cchsflow: An open science approach to transform & combine population health surveys into one dataset

## Introduction

You are a public health epidemiologist who wants to examine the change in Body Mass Index (BMI) in your health unit over the past 15 years. You review the codebook for the Canadian Community Health Survey (CCHS) and note that BMI is collected. BMI *seems* like a straightforward measure that is routinely collected worldwide. Indeed, BMI is included in all CCHS cycles. You examine the documentation and find the variable HWTAGBMI in the CCHS 2001 corresponds to body mass index, but that in other cycles, the variable name changes to HWTCGBMI, HWTDGBMI, HWTEGBMI, etc. On reading the documentation, you notice that some cycles round the value to one decimal, whereas other cycles round to two digits. Furthermore, some cycles don’t calculate BMI for respondents under the age of 20 or over the age of 64 years. Also, some cycles calculate BMI only if height and weight are within specific ranges. After spending hours on the task, you talk with a collegue in a neighbouring health unit. They also did the same task a few years ago. You share your Stata code by email and compare notes, only to realize that there you both have different approaches, each with errors.

We present an open science approach to transforming and harmonizing the CCHS surveys called *cchsflow*. *cchsflow* currently harmonizes 160 variables for CCHS Public Use Microdata File (PUMF) from 2001 to 2014. *cchsflow* use of open science approach means that users can use and contribute to the package, including making suggestions and requests, and identify errors. People can even “fork” the package meaning they can use the *cchsflow* infrastructure for other purposes such as harmonizing other databases. *cchsflow* uses R language package since R is the most commonly used open statistical programming language. The core of *cchsflow*, however, are references files that could be used in other programming languages.

Even this paper was created using the open science principles that were used for *cchsflow*. We wrote the paper using R markdown - a notebook that allow R code to be executed within the document. The notebook is available on [GitHub](https://github.com/Big-Life-Lab/cchsflow/blob/paper-writeups/papers/CJPH/cjph-paper.Rmd) which allows readers to make commnents, suggestions or note errors. Readers can execute or modify all examples in this paper in R.

## Background

### Transforming & cleaning CCHS datasets

When analyzing large health datasets, it is often stated that data cleaning makes up the largest part of data analysis. According to Dasu & Johnson, 80% of data analysis is spent on data cleaning.[1] With the CCHS, similar issues arise when combining CCHS surveys for longitudinal analysis. Currently, there is no standardized method or tool used to combine CCHS survey cycles. Health units across Canada that use the CCHS do their own data cleaning and preparation, which involves hardcoding to transform and harmonize variables, taking time away from data analysis. Without a tool or system in place where transformed CCHS datasets can be accessed, time is always being taken away from analyses.

### Open science and its benefits to public health practice

Open science is defined as “transparent and accessible knowledge that is shared and developed through collaborative networks”.[2] Within this definition includes making literature that is open access (accessible to the general public), the sharing of data, andthe sharing of code.[3,4] In recent years there has been an increase in journals requiring researchers to make their research data and code publicly available. Between 2011 and 2012, the number of journals with data and code policies increased from 33% to 38% and 17% to 22% respectively.[4]

Adopting open science practices comes with many benefits. In a literature review examining open science practices, McKiernan et al. found that open science is associated with increased research exposure in both media and in citations; and an increase in collaboration, funding, job opportunities.[3] For public health professionals using health datasets such as the CCHS, open science allows for collaboration as it allows for data & coding methods to be shared between different health units. Instead of having public health professionals develop their own code to clean & transform a health dataset that has been cleaned in the past, open science opens the opportunity for users to use a transformed dataset and immediately begin the process of analysis for purposes such as surveillance and health status reporting. With a widely used dataset such as the CCHS, there is an opportunity to apply open science practices to optimize its usage for public health professionals.

## Objectives

The objectives of this project is to develop a package that transforms & harmonizes CCHS variables across multiple survey cycles, generating a library of variables that can be used for public health analysis. This package would use open science practices, in which the data & code used would be completely open; and would be open for collaboration with other public health professionals that use CCHS data. The end goal of this package is to minimize the amount of time spent cleaning and transforming CCHS datasets.

## Methods

To be able to transform & harmonize variables and data across numerous CCHS cycles, the *cchsflow* R package was created.[5] This package currently uses datasets from CCHS PUMF surveys, in which the variables of each were harmonized and transformed to use the same set of variables. In *cchsflow*, variables were renamed to the variable names used in CCHS cycles from 2007 to 2014 as they have shown to be most consistent. To develop the package, variables were first mapped across the 10 cycles.

### Mapping of variables

To transform and standardize variables into a common set of variables, CCHS variables were mapped across the 10 survey cycles in which all the names of a particular variable, their variable type (categorical or continuous), and their category structure (for categorical variables) were identified. For most variables, the only difference between cycles were their variable name; as such, only a name change was required to standardize a variable across the 10 cycles. For variables where their category structures varied between cycles, there were two options. The first option was to create a derived variable that could be used for all survey cycles. An example of this was age. In the PUMF CCHS datasets, age is a categorical variable broken into several age groups. In the 2001 and 2003 CCHS survey cycles, there were 15 age categories; while in CCHS survey cycles from 2005 to 2014, there were 16 age categories. To rectify this difference, a continuous age variable was derived that takes the midpoint of each age category for all cycles. The two grouped age variables were still specified and transformed for users interested in particular cycles of CCHS. The second option for variables with different category structures is to keep them as separate variables. In this scenario, there was no viable solution in harmonizing the variable across cycles.

### Transformation of variables through specification worksheets

Two CSV worksheets are included in the *cchsflow* packages: one that specifies all the variables in the package called *variables.csv*, and one that provides details on those variables called *variable\_details.csv*; which specify which CCHS datasets these variables come from; their starting and final variable type; and their category structure. The “rec\_with\_table” function uses the information specified in the two worksheets to create a transformed dataset from a CCHS cycle. Once all CCHS survey cycles have been transformed, they can combined using the “bind\_rows” function from the *dplyr* package [6] to create one large transformed dataset that spans across the 10 CCHS survey cycles. The two CSV worksheets also provide labels for the variables and their categories. The “rec\_with\_table” function uses the labels provided in the worksheets to label individual transformed cycles; while the “set\_data\_labels” function in *cchsflow* labels the combined dataset.

### Selection of variables

Variables from the CCHS that have been selected to be included and transformed across survey cycles fall under three categories: health behaviours, sociodemographic information, and health status. In Gochman’s handbook on health behaviour research, he defines health behaviours as “overt behavioral patterns, actions and habits that relate to health, to health restoration and to health improvement”.[7] This definition encompasses a wide variety of behaviours such as, smoking, alcohol, diet, and physical activity.[8] For this project, the aforementioned behaviours were examined using existing CCHS variables, as well as derived variables. With respect to sociodemographic information, variables such as age, sex, immigration status, country of birth, time spent in Canada, ethnicity, education (individual and highest family), income (adjusted for province and inflation), home ownership, and marital status were collected and added to the *cchsflow* library. According to Chandrakant Shah, health status can be defined as “[t]he degree to which a person is able to function physically, emotionally, socially, with or without aid from the healthcare system.”.[9] Within this definition of health status lies health conditions, human function, well-being, and deaths. Within this project, variables relating to health conditions, human function, and well-being were collected. Deaths were not collected as CCHS data included only living respondents.

### Derived variables

Derived variables were also generated in the *cchsflow* package. These variables were based on derived variables used in previous studies [10], and used transformed variables within the *cchsflow* package. If a derived variable is based on one CCHS variable, it is derived using the “rec\_with\_table” function; while derived variables using multiple CCHS variables required custom functions to be created. Similar to CCHS variables, derived variables are specified in the two worksheets, and are transformed & labelled in the same manner using “rec\_with\_table” & “set\_data\_labels”.

binge\_drinker\_fun <-  
 function(DHH\_SEX, ALW\_2A1, ALW\_2A2, ALW\_2A3, ALW\_2A4, ALW\_2A5, ALW\_2A6,  
 ALW\_2A7) {  
 # Males with at least one day with 5 or more drinks  
 if\_else2((DHH\_SEX == 1 & (ALW\_2A1 >= 5 | ALW\_2A2 >= 5 | ALW\_2A3 >=5 |  
 ALW\_2A4 >= 5 | ALW\_2A5 >= 5 | ALW\_2A6 >= 5 |  
 ALW\_2A7 >= 5)), 1,  
 # Males with no days with 5 or more drinks  
 if\_else2((DHH\_SEX == 1 & (ALW\_2A1 %in% (0:4) & ALW\_2A2 %in% (0:4) &  
 ALW\_2A3 %in% (0:4) & ALW\_2A4 %in% (0:4) &  
 ALW\_2A5 %in% (0:4) & ALW\_2A6 %in% (0:4) &  
 ALW\_2A7 %in% (0:4))), 2,  
 # Females with at least one day with 4 or more drinks  
 if\_else2((DHH\_SEX == 2 & (ALW\_2A1 >= 4 | ALW\_2A2 >= 4 | ALW\_2A3 >= 4 |  
 ALW\_2A4 >= 4 | ALW\_2A5 >= 4 | ALW\_2A6 >= 4 |  
 ALW\_2A7 >= 4)), 1,  
 # Females with no days with 4 or more drinks  
 if\_else2((DHH\_SEX == 2 & (ALW\_2A1 %in% (0:3) & ALW\_2A2 %in% (0:3) &  
 ALW\_2A3 %in% (0:3) & ALW\_2A4 %in% (0:3) &  
 ALW\_2A5 %in% (0:3) & ALW\_2A6 %in% (0:3) &  
 ALW\_2A7 %in% (0:3))), 2, NA))))  
 }

**Figure 1:** The custom function used to generate the binge drinker derived variable. This derived variable takes into account the sex of respondents, and the number of drinks they have reported to have consumed in the last 7 days to flag if a respondent is a binge drinker.

### Collaboration with other users

To promote collaboration in line with open science practices, a GitHub page has been created where users can request new variables to be transformed and added to the package. GitHub is a website where users can share their code and data repository, and is based on the Git version-control system.[11] Here, users can contribute to the *cchsflow* by adding variables directly using the specification worksheets, and create code for custom functions if they are creating derived variables. Users can also request variables to be added using the issues tab on the GitHub page.

## Results

The *cchsflow* has been developed as an R package that has been added to the Comprehensive R Archive Network (CRAN), a network of servers that contain documentation for R packages [12], and can be readily installed using the “install.packages” function in R. **Figure 3a** illustrates the command line to install the CRAN version of *cchsflow*, while **Figure 3b** illustrates the command to install the development version of *cchsflow*, which is a more up to date version of the package. As of version 1.6.0, *cchsflow* includes support for CCHS PUMF datasets from 2001 to 2014, and contains 160 variables.[5] Once combined, users have access to 1,092,951 survey respondents over a 13 year time period. The *cchsflow* package contains the following items: the *variables.csv* worksheet, the *variable\_details.csv* worksheet, the various functions, and subsets of 200 respondents for each CCHS cycle.

install.packages("cchsflow")

**Figure 3a:** The command line to install the *cchsflow* package that is currently saved on CRAN.

devtools::install\_github("Big-Life-Lab/cchsflow")

**Figure 3b:** The command line to install the development version of *cchsflow* from Github.

### Recode with table

The “rec\_with\_table” function is the primary function used to recode variables based on the information from the two specification worksheets. The function has the ability to transform an entire dataset, or a subset of variables. To use the “rec\_with\_table” function, the *cchsflow* package must be loaded in the R environment. Similarly, the CCHS datasets that are to be transformed must also be loaded in the R environment. **Figure 4a** illustrates how to load the *cchsflow* package & 2001 CCHS dataset into the R environment, and how to use “rec\_with\_table” to transform the 2001 survey cycle. **Figure 4b** illustrates how to transform a subset of variables from the 2001 survey cycle.

library(cchsflow)  
cchs2001 <- read.csv("~/data/cchs2001.csv")  
transformed\_cchs <- rec\_with\_table(cchs2001)

**Figure 4a:** The command lines to load the *cchsflow* package, load the 2001 CCHS PUMF dataset and then transform the dataset using the rec\_with\_table function.

library(cchsflow)  
cchs2001 <- read.csv("~/data/cchs2001.csv")  
transformed\_cchs2001 <- rec\_with\_table(cchs2001, c("DHH\_SEX", "DHHGAGE\_cont"))

**Figure 4b:** The command lines to load the *cchsflow* package, load the 2001 CCHS PUMF dataset and then transform the sex & age variables from the dataset using the rec\_with\_table function.

### Combining transformed datasets

Once all the desired CCHS datasets have been transformed, the datasets can now be combined to generate one dataset that consists of multiple CCHS survey cycles. Using the “bind\_rows” function from the *dplyr* package [6], multiple transformed CCHS datasets can be combined without issues. **Figure 5** illustrates how to combine multiple transformed datasets into one.

library(dplyr)  
transformed\_cycles <- bind\_rows(transformed\_cchs2001, transformed\_cchs2003)

**Figure 5:** The command lines to load the *dplyr* package, and combine transformed CCHS cycles.

### Labelling transformed datasets

The “rec\_with\_table” function adds labels to variables and their categories that specified in the specification worksheets. These labels, however, are lost when datasets are combined. The “set\_data\_labels” function is used to reattach labels that were lost during the combination process. **Figures 6a and 6b** demonstrate how to utilize the “set\_data\_labels” function. When installing the *cchsflow* package, the *sjlabelled* package is also installed.[13] This package comes with functions that are used to observe which labels are present in a transformed dataset. **Figure 6b** demonstrates how the “get\_label” function displays the labels that were added to the combined dataset.

labelled\_transformed\_cycles <- set\_data\_labels(transformed\_cycles, variable\_details, variables)

**Figure 6a:** The command line to add data labels to a newly combined dataset.

library(cchsflow)  
bmi2001 <- rec\_with\_table(cchs2001, c("HWTGHTM","HWTGWTK", "HWTGBMI\_der"))  
bmi2003 <- rec\_with\_table(cchs2003, c("HWTGHTM","HWTGWTK", "HWTGBMI\_der"))  
  
combined\_bmi <- bind\_rows(bmi2001, bmi2003)  
labelled\_combined\_bmi <- set\_data\_labels(combined\_bmi, variable\_details, variables)  
  
get\_label(labelled\_combined\_bmi)

**Figure 6b:** Using set\_data\_labels to add labels to a combined dataset. The get\_label function illustrates the labels that are in the combined dataset.

## Discussion

In this project, an R package called *cchsflow* was created that harmonizes and transforms data from 2001 to 2014 [5]. With this package, users of the CCHS now have a repository of over 1 million respondents across a 13 year period. This open science practice improves the efficiency of health data analysis as less time will be spent collecting, transforming, and recombining variables. By introducing a tool that prepares CCHS datasets for analysis, public health professionals that use CCHS data are able to spend less time on data cleaning, and spend more time on analysis such as surveillance and health status reporting. With the use of specification worksheets, users can use *cchsflow* to generate labelled CCHS datasets consisting of multiple survey cycles.

While the CCHS aims to be consistent across survey cycles, there can be differences between cycles that can sometimes be irreconcilable with a simple variable name change. Within the CCHS, there are variables that are measured differently or categorized differently. Within *cchsflow*, variables with irreconcilable differences were either transformed into a new derived variable, or kept as separate variables that can be only be used in select cycles. Along with variables with irreconciliable differences, there are variables in *cchsflow* that were not asked in all CCHS cycles. This means for some variables, data does not span across the length of the CCHS cycles available in *cchsflow*. For users looking to examine specific CCHS cycles this does not pose as an issue. For those looking to use all available cycles in *cchsflow*, further data manipulation will need to be done. A possible solution is to implement imputation, where missing data is replaced with values on based on other respondents and responses to other variables.

The open-access nature of the *cchsflow* package also allows users to add other CCHS variables that might benefit others. On the *cchsflow* GitHub [page](https://github.com/Big-Life-Lab/cchsflow/), users can either add variables, or request new variables to be transformed and added to the package. With the entire *cchsflow* repository on GitHub, there is complete transparency on how the package was developed as the entire source code for the package is publicly accessible on the repository. Along with being transparent, putting the *cchsflow* package on GitHub offers users of the package the opportunity to provide feedback on how to further improve the package. In the issues section of the GitHub repository, users can submit bug reports where they can identify issues they are encountering while using the package. Another way users can contribute to the *cchsflow* package is through the use of “pull requests”, a GitHub feature where users can request their code to be merged into the existing repository. How this works in the context of *cchsflow* is that users can create a branch off the “master” branch of *cchsflow*, make changes to the specification worksheets by adding variables and/or adding custom functions for derived variables, and then initiate a pull request to request those changes to be merged into the package. Research has shown that GitHub provides benefit to users in that it provides them an opportunity to implement better practices in their own code [11]. The implementation of GitHub in the development of *cchsflow* allows public health professionals across Canada to collaborate and share potential variables that can be useful for health surveillance and health status reporting.

### Limitations

While there are many strengths to this project, there are also several limitations. For the development of the *cchsflow* package, the Public Use Microdata File (PUMF) was used for each CCHS survey cycle. Unlike in the CCHS share data files which contain all variables, PUMF datasets remove or aggregate variables that may potentially identify a respondent [14]. In doing so, several variables that may be predictive in health data research were not included. These include socioeconomic risk factors like deprivation index, marginalization index, and urbanity/rurality of respondent and their household. While socioeconomic risk factors play a role in identifying various determinants of health [15], open science practices mean such confidential variables cannot be used or accessed in analysis.

## Conclusion

In this project, an R package called *cchsflow* was created that transforms and harmonizes variables from the Canadian Community Health Survey from 2001 to 2014, generating a dataset of over 1 million respondents. The *cchsflow* package is an open source package, and encourages the contribution & collaboration of public health professionals that use CCHS data. While there are limitations in what data can be shared due to confidentiality, open science is a vital tool in public health as it is transparent and improves the collection and preparation of data for users of health data for surveillance and health status reporting. Open science also allows public health professionals to collaborate and share their work with other colleagues, saving time recoding and cleaning health datasets. By implementing open science practices, *cchsflow* aims to minimize the amount of time needed to clean and prepare CCHS data for the many CCHS users in health units across Canada.

## References

[1] Dasu T, Johnson T. Exploratory Data Mining and Data Cleaning. 2003. <https://doi.org/10.1002/0471448354>.

[2] Vicente-Saez R, Martinez-Fuentes C. Open Science now: A systematic literature review for an integrated definition. Journal of Business Research 2018. <https://doi.org/10.1016/j.jbusres.2017.12.043>.

[3] McKiernan EC, Bourne PE, Brown CT, Buck S, Kenall A, Lin J, et al. How open science helps researchers succeed 2016. <https://doi.org/10.7554/eLife.16800>.

[4] Stodden V, Guo P, Ma Z. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. PLoS ONE 2013;8:2–9. <https://doi.org/10.1371/journal.pone.0067111>.

[5] Manuel D, Yusuf W, Vyuha R, Bennett C. Cchsflow: Transforming and harmonizing cchs variables. 2020.

[6] Wickham H, François R, Henry L, Müller K. Dplyr: A grammar of data manipulation. 2019.

[7] Gochman DS. Handbook of Health Behavior Research I. Springer US; 1997.

[8] Conner M, Norman P. Health behaviour: Current issues and challenges 2017. <https://doi.org/10.1080/08870446.2017.1336240>.

[9] Shah CP. Public Health and Preventive Medicine in Canada. 5th ed. Elsevier; 2003.

[10] Manuel DG, Perez R, Sanmartin C, Taljaard M, Hennessy D, Wilson K, et al. Measuring Burden of Unhealthy Behaviours Using a Multivariable Predictive Approach: Life Expectancy Lost in Canada Attributable to Smoking, Alcohol, Physical Inactivity, and Diet. PLoS Medicine 2016. <https://doi.org/10.1371/journal.pmed.1002082>.

[11] Dabbish L, Stuart C, Tsay J, Herbsleb J. Social coding in GitHub: Transparency and collaboration in an open software repository. In:. Proceedings of the acm conference on computer supported cooperative work, cscw, 2012. <https://doi.org/10.1145/2145204.2145396>.

[12] The Comprehensive R Network 2020.

[13] Lüdecke D. Sjlabelled: Labelled data utility functions (version 1.1.1). 2019. <https://doi.org/10.5281/zenodo.1249215>.

[14] Statistics Canada. Canadian Community Health Survey, 2014: Annual Component 2014.

[15] Bushnik T, Tjepkema M, Martel L. Socioeconomic disparities in life and health expectancy among the household population in Canada. Health Reports 2020;31:3–14. <https://doi.org/10.25318/82-003-x202000100001-eng>.