learning in graphical
    models, 355-368
## 17  IEEE transactions on acoustics, speech, and signal processing 37
     (3), 328-339
## 18                                                   arXiv preprint
    arXiv:1207.0580
## 19 Proceedings of the 27th international conference on machine
    learning (ICML ...
## 20                                       Artificial intelligence 40
    (1-3), 185-234
##    Cited.by Year
## 1    34900* 1986
## 2    27417* 1986
## 3     23094 1985
## 4     18726 1987
## 5     15040 2012
## 6      6618 2006
## 7     6477* 1987
## 8      5614 2006
## 9      3793 2015
## 10     3551 1991
## 11     3515 2014
## 12     3316 1985
## 13     3149 2008
## 14     3147 2012
## 15     2949 2002
## 16     2468 1998
## 17     2396 1989
## 18     2111 2012
```

```
## 19     2040 2010
## 20     1814 1989
```

c

```
library(testthat)

context("Finding invalid input")

test_that("returnpage handles input correctly", {
  expect_equal(returnpage("aba"), "Please enter valid input")
  expect_equal(returnpage(""), "Please enter valid input")
})

test_that("returnpage did not find a match", {
  expect_equal(returnpage("Yingyue Luan"), "Your search didn't match
     any user profiles.")
})
```

I corrected my code directly in part (a).