The dataRetrieval R package

By Laura A. De Cicco and Robert M. Hirsch

December 2, 2014

Contents

Figures

Tables

2

1 Introduction to dataRetrieval

The dataRetrieval package was created to simplify the process of loading hydrologic data into the R environment. It is designed to retrieve the major data types of U.S. Geological Survey (USGS) hydrologic data that are available on the Web, as well as data from the Water Quality Portal (WQP), which currently houses water quality data from the Environmental Protection Agency (EPA), U.S. Department of Agriculture (USDA), and USGS. Direct USGS data is obtained from a service called the National Water Information System (NWIS). A lot of useful information about NWIS can be obtained here:

http://help.waterdata.usgs.gov/

For information on getting started in R and installing the package, see (6): Getting Started. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

A quick workflow for USGS dataRetrieval functions:

```r
library(dataRetrieval)
# Choptank River near Greensboro, MD
siteNumber <- "01491000"
ChoptankInfo <- readNWISsite(siteNumber)
parameterCd <- "00060"

#Raw daily data:
rawDailyData <- readNWISdv(siteNumber,parameterCd,
                    "1980-01-01","2010-01-01")

# Sample data Nitrate:
parameterCd <- "00618"
qwData <- readNWISqw(siteNumber,parameterCd,
                    "1980-01-01","2010-01-01")

pCode <- readNWISpCode(parameterCd)
```

USGS data are made available through the National Water Information System (NWIS).

Table 1 describes the functions available in the dataRetrieval package.

Table 1. dataRetrieval functions

Function Name	Arguments	Description
readNWISdata	... service	NWIS data using user-specified queries
readNWISdv	siteNumber parameterCd startDate endDate statCd	NWIS daily data
readNWISqw	siteNumber parameterCd startDate endDate expanded	NWIS water quality data
readNWISuv	siteNumber parameterCd startDate endDate	NWIS water quality data
readNWISrating	siteNumber type	NWIS rating table for active streamgage
readNWISmeas	siteNumber startDate endDate	NWIS surface-water measurements
readNWISpeak	siteNumber startDate endDate	NWIS peak flow data
readNWISgwl	siteNumber startDate endDate	NWIS groundwater level measurements
readNWISpCode	parameterCd	NWIS parameter code information
readNWISsite	siteNumber	NWIS site information
whatNWISsites	...	NWIS site search using user-specified queries
whatNWISdata	siteNumber service	NWIS data availability, including period of record and count
readWQPdata	...	WQP data using user-specified queries
readWQPqw	siteNumber parameterCd (or characteristic name) startDate endDate	WQP data
whatWQPsites	...	WQP site search using user-specified queries

2 USGS Web Retrievals

In this section, examples of Web retrievals document how to get raw data. This data includes site information (2.1), measured parameter information (2.2), historical daily values(2.3), unit values (which include real-time data but can also include other sensor data stored at regular time intervals) (2.4), water quality data (2.5), groundwater level data (2.6), peak flow data (2.7), rating curve data (2.8, and surface-water measurement data (2.9). Section 5 shows instructions for getting metadata that is attached to each returned data frame.

The USGS organizes hydrologic data in a standard structure. Streamgages are located throughout the United States, and each streamgage has a unique ID (referred in this document and throughout the dataRetrieval package as "siteNumber"). Often (but not always), these ID's are 8 digits for surface-water sites and 15 digits for groundwater sites. The first step to finding data is discovering this siteNumber. There are many ways to do this, one is the National Water Information System: Mapper *http://maps.waterdata.usgs.gov/mapper/index.html*.

Once the siteNumber is known, the next required input for USGS data retrievals is the "parameter code". This is a 5-digit code that specifies the measured parameter being requested. For example, parameter code 00631 represents "Nitrate plus nitrite, water, filtered, milligrams per liter as nitrogen", with units of "mg/l as N".

A useful place to discover USGS codes information, along with other NWIS information is:

http://help.waterdata.usgs.gov/codes-and-parameters

Not every station will measure all parameters. A short list of commonly measured parameters is shown in Table 2.

Table 2. Common USGS Parameter Codes

pCode	shortName
00060	Discharge [ft^3/s]
00065	Gage height [ft]
00010	Temperature [C]
00045	Precipitation [in]
00400	pH

A complete list (as of November 7, 2014) is available as data attached to the package. It is accessed by the following:

```
library(dataRetrieval)
parameterCdFile <-  parameterCdFile
names(parameterCdFile)

[1] "parameter_cd"        "parameter_group_nm"
[3] "parameter_nm"        "casrn"
[5] "srsname"             "parameter_units"
```

Two output columns that may not be obvious are "srsname" and "casrn". Srsname stands for "Substance Registry Services". More information on the srs name can be found here:

http://ofmpub.epa.gov/sor_internet/registry/substreg/home/overview/home.do

Casrn stands for "Chemical Abstracts Service (CAS) Registry Number". More information on CAS can be found here:

http://www.cas.org/content/chemical-substances/faqs

For unit values data (sensor data measured at regular time intervals such as 15 minutes or hourly), knowing the parameter code and siteNumber is enough to make a request for data. For most variables that are measured on a continuous basis, the USGS also stores the historical data as daily values. These daily values are statistical summaries of the continuous data, e.g. maximum, minimum, mean, or median. The different statistics are specified by a 5-digit statistics code. A complete list of statistic codes can be found here:

http://help.waterdata.usgs.gov/code/stat_cd_nm_query?stat_nm_cd=%25&fmt=html&inline=true

Some common codes are shown in Table 3.

Table 3. Commonly used USGS Stat Codes

StatCode	shortName
00001	Maximum
00002	Minimum
00003	Mean
00008	Median

Examples for using these siteNumbers, parameter codes, and stat codes will be presented in subsequent sections.

2.1 Site Information

2.1.1 readNWISsite

Use the `readNWISsite` function to obtain all of the information available for a particular USGS site (or sites) such as full station name, drainage area, latitude, and longitude. `readNWISsite` can also access information about multiple sites with a vector input.

```
siteNumbers <- c("01491000","01645000")
siteINFO <- readNWISsite(siteNumbers)
```

A specific example piece of information can be retrieved, in this case a station name, as follows:

```
siteINFO$station_nm

[1] "CHOPTANK RIVER NEAR GREENSBORO, MD"
```

```
[2] "SENECA CREEK AT DAWSONVILLE, MD"
```

Site information is obtained from *http://waterservices.usgs.gov/rest/Site-Test-Tool.html*

Information on the returned data can be found with the `comment` function as described in section 5.

```
comment(siteINFO)
```

2.1.2 whatNWISdata

To discover what data is available for a particular USGS site, including measured parameters, period of record, and number of samples (count), use the `whatNWISdata` function. It is possible to limit the retrieval information to a subset of services. The possible choices for services are: `"dv"` (daily values), `"uv"`, `"rt"`, or `"iv"` (unit values), `"qw"` (water-quality), `"sv"` (sites visits), `"pk"` (peak measurements), `"gw"` (groundwater levels), `"ad"` (sites included in USGS Annual Water Data Reports External Link), `"aw"` (sites monitored by the USGS Active Groundwater Level Network External Link), and `"id"` (historical instantaneous values).

In the following example, we limit the retrieved data to only daily data. The default for `"service"` is "all", which returns all of the available data for that site. Likewise, there are arguments for parameter code (`parameterCd`) and statistic code (`statCd`) to filter the results. The default for both is to return all possible values ("all"). The returned `"count_nu"` for `"uv"` data is the count of days with returned data, not the actual count of returned values.

```
# Continuing from the previous example:
# This pulls out just the daily, mean data:

dailyDataAvailable <- whatNWISdata(siteNumbers,
                    service="dv", statCd="00003")
```

Table 4. Reformatted version of output from `whatNWISdata` function for the Choptank River near Greensboro, MD, and from Seneca Creek at Dawsonville, MD from the daily values service [Some columns deleted for space considerations]

siteNumber	srsname	startDate	endDate	count	units
01491000	Temperature, water	2010-10-01	2012-05-09	529	deg C
01645000	Stream flow, mean. daily	1930-09-26	2014-12-01	30747	ft^3/s
01491000	Stream flow, mean. daily	1948-01-01	2014-12-01	24441	ft^3/s
01491000	Specific conductance	2010-10-01	2012-05-09	527	μS/cm @25C
01491000	Suspended sediment concentration (SSC)	1980-10-01	1991-09-30	3651	mg/l
01491000	Suspended sediment discharge	1980-10-01	1991-09-30	3652	tons/day

See Section 7 for instructions on converting an R data frame to a table in Microsoft® software Excel or Word to display a data availability table similar to Table 4. Excel, Microsoft, PowerPoint, Windows, and Word are registered trademarks of Microsoft Corporation in the United States and other countries.

7

2.2 Parameter Information

To obtain all of the available information concerning a measured parameter (or multiple parameters), use the `readNWISpCode` function:

```
# Using defaults:
parameterCd <- "00618"
parameterINFO <- readNWISpCode(parameterCd)
colnames(parameterINFO)

[1] "parameter_cd"        "parameter_group_nm"
[3] "parameter_nm"        "casrn"
[5] "srsname"             "parameter_units"
```

A specific example piece of information, in this case parameter name, can be obtained as follows:

```
parameterINFO$parameter_nm

[1] "Nitrate, water, filtered, milligrams per liter as nitrogen"
```

Parameter information can obtained from:

http://help.waterdata.usgs.gov/codes-and-parameters/parameters

2.3 Daily Data

To obtain daily records of USGS data, use the `readNWISdv` function. The arguments for this function are siteNumber, parameterCd, startDate, endDate, and statCd (defaults to `"00003"`). If you want to use the default values, you do not need to list them in the function call. Daily data is pulled from *http://waterservices.usgs.gov/rest/DV-Test-Tool.html*.

The dates (start and end) must be in the format `"YYYY-MM-DD"` (note: the user must include the quotes). Setting the start date to `" "` (no space) will prompt the program to ask for the earliest date, and setting the end date to `" "` (no space) will prompt for the latest available date.

```
# Choptank River near Greensboro, MD:
siteNumber <- "01491000"
parameterCd <- "00060"  # Discharge
startDate <- "2009-10-01"
endDate <- "2012-09-30"

discharge <- readNWISdv(siteNumber,
                parameterCd, startDate, endDate)
names(discharge)

[1] "agency_cd"        "site_no"            "X_00060_00003_cd"
[4] "X_00060_00003"    "Date"
```

The column "datetime" in the returned data frame is automatically imported as a variable of class "Date" in R. Each requested parameter has a value and remark code column. The names of these columns depend on the requested parameter and stat code combinations. USGS daily value qualification codes are often "A" (approved for publication) or "P" (provisional data subject to revision). A more complete list of daily value qualification codes can be found here:

http://help.waterdata.usgs.gov/codes-and-parameters/daily-value-qualification-code-dv_rmk_cd

Another example would be a request for mean and maximum daily temperature and discharge in early 2012:

```
parameterCd <- c("00010","00060")  # Temperature and discharge
statCd <- c("00001","00003")  # Mean and maximum
startDate <- "2012-01-01"
endDate <- "2012-05-01"

temperatureAndFlow <- readNWISdv(siteNumber, parameterCd,
        startDate, endDate, statCd=statCd)
```

The column names can be shortened and simplified using the `renameNWISColumns` function. This is not necessary, but may streamline subsequent data analysis and presentation. Site information, daily statistic information, and measured parameter information is attached to the data frame as attributes. This is discussed further in section 5.

```
names(temperatureAndFlow)

[1] "agency_cd"         "site_no"          "X_00010_00001_cd"
[4] "X_00010_00001"    "X_00010_00003_cd" "X_00010_00003"
[7] "X_00060_00003_cd" "X_00060_00003"     "Date"

temperatureAndFlow <- renameNWISColumns(temperatureAndFlow)
names(temperatureAndFlow)

[1] "agency_cd"    "site_no"       "Wtemp_Max_cd"
[4] "Wtemp_Max"    "Wtemp_cd"      "Wtemp"
[7] "Flow_cd"      "Flow"          "Date"

#Information about the data frame attributes:
names(attributes(temperatureAndFlow))

[1] "names"          "row.names"     "url"
[4] "siteInfo"       "variableInfo"  "disclaimer"
[7] "statisticInfo" "queryTime"      "class"

statInfo <- attr(temperatureAndFlow, "statisticInfo")
variableInfo <- attr(temperatureAndFlow, "variableInfo")
siteInfo <- attr(temperatureAndFlow, "siteInfo")
```

An example of plotting the above data (Figure 1):

```r
variableInfo <- attr(temperatureAndFlow, "variableInfo")
siteInfo <- attr(temperatureAndFlow, "siteInfo")

par(mar=c(5,5,5,5)) #sets the size of the plot window

plot(temperatureAndFlow$Date, temperatureAndFlow$Wtemp_Max,
  ylab=variableInfo$parameter_desc[1],xlab="" )
par(new=TRUE)
plot(temperatureAndFlow$Date, temperatureAndFlow$Flow,
  col="red",type="l",xaxt="n",yaxt="n",xlab="",ylab="",axes=FALSE
  )
axis(4,col="red",col.axis="red")
mtext(variableInfo$parameter_desc[2],side=4,line=3,col="red")
title(paste(siteInfo$station_nm,"2012"))
legend("topleft", variableInfo$param_units,
       col=c("black","red"),lty=c(NA,1),pch=c(1,NA))
```

There are occasions where NWIS values are not reported as numbers, instead there might be text describing a certain event such as "Ice." Any value that cannot be converted to a number will be reported as NA in this package (not including remark code columns).
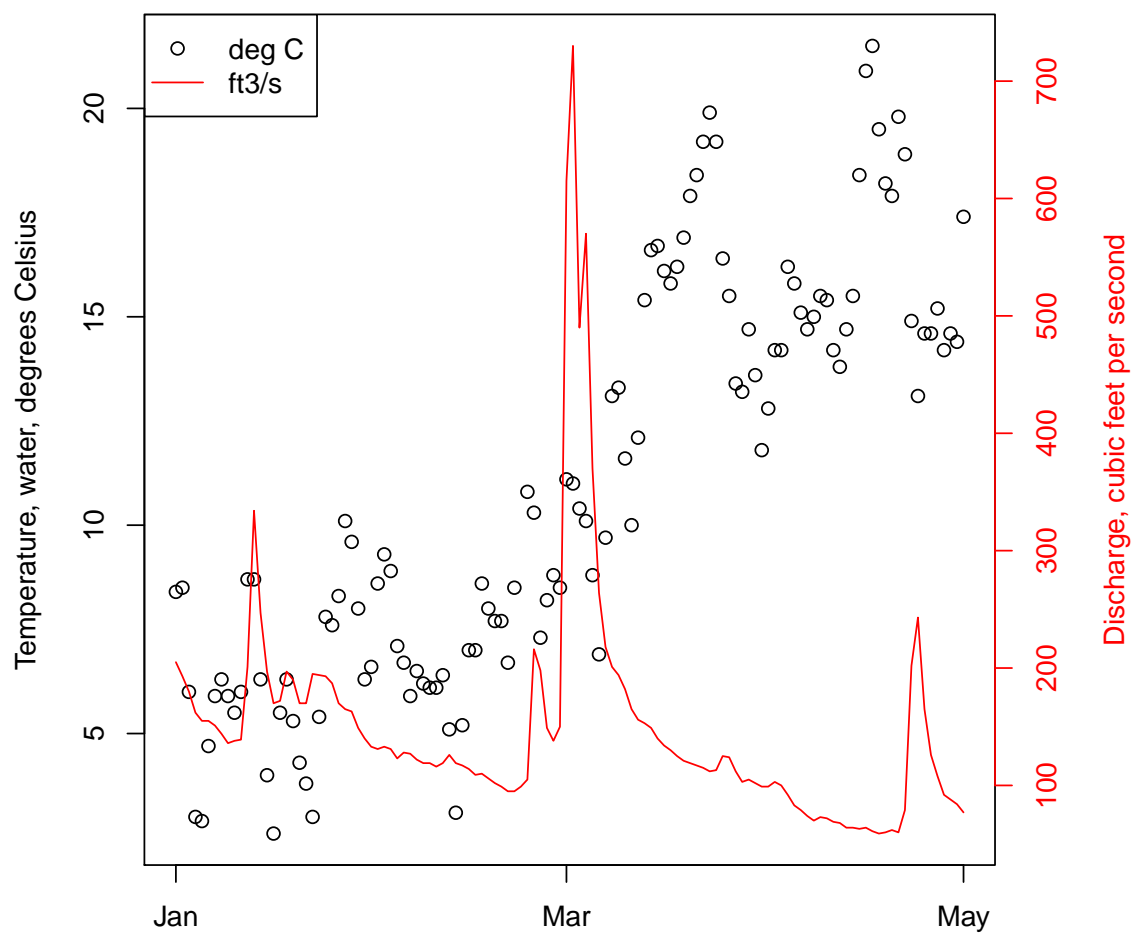
CHOPTANK RIVER NEAR GREENSBORO, MD 2012

Figure 1. Temperature and discharge plot of Choptank River in 2012.

2.4 Unit Data

Any data collected at regular time intervals (such as 15-minute or hourly) are known as "unit values." Many of these are delivered on a real time basis and very recent data (even less than an hour old in many cases) are available through the function `readNWISuv`. Some of these unit values are available for many years, and some are only available for a recent time period such as 120 days. Here is an example of a retrieval of such data.

```
parameterCd <- "00060"  # Discharge
startDate <- "2012-05-12"
endDate <- "2012-05-13"
dischargeUnit <- readNWISuv(siteNumber, parameterCd,
        startDate, endDate)
dischargeUnit <- renameNWISColumns(dischargeUnit)
```

The retrieval produces a data frame that contains 96 rows (one for every 15 minute period in the day). They include all data collected from the startDate through the endDate (starting and ending with midnight locally-collected time). The dateTime column is converted to "UTC" (Coordinated Universal Time), so midnight EST will be 5 hours earlier in the dateTime column (the previous day, at 7pm).

```
head(dischargeUnit)
  agency_cd   site_no             dateTime tz_cd Flow_Inst_cd
1     USGS 01491000 2012-05-12 05:00:00   EST              A
2     USGS 01491000 2012-05-12 05:15:00   EST              A
3     USGS 01491000 2012-05-12 05:30:00   EST              A
4     USGS 01491000 2012-05-12 05:45:00   EST              A
5     USGS 01491000 2012-05-12 06:00:00   EST              A
6     USGS 01491000 2012-05-12 06:15:00   EST              A
  Flow_Inst
1        83
2        83
3        83
4        83
5        85
6        83
```

To override the UTC timezone, specify a valid timezone in the tz argument. Default is `""`, which will keep the dateTime column in UTC. Other valid timezones are:

```
America/New_York
America/Chicago
America/Denver
America/Los_Angeles
America/Anchorage
America/Honolulu
America/Jamaica
```

```
America/Managua
America/Phoenix
America/Metlakatla
```

Data are retrieved from *http://waterservices.usgs.gov/rest/IV-Test-Tool.html*. There are occasions where NWIS values are not reported as numbers, instead a common example is "Ice." Any value that cannot be converted to a number will be reported as NA in this package. Site information and measured parameter information is attached to the data frame as attributes. This is discused further in section 5.

2.5 Water Quality Data

To get USGS water quality data from water samples collected at the streamgage or other monitoring site (as distinct from unit values collected through some type of automatic monitor) we can use the function readNWISqw, with the input arguments: siteNumber, parameterCd, startDate, and endDate. Additionally, the argument "expanded" is a logical input that allows the user to choose between a simple return of datetimes/qualifier/values (expanded=FALSE), or a more complete and verbose output (expanded=TRUE). Expanded = TRUE includes such columns as remark codes, value qualifying text, and detection level for each parameter code. There also includes an argument "reshape", that converts the expanded dataset to a "wide" format (each requested parameter code gets individual columns). The defaults are expanded=TRUE, and reshape=FALSE.

```r
# Dissolved Nitrate parameter codes:
parameterCd <- c("00618","71851")
startDate <- "1985-10-01"
endDate <- "2012-09-30"

dfLong <- readNWISqw(siteNumber, parameterCd,
      startDate, endDate)

# Or the wide return:
# dfWide <- readNWISqw(siteNumber, parameterCd,
#       startDate, endDate, reshape=TRUE)
```

Site information and measured parameter information is attached to the data frame as attributes. This is discused further in section 5. Additional metadata, such as information about the column names can be found by using the comment function, also described in section 5.

```r
comment(dfLong)
```

2.6 Groundwater Level Data

Groundwater level measurements can be obtained with the readNWISgwl function. Information on the returned data can be found with the comment function, and attached attributes as described in section 5.

```r
siteNumber <- "434400121275801"
groundWater <- readNWISgwl(siteNumber)

names(groundWater)

[1] "agency_cd"          "site_no"
[3] "dateTime"           "dateTimeAccuracyCd"
[5] "censorCode"         "sourceCode"
[7] "X_72019_00011_cd"   "X_72019_00011"
```

14

2.7 Peak Flow Data

Peak flow data are instantaneous discharge or stage data that record the maximum values of these variables during a flood event. They include the annual peak flood event but can also include records of other peaks that are lower than the annual maximum. Peak discharge measurements can be obtained with the readNWISpeak function. Information on the returned data can be found with the comment function and attached attributes as described in section 5.

```
siteNumber <- '01594440'
peakData <- readNWISpeak(siteNumber)


names(peakData)

 [1] "agency_cd"      "site_no"        "peak_dt"
 [4] "peak_tm"        "peak_va"        "peak_cd"
 [7] "gage_ht"        "gage_ht_cd"     "year_last_pk"
[10] "ag_dt"          "ag_tm"          "ag_gage_ht"
[13] "ag_gage_ht_cd"
```

2.8 Rating Curve Data

Rating curves are the calibration curves that are used to convert measurements of stage to discharge. Because of changing hydrologic conditions these rating curves change over time. Information on the returned data can be found with the comment function and attached attributes as described in section 5.

Rating curves can be obtained with the readNWISrating function.

```
ratingData <- readNWISrating(siteNumber, "base")
attr(ratingData, "RATING")

[1] "ID=19.0"             "TYPE=STGQ"
[3] "NAME=stage-discharge"  "AGING=W"
[5] "REMARKS="            "EXPANSION=logarithmic"
[7] "OFFSET1=0.2000000E+01"

names(ratingData)

[1] "INDEP" "DEP"   "STOR"
```

2.9 Surface-Water Measurement Data

These data are the discrete measurements of discharge that are made for the purpose of developing or revising the rating curve. Information on the returned data can be found with the comment function

and attached attributes as described in section 5.

Surface-water measurement data can be obtained with the `readNWISmeas` function.

```
surfaceData <- readNWISmeas(siteNumber)

names(surfaceData)

 [1] "agency_cd"              "site_no"
 [3] "measurement_nu"         "measurement_dt"
 [5] "tz_cd"                  "q_meas_used_fg"
 [7] "party_nm"               "site_visit_coll_agency_cd"
 [9] "gage_height_va"         "discharge_va"
[11] "current_rating_nu"      "shift_adj_va"
[13] "diff_from_rating_pc"    "measured_rating_diff"
[15] "gage_va_change"         "gage_va_time"
[17] "control_type_cd"        "discharge_cd"
```

3 Water Quality Portal Web Retrievals

There are additional water quality data sets available from the Water Quality Data Portal (*http://www.waterqualitydata.us/*). These data sets can be housed in either the STORET database (data from EPA), NWIS database (data from USGS), STEWARDS database (data from USDA), and additional databases are slated to be included in the future. Because only USGS uses parameter codes, a "characteristic name" must be supplied. The `readWQPqw` function can take either a USGS parameter code, or a more general characteristic name in the parameterCd input argument. The Water Quality Data Portal includes data discovery tools and information on characteristic names. The following example retrieves specific conductance from a DNR site in Wisconsin.

```
specificCond <- readWQPqw('WIDNR_WQX-10032762',
                'Specific conductance','2011-05-01','2011-09-30')
```

A tool for finding NWIS characteristic names can be found at: *http://www.waterqualitydata.us/public_srsnames.jsp*.

4 Generalized Retrievals

The previous examples all took specific input arguments: siteNumber, parameterCd (or characteristic name), startDate, endDate, etc. However, the Web services that supply the data can accept a wide variety of additional arguments.

4.0.1 NWIS Sites

The function `whatNWISsites` can be used to discover NWIS sites based on any query that the NWIS Site Service offers. This is done by using the `"..."` argument, which allows the user to use any arbitrary input argument. We can then use the service here:

http://waterservices.usgs.gov/rest/Site-Test-Tool.html

to discover many options for searching for NWIS sites. For example, you may want to search for sites in a lat/lon bounding box, or only sites tidal streams, or sites with water quality samples, sites above a certain altitude, etc. The results of this site query generate a URL. For example, the tool provided a search within a specified bounding box, for sites that have daily discharge (parameter code = 00060) and temperature (parameter code = 00010). The generated URL is:

http://waterservices.usgs.gov/nwis/site/?format=rdb&bBox=-83.0,36.5,-81.0,38.5&parameterCd=00010, 00060&hasDataTypeCd=dv

The following dataRetrieval code can be used to get those sites:

```
sites <- whatNWISsites(bBox="-83.0,36.5,-81.0,38.5",
                       parameterCd="00010,00060",
                       hasDataTypeCd="dv")

names(sites)

[1] "agency_cd"    "site_no"      "station_nm"  "site_tp_cd"
[5] "dec_lat_va"   "dec_long_va"  "queryTime"

nrow(sites)

[1] 205
```

4.0.2 NWIS Data

For NWIS data, the function `readNWISdata` can be used. The argument listed in the R help file is `"..."` and `"service"` (only for data requests). Table 5 describes the services are available.

Table 5. NWIS general data calls

Service	Description	Reference URL
daily values	dv	*http://waterservices.usgs.gov/rest/DV-Test-Tool.html*
instantaneous	iv	*http://waterservices.usgs.gov/rest/IV-Test-Tool.html*
groundwater levels	gwlevels	*http://waterservices.usgs.gov/rest/GW-Levels-Test-Tool.html*
water quality	qwdata	*http://nwis.waterdata.usgs.gov/nwis/qwdata*

The `"..."` argument allows the user to create their own queries based on the instructions found in the web links above. The links provide instructions on how to create a URL to request data. Perhaps you

want sites only in Wisconsin, with a drainage area less than 50 mi^2, and the most recent daily dischage data. That request would be done as follows:

```
dischargeWI <- readNWISdata(service="dv",
                            stateCd="WI",
                            parameterCd="00060",
                            drainAreaMin="50",
                            statCd="00003")
names(dischargeWI)

[1] "agency_cd"        "site_no"          "dateTime"
[4] "tz_cd"            "X_00060_00003_cd" "X_00060_00003"

nrow(dischargeWI)

[1] 282

siteInfo <- attr(dischargeWI, "siteInfo")
head(siteInfo)

                       station_nm  site_no agency_cd
1     AMNICON RIVER NEAR POPLAR, WI 04025000      USGS
2     APPLE RIVER NEAR SOMERSET, WI 05341500      USGS
3          BAD RIVER AT MELLEN, WI 04026500      USGS
4        BAD RIVER NEAR MELLEN, WI 04026450      USGS
5        BAD RIVER NEAR ODANAH, WI 04027000      USGS
6 BADFISH CREEK NEAR COOKSVILLE, WI 05430150      USGS
  timeZoneOffset timeZoneAbbreviation dec_lat_va dec_lon_va
1         -06:00                  CST   46.60411  -91.88908
2         -06:00                  CST   45.15750  -92.71639
3         -06:00                  CST   46.32495  -90.65991
4         -06:00                  CST   46.27056  -90.70722
5         -06:00                  CST   46.48661  -90.69630
6         -06:00                  CST   42.83334  -89.19678
       srs siteTypeCd stateCd countyCd    hucCd
1 EPSG:4326         ST      55    55031 04010301
2 EPSG:4326         ST      55    55109 07030005
3 EPSG:4326         ST      55    55003 04010302
4 EPSG:4326         ST      55    55003 04010302
5 EPSG:4326         ST      55    55003 04010302
6 EPSG:4326         ST      55    55105 07090001
```

4.0.3 WQP Sites

Just as with NWIS, the Water Quality Portal (WQP) offers a variety of ways to search for sites and request data. The possible Web service arguments for WQP site searches is found here:

http://www.waterqualitydata.us/webservices_documentation.jsp

To discover available sites in the WQP in New Jersey that have measured Chloride, use the function whatWQPsites.

```
sitesNJ <- whatWQPsites(statecode="US:34",
                        characteristicName="Chloride")
```

4.0.4 WQP Data

Finally, to get data from the WQP using generalized Web service calls, use the function readWQPdata. For example, to get all the pH data in Wisconsin:

```
dataPH <- readWQPdata(statecode="US:55",
                      characteristicName="pH")
```

5 Embedded Metadata

All data frames returned from the Web services have some form of associated metadata. This information is included as attributes to the data frame. All data frames will have a `url` (returning a character of the url used to obtain the data), `siteInfo` (returning a data frame with information on sites), and `queryTime` (returning a POSIXct datetime) attributes. For example, the url and query time used to obtain the data can be found as follows:

```
attr(dischargeWI, "url")

[1] "http://waterservices.usgs.gov/nwis/dv/?format=waterml&stateCd=WI&parameter

attr(dischargeWI, "queryTime")

[1] "2014-12-02 15:32:14 CST"

siteInfo <- attr(dischargeWI, "siteInfo")
```

Depending on the format that the data was obtained (RDB, WaterML1, etc), there will be additional information embedded in the data frame as attributes. To discover the available attributes:

```
names(attributes(dischargeWI))

[1] "names"        "row.names"     "class"
[4] "url"          "siteInfo"      "variableInfo"
[7] "disclaimer"   "statisticInfo" "queryTime"
```

For data obtained from `readNWISuv`, `readNWISdv`, `readNWISgwl` there are two attributes that are particularly useful: `siteInfo` and `variableInfo`.

```
siteInfo <- attr(dischargeWI, "siteInfo")
head(siteInfo)

                       station_nm  site_no agency_cd
1     AMNICON RIVER NEAR POPLAR, WI 04025000      USGS
2     APPLE RIVER NEAR SOMERSET, WI 05341500      USGS
3          BAD RIVER AT MELLEN, WI 04026500      USGS
4        BAD RIVER NEAR MELLEN, WI 04026450      USGS
5        BAD RIVER NEAR ODANAH, WI 04027000      USGS
6 BADFISH CREEK NEAR COOKSVILLE, WI 05430150      USGS
  timeZoneOffset timeZoneAbbreviation dec_lat_va dec_lon_va
1         -06:00                  CST   46.60411  -91.88908
2         -06:00                  CST   45.15750  -92.71639
3         -06:00                  CST   46.32495  -90.65991
4         -06:00                  CST   46.27056  -90.70722
5         -06:00                  CST   46.48661  -90.69630
6         -06:00                  CST   42.83334  -89.19678
        srs siteTypeCd stateCd countyCd    hucCd
```

```
1 EPSG:4326          ST      55      55031 04010301
2 EPSG:4326          ST      55      55109 07030005
3 EPSG:4326          ST      55      55003 04010302
4 EPSG:4326          ST      55      55003 04010302
5 EPSG:4326          ST      55      55003 04010302
6 EPSG:4326          ST      55      55105 07090001

variableInfo <- attr(dischargeWI, "variableInfo")
```

Data obtained from `readNWISpeak`, `readNWISmeas`, and `readNWISrating`, the `comment` attribute is useful.

```
comment(peakData)

#Which is equivalent to:
# attr(peakData, "comment")
```

A subset (due to space considerations) of the `comment` metatdata is shown here:

```
comment(peakData)[c(1:15,58:66)]

 [1] "#"
 [2] "# U.S. Geological Survey"
 [3] "# National Water Information System"
 [4] "# Retrieved: 2014-12-02 16:31:54 EST"
 [5] "#"
 [6] "# --------------------WARNING--------------------"
 [7] "# The data you have obtained from this automated"
 [8] "# U.S. Geological Survey database have not received"
 [9] "# Director's approval and as such are provisional"
[10] "# and subject to revision.  The data are released"
[11] "# on the condition that neither the USGS nor the"
[12] "# United States Government may be held liable for"
[13] "# any damages resulting from its use."
[14] "#"
[15] "# More data may be available offline."
[16] "# Gage height qualification codes(gage_ht_cd,ag_gage_ht_cd):"
[17] "#   1 ... Gage height affected by backwater"
[18] "#   2 ... Gage height not the maximum for the year"
[19] "#   3 ... Gage height at different site and(or) datum"
[20] "#   4 ... Gage height below minimum recordable elevation"
[21] "#   5 ... Gage height is an estimate"
[22] "#   6 ... Gage datum changed during this year"
[23] "#"
[24] "#"
```

6 Getting Started in R

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

6.1 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/*.

At any time, you can get information about any function in R by typing a question mark before the functions name. This will open a file (in RStudio, in the Help window) that describes the function, the required arguments, and provides working examples. This will open a help file similar to Figure 2. To see the raw code for a particular code, type the name of the function, without parentheses.

```
?readNWISpCode
```

Additionally, many R packages have vignette files attached (such as this paper). To view the vignette:
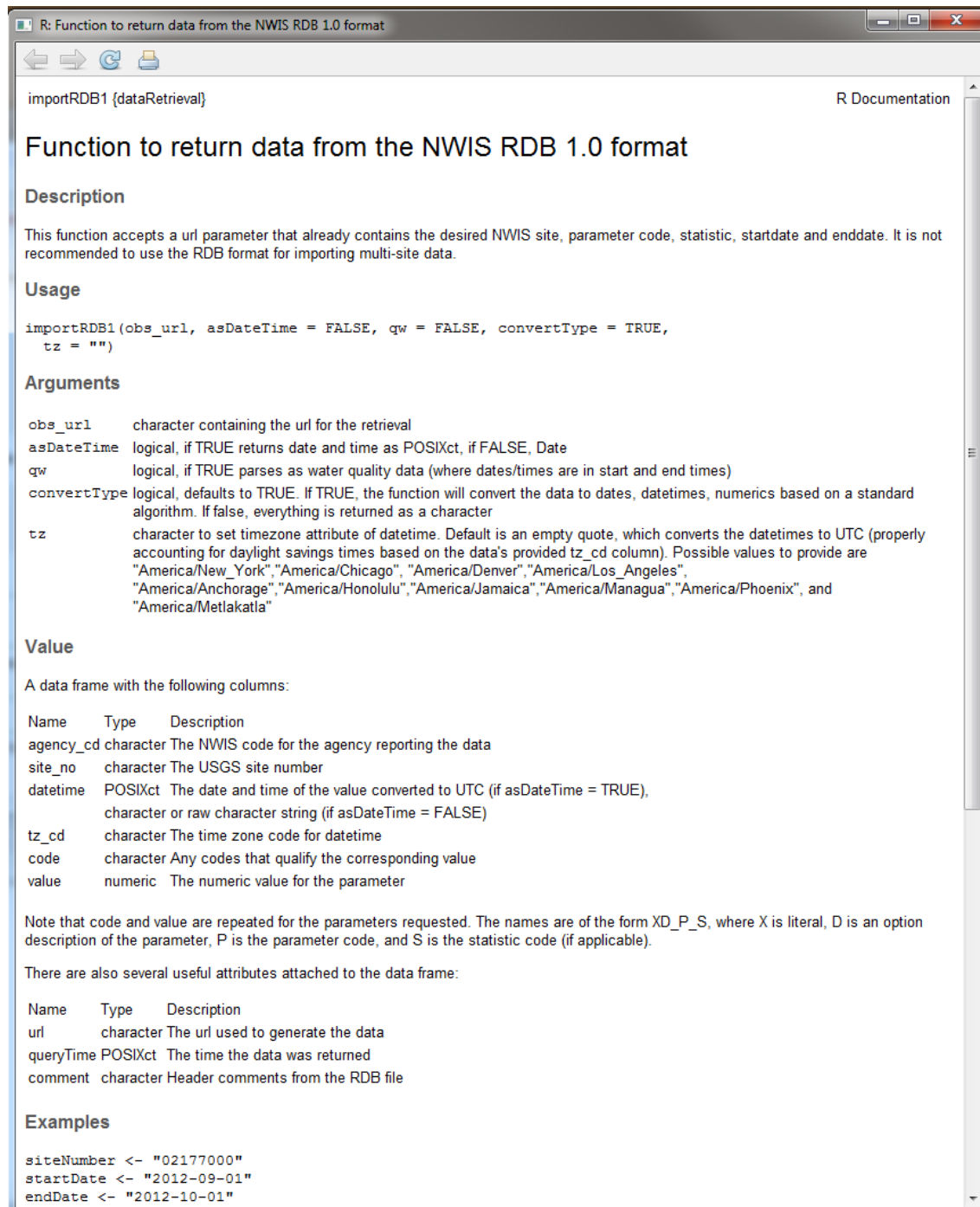
```
vignette(dataRetrieval)
```

R: Function to return data from the NWIS RDB 1.0 format

importRDB1 {dataRetrieval}   R Documentation

Function to return data from the NWIS RDB 1.0 format

Description

This function accepts a url parameter that already contains the desired NWIS site, parameter code, statistic, startdate and enddate. It is not recommended to use the RDB format for importing multi-site data.

Usage

```
importRDB1(obs_url, asDateTime = FALSE, qw = FALSE, convertType = TRUE,
  tz = "")
```

Arguments

obs_url	character containing the url for the retrieval
asDateTime	logical, if TRUE returns date and time as POSIXct, if FALSE, Date
qw	logical, if TRUE parses as water quality data (where dates/times are in start and end times)
convertType	logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character
tz	character to set timezone attribute of datetime. Default is an empty quote, which converts the datetimes to UTC (properly accounting for daylight savings times based on the data's provided tz_cd column). Possible values to provide are "America/New_York","America/Chicago", "America/Denver","America/Los_Angeles", "America/Anchorage","America/Honolulu","America/Jamaica","America/Managua","America/Phoenix", and "America/Metlakatla"

Value

A data frame with the following columns:

Name	Type	Description
agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number
datetime	POSIXct	The date and time of the value converted to UTC (if asDateTime = TRUE), character or raw character string (if asDateTime = FALSE)
tz_cd	character	The time zone code for datetime
code	character	Any codes that qualify the corresponding value
value	numeric	The numeric value for the parameter

Note that code and value are repeated for the parameters requested. The names are of the form XD_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable).

There are also several useful attributes attached to the data frame:

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file

Examples

```
siteNumber <- "02177000"
startDate <- "2012-09-01"
endDate <- "2012-10-01"
```

Figure 2. A simple R help file

23

6.2 R User: Installing dataRetrieval

The following command installs dataRetrieval and subsequent required packages:

```r
install.packages("dataRetrieval")
```

After installing the package, you need to open the library each time you re-start R. This is done with the simple command:

```r
library(dataRetrieval)
```

7 Creating Tables in Microsoft® Software from R

There are a few steps that are required in order to create a table in Microsoft® software (Excel, Word, PowerPoint, etc.) from an R data frame. There are certainly a variety of good methods, one of which is detailed here. The example we will step through here will be to create a table in Microsoft Excel based on the data frame tableData:

```r
availableData <- whatNWISdata(siteNumber, "dv")
dailyData <- availableData["00003" == availableData$stat_cd,]

tableData <- with(dailyData,
      data.frame(
        shortName=srsname,
        Start=begin_date,
        End=end_date,
        Count=count_nu,
        Units=parameter_units)
      )
tableData

                                shortName      Start
1                Stream flow, mean. daily 1977-06-27
2 Suspended sediment concentration (SSC) 1984-10-27
3          Suspended sediment discharge 1984-10-27
        End Count    Units
1 2014-12-01 13671    ft3/s
2 1991-09-30  2164     mg/l
3 1991-09-30  1799 tons/day
```

First, save the data frame as a tab delimited file (you don't want to use comma delimited because there are commas in some of the data elements):

```r
write.table(tableData, file="tableData.tsv",sep="\t",
            row.names = FALSE,quote=FALSE)
```

This will save a file in your working directory called tableData.tsv. You can see your working directory by typing getwd() in the R console. Opening the file in a general-purpose text editor, you should see the following:

```
shortName    Start   End Count Units
Temperature, water 2010-10-01 2012-06-24 575 deg C
Stream flow, mean. daily 1948-01-01 2013-03-13 23814 ft3/s
Specific conductance 2010-10-01 2012-06-24 551 uS/cm @25C
Suspended sediment concentration (SSC) 1980-10-01 1991-09-30 3651 mg/l
Suspended sediment discharge 1980-10-01 1991-09-30 3652 tons/day
```

Next, follow the steps below to open this file in Excel:

1. Open Excel

2. Click on the File tab

3. Click on the Open option

4. Navigate to the working directory (as shown in the results of getwd())

5. Next to the File name text box, change the dropdown type to All Files (*.*)

6. Double click tableData.tsv

7. A text import wizard will open up, in the first window, choose the Delimited radio button if it is not automatically picked, then click on Next.

8. In the second window, click on the Tab delimiter if it is not automatically checked, then click Finished.

9. Use the many formatting tools within Excel to customize the table

From Excel, it is simple to copy and paste the tables in other Microsoft® software. An example using one of the default Excel table formats is here. Additional formatting could be requried in Excel, for example converting u to μ.

shortName	Start	End	Count	Units
Temperature, water	10/1/2010	6/24/2012	575	deg C
Stream flow, mean. daily	1/1/1948	3/13/2013	23814	cfs
Specific conductance	10/1/2010	6/24/2012	551	uS/cm @25C
Suspended sediment concentration (SSC)	10/1/1980	9/30/1991	3651	mg/l
Suspended sediment discharge	10/1/1980	9/30/1991	3652	tons/day

Figure 3. A simple table produced in Microsoft® Excel.

8 Disclaimer

This information is preliminary and is subject to revision. It is being provided to meet the need for timely best science. The information is provided on the condition that neither the U.S. Geological Survey nor the U.S. Government may be held liable for any damages resulting from the authorized or unauthorized use of the information.