Manipulating data from Statistics Canada Tables

LJ Valencia*

Introduction

This paper is is an example on how to manipulate CANSIM data from Statistics Canada. Three R packages are used in this demonstration: `tidyverse`, `cansim`, and `writexl`. The `cansim` package is very useful in retrieving data tables and time-series from Canada's socioeconomic repository, CANSIM. The `tidyverse` package is a collection of R packages designed for data science[1]. Lastly, `writexl` is a package that is used to export the dataframes in '.xlsx' format. In this demonstration, I will be using the Consumer Price Index, monthly, not seasonally adjusted[2].

Importing Packages

The relevant packages are loaded using the `library()` function. It is important to set up a file cache using the `options()` function. Although this step is optional, it is highly recommended as it would speed up data retrieval. This is because the `cansim` package will cache the data in a temporary directory for the duration of the R session.

```
# import packages
library(cansim)
library(tidyverse)
library(writexl)
```

```
options(cansim.cache_path = "StatCan4R") # Set-up File Cache
```

List of Vectors

Next, is to search for the relevant data. Statistics Canada data are accessed by using vector identifiers. This is because retrieving by table number as the package will load the entire package, putting an excessive demand on RAM and slowing down the retrieval process. Searching the desired vectors is done by using the **Statistics Canada Data Search Tool**.

To find the correct vector identifier(s), enter the table number in the "Keyword(s)" field. Then click the matching StatCan table in the search results. Then click on Add/Remove data to select which specific dimensions and customize the data. It is important to be specific and check the right boxes when customizing the table as this yields the desired vector identifier(s). I customized the data table by CPI All-items, and in the four provinces across Western Canada. Copy the relevant identifiers and paste them to a vector object.

*Bachelor of Arts (Honors) in Economics, University of Alberta
[1]For more information, see https://www.tidyverse.org/
[2]Statistics Canada. Table 18-10-0004-01 Consumer Price Index, monthly, not seasonally adjusted

Figure 1: Search by table number

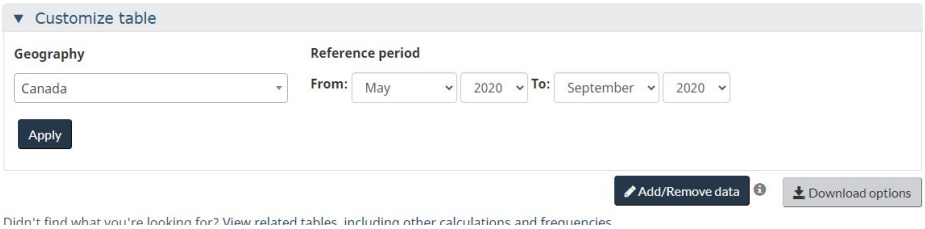

Figure 2: Click Add/Remove data

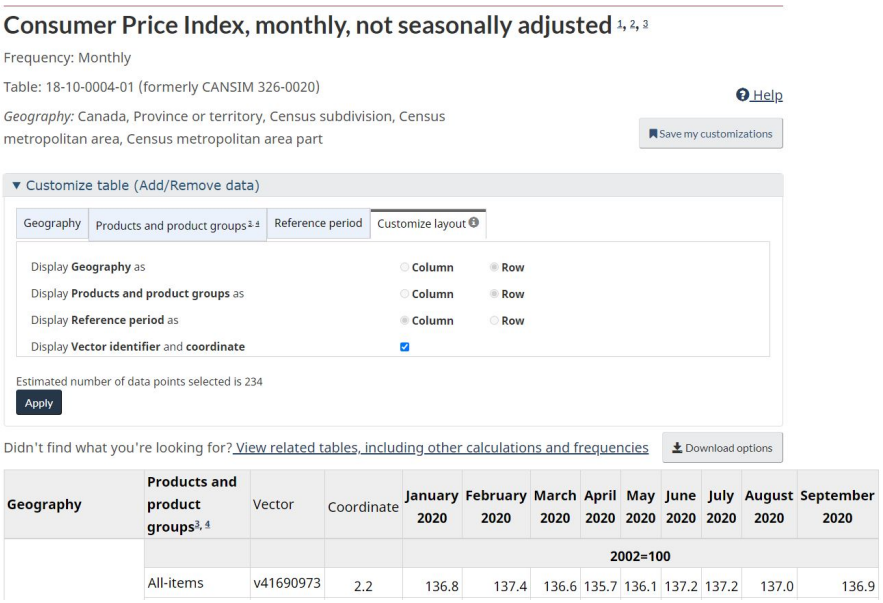

Figure 3: Check 'Display Vector indentifier and coordinate'

In the section of code below, I assigned the necessary vector identifiers into a vector object in R. Second, I set up my desired start and end periods. Next I use the `get_cansim_vector()` to retrieve the necessary tables. Next I changed the variable names in the dataframe and dropped the irrelevant columns.

```
# assign the vector identifiers into a vector
vectors.cpi <- c("v41692055",
                 "v41692191",
                 "v41692327",
                 "v41692462")
date.start <- "2000-01-01" # start period
date.end <- "2020-09-01" # end period
# retrieve tables using the vector
df <- get_cansim_vector(vectors.cpi, date.start, date.end) %>% # start and end dates
      rename(CPIallitems = VALUE) %>% # rename variables; rename VALUE into CPIallitems
      rename(date = REF_DATE) %>%
      rename(vector = VECTOR) %>%
      select(-c(1, 4:8)) # drop the irrelevant variables.
```

Spreading the Data

Although I have shown how to retrieve StatCan data using the `cansim` package, The `get_cansim_vector()` does not return a tidy data. Data points by vector indentifiers are piled on top of one another. To address this, the code below shows how to spread the data using the `spread()` function. This clean data is then exported as a '.xlsx' file.

```
df <- spread(df, key=vector, value=CPIallitems) # spread the dataframe
write_xlsx(df , path="StatCan4R\CPI-all-items.xlsx") # export data
```

Full Implementation using a Function

The code below is a full implementation, but with a function. Functions are very useful to perform specific tasks. First, I used an if-else statement to check if the vector is not a NULL type object. After using the `get_cansim_vector()` for retrieval and selection of useful information, I used a for loop to rename the vectors into more informative names. Then the data is spread using the `spread()` function. It is important to note that when creating a vector of names to replace the original identifiers, they have to match the order of their vector identifiers.

```
vectors.cpi <- c("v41692055",
                 "v41692191",
                 "v41692327",
                 "v41692462")
vectors.names <- c("Manitoba",
                 "Saskatchewan",
                 "Alberta",
                 "British Columbia")
date.start <- "2000-01-01" # start period
date.end <- "2020-09-01" # end period
# function for importing the necessary vectors
query <- function(x, y, date1, date2){
```

```
  if (is.null(x) != TRUE){# if list of vectors is not NULL (empty)
    df <- get_cansim_vector(x, date1, date2) %>% # start and end dates
      rename(totalhrs = VALUE) %>% # rename variables
      rename(date = REF_DATE) %>%
      rename(vector = VECTOR) %>%
      select(-c(1, 4:8)) # drop the irrelevant variables.
    for (i in 1:length(x)){
      df[df==x[i]] <- y[i] # replace vector numbers with actual names
    }
    df <- spread(df, key=vector, value=totalhrs) # spread the dataframe
    return(df) # return dataframe
  }
  else{ # warning statement that says inputs are not valid; 'stop' halts execution of code.
    stop("Your inputs are not valid. Please use a set of CANSIM vectors or a vector.")
  }
}

df <- query(vectors.cpi, vectors.names, date.start, date.start)
write_xlsx(df , path="StatCan4R\CPI-all-items.xlsx") # export data
```

Conclusion

I have shown how to retrieve, manipulate and clean the data from StatsCan using three libraries: `cansim`, `tidyverse`, and `writexl`. The Data Enthusiast's Blog and the `cansim` package reference guide have been been very helpful to me in creating this demonstration. Links to relevant webpages are attached in the References section.

References

Baranovsky, Petr. 2019. "Working with Statistics Canada Data in R, Part 2: Retrieving Cansim Data." https://dataenthusiast.ca/.

Ooms, Jeroen. 2020. *Writexl: Export Data Frames to Excel 'Xlsx' Format.* https://CRAN.R-project.org/package=writexl.

von Bergmann, Jens, and Dmitry Shkolnik. 2020. *Cansim: Functions and Convenience Tools for Accessing Statistics Canada Data Tables.* https://mountainmath.github.io/cansim/.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.