NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. IVth Sem

Subject Code: CS216

Subject Name: Applied Probability

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

Sch Id: 1912160
Name: Subhojit Ghimire
Branch: CSE-B
Subject: Applied Probability Lab
Subject Code: CS216

Q.2. Describe the steps of reading the following files (a)-(d) and writing the following files (e)-(f), using R.

(a) .gz files
(b) .bz2 files
(c) URLs
(d) MS Excel files
(e) Binary files
(f) ASCII files.

➡ (a) Reading a .gz file:

R code snippet : > read.table ("filename.txt.gz")

or, Saving data along with reading,

```
> dataframeName <- read.table ("filename.txt.gz")
> print (dataframeName)
```

The file name can be simply .gz or in various other format like txt.gz or csv.gz, where any extension before .gz are considered to be filename itself.

The read.table can be replaced with other commands like readline, read.csv, read.delim etc. as per needs.

(b) Reading a .bz2 file.

R code snippet : > read.table ("filename.txt.bz2")

The similar conditions hold true for reading .bz2 files as it holds for .gz files.

(c) Reading URLs:

R code snippet : > read.csv(url("https://link.com/name.csv"))

or simply : > read.csv("https://link.com/name.csv")

The similar conditions or alternatives apply to URL files as for .gz or .bz2 files, like read.table, read.csv2 etc.

(d) Reading Excel files:

1. Install "xlsx" package using: > install.packages("xlsx")
2. Load "xlsx" using: > library("xlsx")
3. This enables use of read.xlsx and read.xlsx2.

Code snippet: > read.xlsx("filename.xls", sheetIndex = 1).

Alternatively,

1. Install "readxl" package : > install.packages("readxl")
2. Load "readxl" package : > library("readxl")

Code snippet: > read_excel("filename.xls", sheet = 1)

(e) Writing Binary files:

Code Snippet : > write.filename = file("filename1.dat", "wb")
> writeBin(createdDataFrame, write.filename)
> close(write.filename)

(f) Writing ASCII files:

The simple exporting of files in write.table() manner as mentiond in the Q.1. ~~as well as various examples of~~ is an example of ASCII writing of a file.

Code snippet: > write.table(createdDataframe, file="filename.txt", sep="\t", row.names = F, col.names=F)

2. a. Example of Reading a random .gz file downloaded from the internet, using R:

```
> read.table(gzfile ("gap.txt.gz"))
   V1       V2   V3   V4    V5 V6 V7    V8  V9
1  585 JH588217 8550 8551 11789  N  1 other yes
2  585 JH588405 1058 1059 11797  N  1 other yes
3  585 JH588421 2011 2012 11804  N  1 other yes
4  585 JH588454 3149 3150 11809  N  1 other yes
5  585 JH588564 1787 1788 11791  N  1 other yes
6  585 JH588583   18   19 11805  N  1 other yes
7  585 JH588622   25   26 11798  N  1 other yes
8  585 JH588637 1562 1563 11810  N  1 other yes
9  585 JH588637 1590 1591 11811  N  1 other yes
10 585 JH588644 4565 4566 11812  N  1 other yes
11 585 JH588822 4790 4791 11807  N  1 other yes
12 585 JH588858 3830 3831 11818  N  1 other yes
13 585 JH588861 1387 1388 11815  N  1 other yes
14 585 JH588908   11   12 11803  N  1 other yes
15 585 JH588962 2959 2960 11817  N  1 other yes
16 585 JH589082 1892 1893 11794  N  1 other yes
17 585 JH589173   28   29 11814  N  1 other yes
18 585 JH589270 2151 2152 11821  N  1 other yes
19 585 JH589280 4368 4369 11813  N  1 other yes
20 585 JH589327   61   62 11808  N  1 other yes
21 585 JH589347   33   34 11820  N  1 other yes
22 585 JH589499 1672 1673 11823  N  1 other yes
23 585 JH589527 4042 4043 11816  N  1 other yes
24 585 JH589642   15   16 11826  N  1 other yes
25 585 JH589750 3884 3885 11819  N  1 other yes
26 585 JH589880 2765 2766 11802  N  1 other yes
27 585 JH590107  927  928 11835  N  1 other yes
28 585 JH590158 1111 1112 11830  N  1 other yes
29 585 JH590164 3008 3009 11829  N  1 other yes
30 585 JH590262  305  306 11825  N  1 other yes
31 585 JH590334 3409 3410 11822  N  1 other yes
```

2. b. Example of reading the same .gz file converted to .bz2 file as a process of reading .bz2 file, using R:

```
> read.table("gap.txt.bz2")
   V1       V2   V3   V4    V5 V6 V7    V8  V9
1  585 JH588217 8550 8551 11789  N  1 other yes
2  585 JH588405 1058 1059 11797  N  1 other yes
3  585 JH588421 2011 2012 11804  N  1 other yes
4  585 JH588454 3149 3150 11809  N  1 other yes
5  585 JH588564 1787 1788 11791  N  1 other yes
6  585 JH588583   18   19 11805  N  1 other yes
7  585 JH588622   25   26 11798  N  1 other yes
8  585 JH588637 1562 1563 11810  N  1 other yes
9  585 JH588637 1590 1591 11811  N  1 other yes
10 585 JH588644 4565 4566 11812  N  1 other yes
11 585 JH588822 4790 4791 11807  N  1 other yes
12 585 JH588858 3830 3831 11818  N  1 other yes
13 585 JH588861 1387 1388 11815  N  1 other yes
14 585 JH588908   11   12 11803  N  1 other yes
15 585 JH588962 2959 2960 11817  N  1 other yes
16 585 JH589082 1892 1893 11794  N  1 other yes
17 585 JH589173   28   29 11814  N  1 other yes
18 585 JH589270 2151 2152 11821  N  1 other yes
19 585 JH589280 4368 4369 11813  N  1 other yes
20 585 JH589327   61   62 11808  N  1 other yes
21 585 JH589347   33   34 11820  N  1 other yes
22 585 JH589499 1672 1673 11823  N  1 other yes
23 585 JH589527 4042 4043 11816  N  1 other yes
24 585 JH589642   15   16 11826  N  1 other yes
25 585 JH589750 3884 3885 11819  N  1 other yes
26 585 JH589880 2765 2766 11802  N  1 other yes
27 585 JH590107  927  928 11835  N  1 other yes
28 585 JH590158 1111 1112 11830  N  1 other yes
29 585 JH590164 3008 3009 11829  N  1 other yes
30 585 JH590262  305  306 11825  N  1 other yes
31 585 JH590334 3409 3410 11822  N  1 other yes
```

2. c. Example of reading random URL off the internet containing some csv2 data, using R:

```
> read.csv2 (url("https://support.staffbase.com/hc/en-us/article_attachments/360009197031/username.csv"))
   Username Identifier First.name Last.name
1  booker12       9012     Rachel    Booker
2    grey07       2070      Laura      Grey
3 johnson81       4081      Craig   Johnson
4 jenkins46       9346       Mary   Jenkins
5   smith79       5079      Jamie     Smith
```

2. d. Example of reading a random MS Excel file downloaded off the internet, using R:

USING read.xlsl:

```
> read.xlsx("tenrow.xls", sheetIndex = 1)
  X0 First.Name Last.Name Gender       Country Age
1  1      Dulce     Abril Female United States  32
2  2       Mara Hashimoto Female Great Britain  25
3  3     Philip      Gent   Male        France  36
4  4   Kathleen    Hanner Female United States  25
5  5    Nereida   Magwood Female United States  58
6  6     Gaston     Brumm   Male United States  24
7  7       Etta      Hurn Female Great Britain  56
8  8    Earlean    Melgar Female United States  27
9  9   Vincenza   Weiland Female United States  40
         Date   Id
1 15/10/2017 1562
2 16/08/2016 1582
3 21/05/2015 2587
4 15/10/2017 3549
5 16/08/2016 2468
6 21/05/2015 2554
7 15/10/2017 3598
8 16/08/2016 2456
9 21/05/2015 6548
```

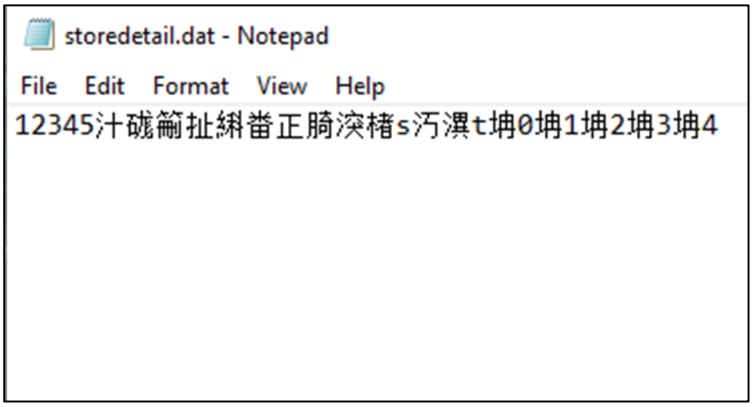
USING read_excel:

```
> read_excel ("tenrow.xls", sheet = 1)
# A tibble: 9 x 8
     `0` `First Name` `Last Name` Gender Country      Age
   <dbl> <chr>        <chr>       <chr>  <chr>      <dbl>
1     1 Dulce        Abril       Female United St~    32
2     2 Mara         Hashimoto   Female Great Bri~    25
3     3 Philip       Gent        Male   France        36
4     4 Kathleen     Hanner      Female United St~    25
5     5 Nereida      Magwood     Female United St~    58
6     6 Gaston       Brumm       Male   United St~    24
7     7 Etta         Hurn        Female Great Bri~    56
8     8 Earlean      Melgar      Female United St~    27
9     9 Vincenza     Weiland     Female United St~    40
# ... with 2 more variables: Date <chr>, Id <dbl>
```

2. e. Example of writing a Binary file, using R:

```
> S.No. <- c(1, 2, 3, 4, 5)
> Name <- c ("Alex", "Bob", "Chuck", "Dennis", "Eliot")
> Sch.Id. <- (160, 161, 162, 163, 164)
Error: unexpected ',' in "Sch.Id. <- (160,"
> Sch.Id. <- c(160, 161, 162, 163, 164)
> detail <- c(S.No., Name, Sch.Id.)
> write.filedetail = file ("storedetail.dat", "wb")
> writeBin (detail, write.filedetail)
> close (write.filedetail)
```

Notepad view of the binary file:



Also, reading the same binary file in R:

```
> read.details <- file ("storedetail.dat", "rb")
> det <- readBin (read.details, character(), n=15)
> close (read.details)
> print (det)
 [1] "1"      "2"      "3"      "4"      "5"
 [6] "Alex"   "Bob"    "Chuck"  "Dennis" "Eliot"
[11] "160"    "161"    "162"    "163"    "164"
```

2. f. Example of writing an ASCII file, using R:

```
> S.No. <- c (1, 2, 3, 4, 5)
> Name <- c ("Alex Rider", "Bob Marshal", "Chuck Norris", "D
ennis Cunningham", "Eliot Prince")
> Sch.Id. <- c (1912160, 1912161, 1912162, 1912163, 1912164)
> details <- data.frame (S.No., Name, Sch.Id.)
> print (details)
  S.No.              Name Sch.Id.
1     1       Alex Rider 1912160
2     2      Bob Marshal 1912161
3     3     Chuck Norris 1912162
4     4 Dennis Cunningham 1912163
5     5      Eliot Prince 1912164
> write.table(details, file = "details.txt", sep = "\t", ro
w.names = FALSE, col.names = TRUE)
```