Reading GUK files

March 9, 2017
10:21

Seiro Ito

Contents

Based on files received in March, 2017.

I read

List folder names.

```
setwd(pathreceived.mar)
foldername ← list.dirs(path = ".", recursive = T, full.names = T)
foldername ← foldername[!grepl("p.$|^.$", foldername)]
```

```
library(foreign)
library(readstata13)
setwd(pathreceived.mar)
fn ← list.files(path = foldername, pattern = ".dta$",
        recursive = T, full.names = T)
```

There will be warnings due to duplicated factor levels in dta which should be a fair warning but suppress them. In ./p1/1/Section8b_LivestockProduction.dta, iga seems to be assigned with noninteger values even if it is coded as a factor variable. I will assign the factor labels from stata value labels.

```
setwd(pathreceived.mar)
X ← lapply(fn, read.dta13, generate.factors = T, nonint.factors = T)
X ← lapply(X, data.table)
```

Save to source folder.

```
setwd(pathsource.mar)
saveRDS(X, "all_data.RDS")
```

Check if files are stored correctly. Section 3B is food security and comes with many yes/no answers.

```
fnshort ← gsub("^\\.\\/(.*?\\/).*(\\/.*?$)", "/\\1...\\2", fn)
fnshort ← gsub("^\\.", "", fnshort)
grepout(" 3B", fnshort)
```

```
[1] "/p2/.../Section 3B.dta" "/p3/.../Section 3B.dta"
```

```
X[grep(" 3B", fnshort)]
```

```
[[1]]
                id s8bq1 s8bq2_1 s8bq2_2 s8bq2_3 s8bq2_4 s8bq2_5 s8bq2_6 s8bq2_7
   1:      7010102    no     yes     yes      no      no      no     yes     yes
```

```
   2:     7010105    no      yes      yes       no       no       no       no      yes
   3:     7010106    no      yes      yes       no       no       no       no      yes
   4:     7010107    no       no      yes       no       no       no       no      yes
   5:     7010108    no      yes      yes       no       no       no       no      yes
  ---
2079: 99081912415    no      yes      yes      yes      yes       no      yes       no
2080: 99081912417   yes       NA       NA       NA       NA       NA       NA       NA
2081: 99081912418   yes       NA       NA       NA       NA       NA       NA       NA
2082: 99081912419    no      yes      yes      yes       no       no       no       no
2083: 99081912420   yes       NA       NA       NA       NA       NA       NA       NA
      s8bq2_8 s8bq2_9 s8bq2_10 s8bq2_11 s8bq2_12 s8bq3_1 s8bq3_2 s8bq3_3
   1:      no      no       no       no      yes        3       10        2
   2:      no      no       no       no      yes        3        7        2
   3:      no      no       no       no      yes        3       12        2
   4:      no      no       no      yes      yes        3        8        2
   5:      no      no       no      yes      yes        3       10        3
  ---
2079:      no     yes       no       no       no        3        2       NA
2080:      NA      NA       NA       NA       NA        3        2        2
2081:      NA      NA       NA       NA       NA        3        2        2
2082:      no      no       no       no       no        3        2        2
2083:      NA      NA       NA       NA       NA        3        2        2
      s8bq3_4 s8bq3_5 s8bq3_6 s8bq3_7 s8bq3_8 s8bq3_9 s8bq3_10 s8bq3_11
   1:       3      30        2        2        6        1         4        24
   2:       3      28        7        2        3        1         2        20
   3:       5      30        1        2        4        1         2        20
   4:       5      30       NA        2        4        1        NA        20
   5:       5      27       NA        2        4        1         4        30
  ---
2079:       2      24       NA        2        1       NA         2        30
2080:       1      30       NA        3        1        1         1        30
2081:       2      30       NA        3        1        1         2        30
2082:       2      28       NA        2        1        1         1        30
2083:      NA      30       NA        3        1        1         1        30
      s8bq3_12                 s8bq4_a_a s8bq4_a_2                 s8bq4_b_a
   1:       NA     selling advance labor           buying from shop in debt.
   2:       NA     selling advance labor                                  NA
   3:       NA     selling advance labor           buying from shop in debt.
   4:       NA buying from shop in debt.                                   NA
   5:       NA     selling advance labor                  loan from friends
  ---
2079:       NA buying from shop in debt.                      regular income
2080:       NA            regular income   savings                        NA
2081:       NA            regular income   savings                        NA
2082:       NA            regular income   savings                        NA
2083:       NA            regular income   savings                        NA
      s8bq4_b_2                s8bq4_c_a s8bq4_c_2 s8bq4_d_a s8bq4_d_2
   1:                 loan from friends                 NA
   2:          buying from shop in debt.                 NA
   3:                 loan from friends                 NA
   4:                                NA                 NA
   5:                                NA                 NA
  ---
2079:   savings                      NA                 NA
2080:                                NA                 NA
2081:                                NA                 NA
2082:                                NA                 NA
2083:                                NA                 NA

[[2]]
          id s8bq1 s8bq2_1 s8bq2_2 s8bq2_3 s8bq2_4 s8bq2_5 s8bq2_6 s8bq2_7
```

```
   1: 7042113   yes       NA       NA       NA       NA       NA       NA       NA
   2: 7042114   yes       NA       NA       NA       NA       NA       NA       NA
   3: 7042115   yes       NA       NA       NA       NA       NA       NA       NA
   4: 7042116   yes       NA       NA       NA       NA       NA       NA       NA
   5: 7042117   yes       NA       NA       NA       NA       NA       NA       NA
  ---
2090: 7042110   yes       NA       NA       NA       NA       NA       NA       NA
2091: 7042111   yes       NA       NA       NA       NA       NA       NA       NA
2092: 7042112   yes       NA       NA       NA       NA       NA       NA       NA
2093: 7021011    no       no      yes      yes      yes       no       no      yes
2094: 7021010   yes       NA       NA       NA       NA       NA       NA       NA
      s8bq2_8 s8bq2_9 s8bq2_10 s8bq2_11 s8bq2_12 s8bq3_1 s8bq3_2 s8bq3_3
   1:      NA      NA       NA       NA       NA        3       18        2
   2:      NA      NA       NA       NA       NA        3       20        5
   3:      NA      NA       NA       NA       NA        3       18        3
   4:      NA      NA       NA       NA       NA        3       25        3
   5:      NA      NA       NA       NA       NA        3       14        2
  ---
2090:      NA      NA       NA       NA       NA        3       14        2
2091:      NA      NA       NA       NA       NA        3       23        4
2092:      NA      NA       NA       NA       NA        3       19        3
2093:      no      no      yes       no       no        3       10        3
2094:      NA      NA       NA       NA       NA        3       12        1
      s8bq3_4 s8bq3_5 s8bq3_6 s8bq3_7 s8bq3_8 s8bq3_9 s8bq3_10 s8bq3_11
   1:      12      28      NA        3       10        1         8        28
   2:      18      28       5        3       15        3        12        27
   3:      13      25      NA        3       12        2         8        28
   4:      14      28      NA        3       19        1        10        27
   5:      15      18      NA        3       12        1         9        29
  ---
2090:      17      27      NA        3       12        1        10        29
2091:      17      27       3        3       12        2        12        27
2092:      15      26       7        3       11        1        10        28
2093:      10      30      NA        2       15       NA         4        30
2094:       7      30      NA        3       10        1         6        30
      s8bq3_12      s8bq4_a_a      s8bq4_a_2         s8bq4_b_a s8bq4_b_2
   1:       NA regular income        savings                NA
   2:        2 regular income        savings                NA
   3:        0 regular income        savings                NA
   4:       NA regular income        savings                NA
   5:       NA regular income        savings                NA
  ---
2090:       NA regular income        savings                NA
2091:        1 regular income        savings                NA
2092:        2 regular income        savings                NA
2093:       NA regular income regular income loan from friends
2094:       NA regular income        do work                NA
      s8bq4_c_a s8bq4_c_2 s8bq4_d_a s8bq4_d_2
   1:        NA                  NA
   2:        NA                  NA
   3:        NA                  NA
   4:        NA                  NA
   5:        NA                  NA
  ---
2090:        NA                  NA
2091:        NA                  NA
2092:        NA                  NA
2093:        NA                  NA
2094:        NA                  NA
```

```
setwd(pathsource.mar)
```

II save as text files

Create subfolders in source folder. Following correspondence:

```
if (grepl("oct", getwd())) newfol ← c("1/additional", "1/original", "1/generated", 2:4, "
        newfol ← c("1/original", "1/additional", 2:3)
newfol ← paste0("./", newfol)
fol0 ← fol ← data.table(original = gsub("^\\.", "", foldername), new = newfol)
if (grepl("oct", getwd()))
fol0[, original := paste0(substr(original, 1, 10), "...",
        substr(original, nchar(original)-20, nchar(original)))]
#fol[, new := factor(new)]
lapply(as.list(fol[, new]), dir.create, recursive = T, showWarnings = F)
```

```
setwd(pathsource.mar)
X ← readRDS("all_data.RDS")
```

Pick files in each folder, save. I will skip the first folder (baseline additional) which use xls files.

```
folstr ← substr(fol[, original], nchar(fol[, original])-15, nchar(fol[, original]))
folstr ← gsub("\\(|_|\\)", ".", folstr)
fname0 ← asc(strsplit(fn, "^.*\\/"))
fname0 ← asc(strsplit(fname0, "\\.dta"))
fname0 ← tolower(gsub(" ", "_", fname0))
setwd(pathsource.mar)
write.to.gzip ← function(data, filename) {
        write.tablev(data, filename)
        system(paste("gzip", filename))
}
for (i in c(1, 3:4)) {
        ii ← (1:length(fn))[grepl(folstr[i], fn)]
        currfol ← paste0(getwd(), gsub("^\\.", "", fol[i, new]), "/")
        relfol ← paste0(gsub("^\\.", "", fol[i, new]), "/")
        fn0 ← paste0(currfol, fname0[ii], ".prn")
        relative.fn0 ← paste0(relfol, fname0[ii], ".prn")
        fn0dta ← gsub("prn", "dta", fn0)
        #write.to.gzip(X[ii][[3]], fn0[3])
        #system(paste("gzip", fn0[1]))
        lapply(seq_along(fn0), function(i) write.tablev(X[ii][[i]], fn0[i]))
        #lapply(seq_along(fn0), function(i) write.dta(X[[i]], fn0dta[i]))
        #zip(paste0("prn", gsub("\\.?\\/", "_", fol[i, new]), ".zip"), files = relative.fn
        #zip(paste0("dta", gsub("\\.?\\/", "_", fol[i, new]), ".zip"), files = fn0)
}
```

III read additional char households

List folder names.

```
setwd(pathreceived.mar)
foldernamea ← list.dirs(path = ".", recursive = T, full.names = T)
foldernamea ← grepout("last", foldernamea)
```

Read using XLConnect and save as tab-separated text files. Note: Memory limit binds. Some files are hand copied.

```
library(XLConnect)
setwd(pathreceived.mar)
fnx ← list.files(path = foldernamea, pattern = ".xlsx$",
        recursive = T, full.names = T)
fnxs ← gsub("^.*\\/", "", fnx)
options(java.parameters = "-Xmx8g" )
# skip section 2c as the file size is too large to read
#for (i in c(1:21, 23, 25, 27:length(fnx))) {
for (i in 31:length(fnx)) {
  gc()
  wb ← loadWorkbook(fnx[i])
  #  set "header = F" so I can read it as entries
  sc ← readWorksheet(wb, header = F, sheet = "Sheet1",
    startRow = 1, startCol = 1, endCol = 200, endRow = 3000)
  colnames(sc) ← tolower(sc[1, ])
  sc ← sc[-1, ]
  sc ← a2b(sc, "na", NA)
  write.tablev(sc, paste0(pathsource.mar, "1/additional/", gsub("xlsx", "prn", fnxs[i])))
}
```