KOSTAT-UNFPA Summer Seminar on Population

Workshop 1. Demography in R

Day 6: Data Prep for Processing and visualizing South Korean fertility microdata

Instructor: Tim Riffe
tim.riffe@ehu.eus

Assistants:
Jinyeon Jo: jyjo43043@gmail.com
Rustam Tursun-Zade: rustam.tursunzade@gmail.com

3 August 2022

1 Summary

This markdown file steps through the data prep of the microdata. Here I have the simple task of converting the (very nicely formatted) `.csv` files provided by KOSTAT into a fixed width format. I want to do this because very often when we get microdata from official statistics offices, it is delivered in fixed-width files, often public-use files where we need to to know column positions and widths in order to read the data into `R`. That is, here we take a step *backwards*! The metadata about which column is located in which position, and how wide it is is best delivered in a spreadsheet, but sometimes it is given in in `pdf` files, which can be laborious to prepare.

2 examine file

```
library(tidyverse)
library(readr)
folder_pieces <- c("Data",
```

```
                  "2000-2020_birth_annual data (for W1)_20220706",
                  "2000-2020_birth_annual data","Excel")

folder <- paste(folder_pieces,collapse="/")
files  <- dir(folder)

A <- read_csv(file.path(folder, files[1]),
              show_col_types = FALSE)
# View(A)
```

2.1 Improvise way to write fwf

Solution inspired by: Hao Zhu's GitHub gist: https://gist.githubusercontent.com/haozhu233/28d1309b58431f4929f78243054f1f58/raw/3c888574eba9bdde41b5f9081de7c9533e84f20d/write_fwf.R I have modified the function a bit to get the result we want: no headers given, this information is delivered with the column positions in a metadata file.

In real life, you will NEVER need to do what I'm doing here.

```
write_fwf = function(dt, file, width,
                     justify = "l", replace_na = "NA") {
  fct_col = which(sapply(dt, is.factor))
  if (length(fct_col) > 0) {
    for (i in fct_col) {
      dt[,i] <- as.character(dt[,i])
    }
  }
  dt[is.na(dt)] = replace_na
  n_col = ncol(dt)
  justify = unlist(strsplit(justify, ""))
  justify = as.character(factor(justify, c("l", "r"), c("-", "")))
  if (n_col != 1) {
    if (length(width) == 1) width = rep(width, n_col)
    if (length(justify) == 1) justify = rep(justify, n_col)
  }
  sptf_fmt = paste0(
    paste0("%", justify, width, "s"), collapse = ""
  )
  tbl_content = do.call(sprintf, c(fmt = sptf_fmt, dt))
  writeLines(tbl_content, file)
}
```

2.2 Write out as flat files

We now apply this function in a loop, sending the resulting text files to a folder called `Korea_births_fwf` inside `Data`.

```
# based on eyeballing the data; it's OK if these are bigger than needed; not
# OK if they are smaller than needed.
widths <- c(7,7,4,7,1,7,2,10,10,6,6)

years <- 2000:2020
```

```r
# create our metadata and save a copy for sharing
positions_names <- tibble(colname = colnames(A),
                          width = widths)
write_csv(positions_names, "Data/Korea_births_fwf_metadata.csv")

if (!file.exists("Data/Korea_births_fwf")){
  dir.create("Data/Korea_births_fwf")
}
# a loop! i is an index counter of the elements of `years`
for (i in 1:length(years)){
  file_in       <- read_csv(file.path(folder, files[i]),
                            # suppresses messages when reading in
                            show_col_types = FALSE)

  # create name for the outgoing file
  file_out_name <- paste0("kb",years[i],".txt")

  # write the file using the custom function
  write_fwf(file_in,
            file = file.path("Data/Korea_births_fwf",file_out_name),
            width = widths)
  # remove biggish file from workspace, clear memory of it
  rm(file_in);gc()
}
```

2.3 Read back in?

```r
library(vroom)

specs <- fwf_widths(widths = positions_names$width,
                                    col_names = positions_names$colname)


files <- file.path("Data/Korea_births_fwf",
                   paste0("kb",2000:2020,".txt"))
# vroom reads them all in and rbinds them together nicely :-)
A     <- vroom_fwf(files,
                   col_positions = specs)
```

```
## Rows: 9369102 Columns: 11
## -- Column specification --------------------------------------------------------
##
## dbl (11): Report year, Year, Report month, Report date, Sex, Year of birth, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# In Terminal, I typed:
# cd workspace/KOSTAT_Workshop1/Data
# zip -r Korea_births_fwf.zip Korea_births_fwf
# Also possible from R, but I didn't bother
```

2.4 Test unzip

We can unzip in session, no problem

```
unzip("Data/Korea_births_fwf.zip",

      # in case this isn't the first time
      overwrite = TRUE,

      # which directory to stash the result in
      exdir = "Data")
```

```
## Warning in unzip("Data/Korea_births_fwf.zip", overwrite = TRUE, exdir = "Data"):
## error 1 in extracting from zip file
```

Objective achieved!