Introduction to the dataRetrieval package

Laura De Cicco[1] and Robert Hirsch[1]

[1] *United States Geological Survey*

January 21, 2013

Contents

1 Introduction to dataRetrieval

The dataRetrieval package was created to simplify the process of getting hydrologic data in the R enviornment. It has been specifically designed to work seamlessly with the EGRET package: Exploration and Graphics for RivEr Trends (EGRET). See: https://github.com/USGS-R/EGRET/wiki for information on EGRET.

There is a plethora of hydrological data available on the web. This package is designed specifically to load United States Geological Survey (USGS) hydrologic data to the R enviornment. This includes daily values, real-time (unit values), site information, and water quality sample data.

2 Getting Started

This section describes the options for downloading and installing the dataRetrieval package.

2.1 New to R?

If you are new to R, you will need to first install the latest version of R, which can be found here: http://www.r-project.org/.

There are many options for running and editing R code, one nice enviornment to learn R is RStudio. RStudio can be downloaded here: http://rstudio.org/. Once R and RStudio are installed, the dataRetrieval package needs to be installed as described in the next section.

2.2 R User: Installing dataRetrieval from downloaded binary

The latest dataRetrieval package build is available for download at https://github.com/USGS-R/dataRetrieval/blob/master/dataRetrieval_1.2.1.tar.gz. If the package's tar.gz file is saved in R's working directory, then the following command will fully install the package:

```
> install.packages("dataRetrieval_1.2.1.tar.gz",
+                  repos=NULL, type="source")
```

If the downloaded file is stored in an alternative location, include the path in the install command. A Windows example looks like this (notice the direction of the slashes, they are in the opposite direction that Windows normally creates paths):

```
> install.packages(
+   "C:/RPackages/Statistics/dataRetrieval_1.2.1.tar.gz",
+   repos=NULL, type="source")
```

A Mac example looks like this:

```
> install.packages(
+   "/Users/userA/RPackages/Statistic/dataRetrieval_1.2.1.tar.gz",
+   repos=NULL, type="source")
```

It is a good idea to re-start the R enviornment after installing the package, especially if installing an updated version (that is, restart RStudio). Some users have found it necessary to delete the previous version's package folder before installing newer version of dataRetrieval. If you are experiencing issues after updating a package, trying deleting the package folder - the default location for Windows is something like this: C:/Users/userA/Documents/R/win-library/2.15/dataRetrieval, and the default for a Mac: /Users/userA/Library/R/2.15/library/dataRetrieval. Then, re-install the package using the directions above. Moving to CRAN should solve this problem.

2.3 R Developers: Installing dataRetrieval from gitHub

Alternatively, R-developers can install the latest version of dataRetrieval directly from gitHub using the devtools package. devtools is available on CRAN. Simpley type the following commands into R to install the latest version of dataRetrieval available on gitHub. Rtools (for Windows) and latex tools are required.

```
> library(devtools)
> install_github("dataRetrieval", "USGS-R")
```

To then open the library, simply type:

```
> library(dataRetrieval)
```

3 Raw Data: USGS Web Retrieval Examples

In this section, we will run through 4 examples, documenting how to get raw data from the web. This includes historical daily values, real-time current values, site information, and water quality data.

3.1 USGS Web Retrieval Introduction

The United States Geological Survey organizes their hydrological data in fairly standard structure. Gage stations are located throughout the United States, each station has a unique ID. Often (but not always), these ID's are 8 digits. The first step to finding data is discovering this 8-digit ID. One potential tool for discovering data is Environmental Data Discovery and Transformation (EnDDaT): http://cida.usgs.gov/enddat/. Follow the example in the User's Guide to learn how to discover USGS stations and available data from any location in the United States. Essentially, you can create a Project Location on the map, set a bounding box (in miles), then search for USGS Time Series and USGS Water Quality Data. Locations, ID's, available data, and available time periods will load on the map and appropriate tabs.

Once the site-ID is known, the next required input for USGS data retrievals is the 'parameter code'. This is a 5-digit code that specifies what measured paramater is being requested. A complete list of possible USGS parameter codes can be found here:

http://nwis.waterdata.usgs.gov/usa/nwis/pmcodes?radio_pm_search=param_group&pm_group=All+--+include+all+parameter+groups&pm_search=&casrn_search=&srsname_search=&format=html_table&show=parameter_group_nm&show=parameter_nm&show=casrn&show=srsname&show=parameter_units

Not every station will measure all parameters. The following is a list of commonly measured parameters:

Table 1: Commonly found USGS Parameter Codes

	pCode	shortName
1	00060	Discharge [cfs]
2	00065	Gage height [ft]
3	00010	Temperature [C]
4	00045	Precipitation [in]
5	00400	pH

For real-time data, the parameter code and site ID will suffice. The USGS stores historical data as daily values however. The statistical process used to store the daily data is the final requirement for daily value retrievals. A 5-digit 'stat code' specifies the requested processing. A complete list of possible USGS stat codes can be found here:

http://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table

The most common stat codes are:

Table 2: Commonly found USGS Stat Codes

	StatCode	shortName
1	00001	Maximum
2	00002	Minimum
3	00003	Mean
4	00008	Median

We will use the Choptank River near Greensboro, MD as an example. The site-ID for this gage station is 01491000. Daily discharge measurements are available as far back as 1948. Additionally, forms of nitrate and nitrogen have been measured dating back to 1964.

3.2 USGS Daily Value Retrievals

To obtain historic daily records of USGS data, use the retrieveNWISData function. The arguments for the function are siteNumber, parameterCd, startDate, endDate, statCd, and a logical (true/false) interactive. There are 2 default argument: statCd defaults to ”00003” and interactive defaults to TRUE. If you want to use the default values, you do not need to list them in the function call. Setting the ’interactive’ option to true will walk you through the function. It might make more sense to run large batch collections with the interactive option set to FALSE.

The dates (start and end) need to be in the format ”YYYY-MM-DD”. Setting the start date to ”” will indicate to the program to ask for the earliest date, setting the end date to ”” will ask for the latest available date.

```
> # Using defaults:
> siteNumber <- "01491000" # Site ID for Choptank River near Greensboro, MD
> parameterCd <- "00060"  # Discharge in cubic feet per second
> startDate <- ""
> endDate <- ""
> discharge <- retrieveNWISData(siteNumber, parameterCd, startDate, endDate)
```

A dataframe is returned that looks like the following:

```
  agency     site   dateTime value code
1   USGS 01491000 1948-01-01   190    A
2   USGS 01491000 1948-01-02   900    A
3   USGS 01491000 1948-01-03   480    A
4   USGS 01491000 1948-01-04   210    A
```

```
5   USGS 01491000 1948-01-05   210    A
6   USGS 01491000 1948-01-06   220    A
```

The structure of the dataframe is:

```
'data.frame':        23762 obs. of  5 variables:
 $ agency  : chr  "USGS" "USGS" "USGS" "USGS" ...
 $ site    : chr  "01491000" "01491000" "01491000" "01491000" ...
 $ dateTime: Date, format: "1948-01-01" "1948-01-02" ...
 $ value   : num  190 900 480 210 210 220 160 130 120 100 ...
 $ code    : chr  "A" "A" "A" "A" ...
```

Note that dateTime is imported as a Date, value is a number, and code is a string. USGS codes are often ”A” (approved for publication) or ”P” (provisional data subject to revision). A more complete list of qualification codes can be found here: `http://waterdata.usgs.gov/usa/nwis/help?codes_help`

An example that doesn't use the defaults would be a request for maximum daily temperature in early 2012:

```
> # Using defaults:
> siteNumber <- "01491000" # Site ID for Choptank River near Greensboro, MD
> parameterCd <- "00010"  # Temperature
> statCd <- "00001"
> startDate <- "2012-01-01"
> endDate <- "2012-06-30"
> temperature <- retrieveNWISData(siteNumber, parameterCd, startDate, endDate, StatCd=stat
```

3.3 USGS Unit Value Retrievals

We can also get real-time, instantaneous measurements using the retrieveUnitNWISData function:

```
> # Using defaults:
> siteNumber <- "01491000" # Site ID for Choptank River near Greensboro, MD
> parameterCd <- "00060"  # Discharge in cubic feet per second
> startDate <- as.character(Sys.Date())
> endDate <- as.character(Sys.Date())
> dischargeToday <- retrieveUnitNWISData(siteNumber, parameterCd, startDate, endDate)
```

Which produces the following dataframe:

```
  agency     site            dateTime tzone value code
1   USGS 01491000 2013-01-21 00:00:00   EST   231    P
```

```
2   USGS 01491000 2013-01-21 00:15:00   EST   231    P
3   USGS 01491000 2013-01-21 00:30:00   EST   234    P
4   USGS 01491000 2013-01-21 00:45:00   EST   231    P
5   USGS 01491000 2013-01-21 01:00:00   EST   231    P
6   USGS 01491000 2013-01-21 01:15:00   EST   228    P
```

The structure of the dataframe is:

```
'data.frame':        66 obs. of  6 variables:
 $ agency  : chr  "USGS" "USGS" "USGS" "USGS" ...
 $ site    : chr  "01491000" "01491000" "01491000" "01491000" ...
 $ dateTime: POSIXct, format: "2013-01-21 00:00:00" "2013-01-21 00:15:00" ...
 $ tzone   : chr  "EST" "EST" "EST" "EST" ...
 $ value   : num  231 231 234 231 231 228 228 228 228 228 ...
 $ code    : chr  "P" "P" "P" "P" ...
```

Note that time now becomes important, so the dateTime is a POSIXct, and the time zone is included.

3.4 USGS Site Information Retrievals

To obtain all of the available site information, use the getSiteFileData function:

```
> # Using defaults:
> siteNumber <- "01491000" # Site ID for Choptank River near Greensboro, MD
> ChopTankInfo <- getSiteFileData(siteNumber)
```

The available date for these for the USGS sites are:

```
> colnames(ChopTankInfo)
```

```
 [1] "agency.cd"              "site.no"                "station.nm"
 [4] "site.tp.cd"             "lat.va"                 "long.va"
 [7] "dec.lat.va"             "dec.long.va"            "coord.meth.cd"
[10] "coord.acy.cd"           "coord.datum.cd"         "dec.coord.datum.cd"
[13] "district.cd"            "state.cd"               "county.cd"
[16] "country.cd"             "land.net.ds"            "map.nm"
[19] "map.scale.fc"           "alt.va"                 "alt.meth.cd"
[22] "alt.acy.va"             "alt.datum.cd"           "huc.cd"
[25] "basin.cd"               "topo.cd"                "instruments.cd"
[28] "construction.dt"        "inventory.dt"           "drain.area.va"
[31] "contrib.drain.area.va" "tz.cd"                  "local.time.fg"
```

```
[34] "reliability.cd"        "gw.file.cd"             "nat.aqfr.cd"
[37] "aqfr.cd"               "aqfr.type.cd"           "well.depth.va"
[40] "hole.depth.va"         "depth.src.cd"           "project.no"
[43] "queryTime"
```

3.5 USGS Water Quality Retrievals

In this example, we use 3 dataRetrieval functions to get daily streamflow data and inorganic nitrogen sample results, and site information for a USGS gaging station with the ID 06934500. The station is Missouri River at Hermann, MO (which is discovered in the INFO dataset).

```
> Daily <- getDVData("06934500","00060","1970-10-01","2011-09-30")
```

```
There are  14975 data points, and  14975 days.
There are  0 zero flow days
If there are any zero discharge days, all days had 0 cubic meters per second added to the d
```

```
> head(Daily)
```

```
        Date        Q Julian Month Day  DecYear MonthSeq Qualifier i     LogQ
1 1970-10-01 3879.408  44102    10 274 1970.747     1450         A 1 8.263438
2 1970-10-02 3454.655  44103    10 275 1970.750     1450         A 2 8.147478
3 1970-10-03 3029.903  44104    10 276 1970.753     1450         A 3 8.016286
4 1970-10-04 2644.793  44105    10 277 1970.755     1450         A 4 7.880348
5 1970-10-05 2293.665  44106    10 278 1970.758     1450         A 5 7.737906
6 1970-10-06 2072.793  44107    10 279 1970.761     1450         A 6 7.636652
  Q7 Q30
1 NA  NA
2 NA  NA
3 NA  NA
4 NA  NA
5 NA  NA
6 NA  NA
```

```
> Sample <-getSampleData("06934500","00631","1970-10-01","2011-09-30")
> head(Sample)
```

```
        Date ConcLow ConcHigh Uncen ConcAve Julian Month Day  DecYear MonthSeq
1 1979-09-26    1.10     1.10     1    1.10  47384     9 269 1979.734     1557
2 1979-10-16    0.42     0.42     1    0.42  47404    10 289 1979.788     1558
3 1979-11-27    2.00     2.00     1    2.00  47446    11 331 1979.903     1559
```

```
4 1979-12-18    1.70     1.70     1    1.70  47467    12 352 1979.960     1560
5 1980-01-29    1.30     1.30     1    1.30  47509     1  29 1980.078     1561
6 1980-02-21    1.10     1.10     1    1.10  47532     2  52 1980.141     1562
       SinDY      CosDY
1 -0.9946999 -0.1028210
2 -0.9712570  0.2380333
3 -0.5724040  0.8199718
4 -0.2463613  0.9691781
5  0.4699767  0.8826788
6  0.7733507  0.6339785
```

```
> INFO <-getMetaData("06934500","00631", interactive=FALSE)
> colnames(INFO)
```

```
 [1] "agency.cd"             "site.no"               "station.nm"
 [4] "site.tp.cd"            "lat.va"                "long.va"
 [7] "dec.lat.va"            "dec.long.va"           "coord.meth.cd"
[10] "coord.acy.cd"          "coord.datum.cd"        "dec.coord.datum.cd"
[13] "district.cd"           "state.cd"              "county.cd"
[16] "country.cd"            "map.nm"                "map.scale.fc"
[19] "alt.va"                "alt.meth.cd"           "alt.acy.va"
[22] "alt.datum.cd"          "huc.cd"                "basin.cd"
[25] "topo.cd"               "construction.dt"       "inventory.dt"
[28] "drain.area.va"         "contrib.drain.area.va" "tz.cd"
[31] "local.time.fg"         "reliability.cd"        "project.no"
[34] "queryTime"             "drainSqKm"             "staAbbrev"
[37] "param.nm"              "param.units"           "paramShortName"
[40] "paramNumber"           "constitAbbrev"
```

```
> INFO$station.nm
```

```
[1] "Missouri River at Hermann, MO"
```

```
> Sample <- mergeReport()
```

```
 Discharge Record is 14975 days long, which is 41 years
 First day of the discharge record is 1970-10-01 and last day is 2011-09-30
 The water quality record has 437 samples
 The first sample is from 1979-09-26 and the last sample is from 2011-09-29
 Discharge: Minimum, mean and maximum 394 2660 20900
 Concentration: Minimum, mean and maximum 0.02 1.3 4.2
 Percentage of the sample values that are censored is 1.4 %
```

In the next section, we will go into detail the available functions in dataRetrieval, their required input and generated output.

4 Function Details

4.1 Daily Value Retrievals

4.2 Water Quality Retrievals

4.3 Site Information Retrievals

References

[1] Helsel, D.R. and R. M. Hirsch, 2002. Statistical Methods in Water Resources Techniques of Water Resources Investigations, Book 4, chapter A3. U.S. Geological Survey. 522 pages. http://pubs.usgs.gov/twri/twri4a3/

[2] Hirsch, R. M., Moyer, D. L. and Archfield, S. A. (2010), Weighted Regressions on Time, Discharge, and Season (WRTDS), with an Application to Chesapeake Bay River Inputs. JAWRA Journal of the American Water Resources Association, 46: 857-880. doi: 10.1111/j.1752-1688.2010.00482.x http://onlinelibrary.wiley.com/doi/10.1111/j.1752-1688.2010.00482.x/full

[3] Sprague, L. A., Hirsch, R. M., and Aulenbach, B. T. (2011), Nitrate in the Mississippi River and Its Tributaries, 1980 to 2008: Are We Making Progress? Environmental Science & Technology, 45 (17): 7209-7216. doi: 10.1021/es201221s http://pubs.acs.org/doi/abs/10.1021/es201221s