The Journal of Open Source Software

DOI: 10.21105/joss.00511

Software

- Review
- Repository
- Archive

Submitted: 14 December 2017
Published: 15 December 2017

tidyhydat: Extract and Tidy Canadian Hydrometric Data

Sam Albers[1]

1 Hydrology and Hydrometric Programs, Ministry of Environment and Climate Change Strategy, British Columbia Provincial Government

> "Tidy datasets are all alike but every messy dataset is messy in its own way -" Wickham (2014)

Introduction

Environment and Climate Change Canada (ECCC) through the Water Survey of Canada (WSC) maintains several national hydrometric data sources. These data are partially funded by provincial partners and constitute the main data products of a national integrated hydrometric network. Historical data are stored in the HYDAT database. HYDAT is the Canadian national Water Data Archive, published quarterly by the Government of Canada's Department of Environment and Climate Change. It is relational database that contains daily, monthly and annual data on water flow, water levels and sediment.

Real-time data are provided by ECCC over the web. Files are updated to a datamart on an hourly basis though the lag between actual hydrometric measurement and the availability of hydrometric data is approximately 2.5 hours. The objective of this document is the outline the usage of `tidyhydat` (Albers 2017), an R package that accesses these hydrometric data sources and *tidies* them. `tidyhydat` is part of the rOpenSci suite of packages and resides at https://github.com/ropensci/tidyhydat. The objective of `tidyhydat` is to provide a standard method of accessing ECCC data sources using a consistent and easy to use interface that employs tidy data principles developed by Wickham (2014) within the R project (R Core Team 2017).

Why use R in hydrology?

There are many statistical computing projects that offer great functionality for users. For `tidyhydat` I have chosen to use R. R is a mature open-source project that provides significant potential for advanced modelling, visualization and data manipulation. For hydrologists considering data analysis tools there are several commonly cited reasons to use R:

- R is and always will be free to use and modify.
- R is easily extensible and comprehensive. It is complimented by a rich suite of packages that implement a vast array of classical and modern statistical methods, exceptionally high-quality graphing capabilities and powerful data manipulation tools to handle a wide variety of data formats.
- R facilitates the scientific method by allowing for a fully reproducible data workflow that can be repeated by others when code is shared.
- R has a friendly community which is an important infrastructure element of any open source project.

There have been recent calls to use R more broadly in the field of hydrology (Moore and Hutchinson 2017). The `tidyhydat` package is an effort to push this call forward by being a standard package by which hydrologists and other users interact with WSC data in R. Conducting hydrological analysis in a programming environment like R allows hydrologists the ability to create fully reproducible workflows, automate repetitive tasks and provide the same rigour to the data analysis process that hydrologists apply to field equipment and experimental design (Wilson et al. 2014).

Why use tidy data?

Embedded within `tidyhydat` is the principle of *tidy data*. Wickham (2014) defines tidy data by three principles:

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit forms a table

It is illustrative here to provide an example of the types of data *tidying* processes that `tidyhydat` does for you automatically. The raw `DLY_FLOWS` table in the HYDAT database returns data that looks like this:

```
## # Source:   table<DLY_FLOWS> [?? x 73]
## # Database: sqlite 3.19.3
## #  [C:\Users\salbers\R\win-library\3.4\tidyhydat\test_db\tinyhydat.sqlite3]
##    STATION_NUMBER  YEAR MONTH FULL_MONTH NO_DAYS MONTHLY_MEAN
##             <chr> <int> <int>      <int>   <int>        <dbl>
##  1        05AA008  1910     7          0      31           NA
##  2        05AA008  1910     8          1      31         3.08
##  3        05AA008  1910     9          1      30         3.18
##  4        05AA008  1910    10          1      31         5.95
##  5        05AA008  1911     1          1      31         1.42
##  6        05AA008  1911     2          1      28         1.31
##  7        05AA008  1911     3          1      31         1.65
##  8        05AA008  1911     4          1      30         6.33
##  9        05AA008  1911     5          1      31        18.20
## 10        05AA008  1911     6          1      30        24.20
## # ... with more rows, and 67 more variables: MONTHLY_TOTAL <dbl>,
## #   FIRST_DAY_MIN <int>, MIN <dbl>, FIRST_DAY_MAX <int>, MAX <dbl>,
## #   FLOW1 <dbl>, FLOW_SYMBOL1 <chr>, FLOW2 <dbl>, FLOW_SYMBOL2 <chr>,
## #   FLOW3 <dbl>, FLOW_SYMBOL3 <chr>, FLOW4 <dbl>, FLOW_SYMBOL4 <chr>,
## #   FLOW5 <dbl>, FLOW_SYMBOL5 <chr>, FLOW6 <dbl>, FLOW_SYMBOL6 <chr>,
## #   FLOW7 <dbl>, FLOW_SYMBOL7 <chr>, FLOW8 <dbl>, FLOW_SYMBOL8 <chr>,
## #   FLOW9 <dbl>, FLOW_SYMBOL9 <chr>, FLOW10 <dbl>, FLOW_SYMBOL10 <chr>,
## #  FLOW11 <dbl>, FLOW_SYMBOL11 <chr>, FLOW12 <dbl>, FLOW_SYMBOL12 <chr>,
## #  FLOW13 <dbl>, FLOW_SYMBOL13 <chr>, FLOW14 <dbl>, FLOW_SYMBOL14 <chr>,
## #  FLOW15 <dbl>, FLOW_SYMBOL15 <chr>, FLOW16 <dbl>, FLOW_SYMBOL16 <chr>,
## #  FLOW17 <dbl>, FLOW_SYMBOL17 <chr>, FLOW18 <dbl>, FLOW_SYMBOL18 <chr>,
## #  FLOW19 <dbl>, FLOW_SYMBOL19 <chr>, FLOW20 <dbl>, FLOW_SYMBOL20 <chr>,
## #  FLOW21 <dbl>, FLOW_SYMBOL21 <chr>, FLOW22 <dbl>, FLOW_SYMBOL22 <chr>,
## #  FLOW23 <dbl>, FLOW_SYMBOL23 <chr>, FLOW24 <dbl>, FLOW_SYMBOL24 <chr>,
## #  FLOW25 <dbl>, FLOW_SYMBOL25 <chr>, FLOW26 <dbl>, FLOW_SYMBOL26 <chr>,
## #  FLOW27 <dbl>, FLOW_SYMBOL27 <chr>, FLOW28 <dbl>, FLOW_SYMBOL28 <chr>,
## #  FLOW29 <dbl>, FLOW_SYMBOL29 <chr>, FLOW30 <dbl>, FLOW_SYMBOL30 <chr>,
## #   FLOW31 <dbl>, FLOW_SYMBOL31 <chr>
```

This data structure clearly violates the principles of tidy data - this is messy data. For example, column headers (e.g. `FLOW1`) contain the day number - a value. HYDAT is structured like this for very reasonable historical reasons. It does, however, significantly limit a hydrologists ability to efficiently use hydrometric data.

tidyhydat aims to make interacting with WSC data sources simpler. I have applied tidy data principles so that users can avoid thinking about the basic data process of importing and tidying and focus on the iterative process of visualizing and modelling their data (Wickham and Grolemund 2016). After loading tidyhydat itself, we simply need to supply a station_number argument to the hy_daily_flows() function:

```r
library(tidyhydat)
hy_daily_flows(station_number = "08MF005")
```

```
## # A tibble: 37,561 x 5
##    STATION_NUMBER       Date Parameter Value Symbol
##             <chr>     <date>     <chr> <dbl>  <chr>
##  1        08MF005 1912-03-01      FLOW   538   <NA>
##  2        08MF005 1912-03-02      FLOW   538   <NA>
##  3        08MF005 1912-03-03      FLOW   538   <NA>
##  4        08MF005 1912-03-04      FLOW   538   <NA>
##  5        08MF005 1912-03-05      FLOW   538   <NA>
##  6        08MF005 1912-03-06      FLOW   538   <NA>
##  7        08MF005 1912-03-07      FLOW   479   <NA>
##  8        08MF005 1912-03-08      FLOW   479   <NA>
##  9        08MF005 1912-03-09      FLOW   459   <NA>
## 10        08MF005 1912-03-10      FLOW   459   <NA>
## # ... with 37,551 more rows
```

As you can see, this is much tidier data and is much easier to work with. In addition to these tidy principles, specific to tidyhydat, we can also define that *for a common data source, variables should be referred to by a common name*. For example, hydrometric stations are given a unique 7 digit identifier that contains important watershed information. This identifier is variously referred to as STATION_NUMBER or ID depending on the exact ECCC data source. To tidy this hydrometric data, we have renamed, where necessary, each instance of the unique identifier as STATION_NUMBER. This consistency to data formats, and in particular tidy data, situates tidyhydat well to interact seamlessly with the powerful tools being developed in the tidyverse (Wickham 2017) and provides a path in R to realize some of the goals outlined by Moore and Hutchinson (2017).

References

Albers, Sam. 2017. *Tidyhydat: Extract and Tidy Canadian Hydrometric Data*. https://github.com/ropensci/tidyhydat.

Moore, RD Dan, and David Hutchinson. 2017. "Why Watershed Analysts Should Use R for Data Processing and Analysis." *Confluence: Journal of Watershed Science and Management* 1 (1).

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10). Foundation for Open Access Statistics:1–23.

———. 2017. *Tidyverse: Easily Install and Load 'Tidyverse' Packages*. https://CRAN.R-project.org/package=tidyverse.

Wickham, Hadley, and Garrett Grolemund. 2016. "R for Data Science." Sebastopol, CA: O'Reilly. http://r4ds.had.co.nz.

Wilson, Greg, Dhavide A Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, et al. 2014. “Best Practices for Scientific Computing.” *PLoS Biology* 12 (1). Public Library of Science:e1001745.