

Working with Water-Quality Data

Dave Lorenz

December 30, 2015

Abstract

These examples demonstrate some of the functions and methods for importing, managing, and manipulating water-quality data that are available in the **smwrQW** package.

Contents

1	Introduction	2
2	Importing Water-Quality Data	3
3	Arithmetic Operations	8
4	Comparison Operations	10
5	Miscellaneous Manipulations	12
6	Conversions for Analysis	16
7	Mathematical Transformations	21
8	Other Applications	22

1 Introduction

The class "qw" provides a mechanism for storing water-quality data that facilitates managing and analyzing those data. The information stored in class "qw" includes quantitative and qualitative data. The quantitative data pertain to assigning values to the data and is used to convert the data to either class "lcens," for un- or left-censored values, or "mcens," for any other or mixture of censoring, that are required for the analytic functions in **smwrQW**. The quantitative data are in slots named ".Data" that is a numeric matrix of two columns, required to store data that are potentially interval censored; "remark.codes" that provide additional information about the type of censoring; and "reporting.level" that records the censoring level that will be used for each value when converted to either class "lcens" or "mcens." The qualitative data are provided for the user to better interpret the data. The qualitative data are in slots named "value.codes" that contains additional information about each value, possibly supplied by the analyzing lab; "reporting.method" that contains any information about the method for computing the "reporting.level"; "reporting.units" that contains the concentration units and can be used by some analytic routines if necessary; "analyte.method" that contains the method code for the analytic method; "analyte.name" that contains the name of the constituent; "rounding" is a two element vector that rounds the data for printing; "unique.code" that provide a mechanism for distinguishing between various types of data, equivalent to the parameter code; and "names" that is used only internally. The qualitative data are hereinafter referred to as meta data.

The two column matrix in the slot named ".Data" provides a mechanism for storing censored water-quality data when combined with the censoring information in the slot named "remark.codes." Quantified values are stored with the same value in the two columns and a remark code set to "" (the empty character string) or " " (a single blank character). So called less-than values are stored with 0 in the first column and the reported upper limit of concentration in the second column and the remark code must be "<." Right-censored values are stored as the minimum value in the first column, positive infinity in the second column, and the remark code must be ">." Interval-censored values are stored with the minimum value of in the first column and the maximum value in the second column; the remark code can be set to "I," but either of the valid strings for quantified values are permitted. Invalid remark codes are permitted, but can generate warnings under certain circumstances. This approach for storing data facilitates easy mathematical manipulation and conversion to forms that can easily be analyzed.

```
> # Load the smwrQW package
> library(smwrQW)
> # print the slot names of the class "qw"
> slotNames("qw")
```

```
[1] ".Data"          "remark.codes"    "value.codes"     "reporting.level"
[5] "reporting.method" "reporting.units" "analyte.method"  "analyte.name"
[9] "rounding"       "unique.code"     "names"
```

2 Importing Water-Quality Data

Data from the U.S. Geological Survey (USGS) NWISWeb can easily be imported into data frames using the `importNWISqw` function. The `importNWISqw` function requires at least one USGS station identifier at a minimum and optional parameter codes or a parameter group code, and starting and ending dates for the retrieval. See the documentation for `importNWISqw` for more information. The code following this paragraph retrieves a small data set of ammonia, parameter code "00608" for a single USGS station identifier 0531656290, West Fork Beaver Creek at 320 St. near Bechyn, Minn. and more complete nutrient data for USGS station identifier 05320270, Little Cobb River near Beauford, Minn.

```
> # get the data
> WFBC.NH3 <- importNWISqw("0531656290", "00608", end.date="2006-09-30")
> # print the structure
> str(WFBC.NH3)

'data.frame':      9 obs. of  8 variables:
 $ site_no      : chr  "0531656290" "0531656290" "0531656290" "0531656290" ...
 $ sample_dt    : Date, format: "2005-10-26" "2005-10-26" ...
 $ sample_tm    : chr  "14:00" "15:45" "16:00" "10:17" ...
 $ tzone_cd     : chr  "CDT" "CDT" "CST" "CDT" ...
 $ medium_cd    : chr  "WS" "WS" "WS" "WS" ...
 $ sample_end_dt: Date, format: NA NA ...
 $ sample_end_tm: chr  "" "" "16:00" "08:18" ...
 $ Ammonia.N    : qw, left censoring: "<0.040" "<0.040" ...

> # Now get the nutrient data for Little Cobb
> LCobb.nuts <- importNWISqw("05320270", "NUT")
```

Data from other sources can come in a wide variety of formats. One source is the Water Quality Portal (WQP), a cooperative service sponsored by the USGS, the Environmental Protection Agency, and the National Water Quality Monitoring Council. It serves data collected by over 400 state, federal, tribal, and local agencies. Data stored in the WQP can be retrieved using the `readWQPqw` function in the **dataRetrieval** package, as shown in the code following this paragraph.

```
> # get the data
> WFBC.wqp <- readWQPqw("USGS-0531656290", "00608", endDate="2006-09-30")
> # print the structure, note that the output is captured and modified to fit
> # in a narrow output; the additional attribute data frames are also stripped
> Tmp <- capture.output(str(data.frame(WFBC.wqp), vec.len=1))
> Tmp <- sapply(Tmp, sub, pattern=":", replacement="\n  ", fixed=TRUE)
> cat(Tmp, sep="\n")

'data.frame'
      9 obs. of  67 variables:
 $ OrganizationIdentifier
   chr  "USGS-MN" ...
 $ OrganizationFormalName
   chr  "USGS Minnesota Water Science Center" ...
```

```

$ ActivityIdentifier
  chr "nwismn.01.00600014" ...
$ ActivityTypeCode
  chr "Sample-Routine" ...
$ ActivityMediaName
  chr "Water" ...
$ ActivityMediaSubdivisionName
  chr "Surface Water" ...
$ ActivityStartDate
  Date, format: "2005-10-26" ...
$ ActivityStartTime.Time
  chr "14:00:00" ...
$ ActivityStartTime.TimeZoneCode
  chr "CDT" ...
$ ActivityEndDate
  Date, format: NA ...
$ ActivityEndTime.Time
  chr "" ...
$ ActivityEndTime.TimeZoneCode
  chr "" ...
$ ActivityDepthHeightMeasure.MeasureValue
  num 1 1.5 ...
$ ActivityDepthHeightMeasure.MeasureUnitCode
  chr "feet" ...
$ ActivityDepthAltitudeReferencePointText
  chr "" ...
$ ActivityTopDepthHeightMeasure.MeasureValue
  chr "" ...
$ ActivityTopDepthHeightMeasure.MeasureUnitCode
  chr "" ...
$ ActivityBottomDepthHeightMeasure.MeasureValue
  chr "" ...
$ ActivityBottomDepthHeightMeasure.MeasureUnitCode
  chr "" ...
$ ProjectIdentifier
  chr "" ...
$ ActivityConductingOrganizationText
  chr "U.S. Geological Survey-Water Resources Discipline" ...
$ MonitoringLocationIdentifier
  chr "USGS-0531656290" ...
$ ActivityCommentText
  chr "L-3040077 Received October 29,2005" ...
$ SampleAquifer
  chr "" ...
$ HydrologicCondition
  chr "Not determined" ...
$ HydrologicEvent
  chr "Routine sample" ...
$ SampleCollectionMethod.MethodIdentifier
  chr "10" ...
$ SampleCollectionMethod.MethodIdentifierContext

```

```

    chr "USGS parameter code 82398" ...
$ SampleCollectionMethod.MethodName
    chr "Equal width increment (ewi)" ...
$ SampleCollectionEquipmentName
    chr "Grab sample" ...
$ ResultDetectionConditionText
    chr "Not Detected" ...
$ CharacteristicName
    chr "Ammonia and ammonium" ...
$ ResultSampleFractionText
    chr "Dissolved" ...
$ ResultMeasureValue
    num NA NA ...
$ ResultMeasure.MeasureUnitCode
    chr "" ...
$ MeasureQualifierCode
    chr "" ...
$ ResultStatusIdentifier
    chr "Accepted" ...
$ StatisticalBaseCode
    chr "" ...
$ ResultValueTypeName
    chr "Actual" ...
$ ResultWeightBasisText
    chr "" ...
$ ResultTimeBasisText
    chr "" ...
$ ResultTemperatureBasisText
    chr "" ...
$ ResultParticleSizeBasisText
    chr "" ...
$ PrecisionValue
    chr "" ...
$ ResultCommentText
    chr "" ...
$ USGSPCode
    chr "00608" ...
$ ResultDepthHeightMeasure.MeasureValue
    chr "" ...
$ ResultDepthHeightMeasure.MeasureUnitCode
    chr "" ...
$ ResultDepthAltitudeReferencePointText
    chr "" ...
$ SubjectTaxonomicName
    chr "" ...
$ SampleTissueAnatomyName
    chr "" ...
$ ResultAnalyticalMethod.MethodIdentifier
    chr "CL037" ...
$ ResultAnalyticalMethod.MethodIdentifierContext
    chr "USGS" ...

```

```

$ ResultAnalyticalMethod.MethodName
  chr "Nutrients, salic/hypo, color" ...
$ MethodDescriptionText
  chr "USGS OF 93-125, p 125" ...
$ LaboratoryName
  chr "" ...
$ AnalysisStartDate
  chr "2005-11-16" ...
$ ResultLaboratoryCommentText
  chr "" ...
$ DetectionQuantitationLimitTypeName
  chr "Laboratory Reporting Level" ...
$ DetectionQuantitationLimitMeasure.MeasureValue
  num 0.04 0.04 ...
$ DetectionQuantitationLimitMeasure.MeasureUnitCode
  chr "mg/l as N" ...
$ PreparationStartDate
  chr "" ...
$ ProviderName
  chr "NWIS" ...
$ timeZoneStart
  num 5 5 ...
$ timeZoneEnd
  num 0 0 ...
$ ActivityStartDateTime
  POSIXct, format: "2005-10-26 19:00:00" ...
$ ActivityEndDateTime
  POSIXct, format: NA ...

```

The `importQW` function can be used to process the data in a dataset such as `WFBC.wqp` and create a dataset with the concentration data as class "qw." Except for `values` and `ColNames`, the argument names in `importQW` correspond to the slot names in the object of class "qw." The argument `values` processes the data in a single column and with `remark.codes` construct the data stored in the slot named ".Data." The argument `ColNames` can be used to set the slot named "names" but that is typically not used. Note that `importQW` cannot be used to import interval-censored data, those data must be constructed using the `as.qw` function.

The user must verify that the data are in the correct format for the conversion. Two critical variables are the numeric values and the remark codes. The column containing the numeric values must contain the values for all data, censored and uncensored. The remark codes must be the valid codes for "qw" data objects described in the Introduction. The data retrieved from the WQP must be modified to meet these standards. Censored values are not stored in the numeric result column and must be merged with the reporting level and the remark codes must be generated from the column named "ResultDetectionConditionText." Those modifications and the data conversion are performed in the code following this paragraph.

```

> # Combine the numeric data and convert the remark codes
> WFBC.wqp <- transform(WFBC.wqp,
+   values=ifelse(ResultDetectionConditionText == "Not Detected",
+     DetectionQuantitationLimitMeasure.MeasureValue, ResultMeasureValue),
+   remark.codes = ifelse(ResultDetectionConditionText == "Not Detected",

```

```

+     "<", "")) # Everything else is passed through as uncensored
> # Convert the data
> WFBC.nh3 <- importQW(WFBC.wqp, keep=c("MonitoringLocationIdentifier",
+   "ActivityStartDate", "ActivityStartTime.Time", "ActivityEndDate",
+   "ActivityEndTime.Time", "ActivityStartTime.TimeZoneCode", "ActivityMediaName"),
+   values="values",
+   remark.codes="remark.codes",
+   value.codes="ResultCommentText",
+   reporting.level="DetectionQuantitationLimitMeasure.MeasureValue",
+   reporting.method="DetectionQuantitationLimitTypeName",
+   reporting.units="DetectionQuantitationLimitMeasure.MeasureUnitCode",
+   analyte.method="ResultAnalyticalMethod.MethodIdentifier",
+   analyte.name="CharacteristicName",
+   unique.code="USGSPCode")
> # And show what we've got
> # print the structure, note that the output is captured and modified to fit
> # in a narrow output
> Tmp <- capture.output(str(WFBC.nh3, vec.len=2))
> Tmp <- sapply(Tmp, sub, pattern=":", replacement="\n  ", fixed=TRUE)
> cat(Tmp, sep="\n")

```

```

'data.frame'
      9 obs. of  8 variables:
 $ MonitoringLocationIdentifier
   chr  "USGS-0531656290" "USGS-0531656290" ...
 $ ActivityStartDate
   Date, format: "2005-10-26" "2005-10-26" ...
 $ ActivityStartTime.Time
   chr  "14:00:00" "15:45:00" ...
 $ ActivityEndDate
   Date, format: NA NA ...
 $ ActivityEndTime.Time
   chr  "" "" ...
 $ ActivityStartTime.TimeZoneCode
   chr  "CDT" "CDT" ...
 $ ActivityMediaName
   chr  "Water" "Water" ...
 $ Ammonia.N
   qw, left censoring: "<0.040" "<0.040" ...

```

3 Arithmetic Operations

Addition and multiplication are accomplished by using the `add` and `multiply` functions rather than the arithmetic operators, "+" and "*" in order to preserve or update the meta data. The `add` function can add or subtract water-quality data of class "qw" or add or subtract numeric values to water-quality data. It is typically used to compute or recompute constituents that are not measured directly. The `multiply` function provides a method for multiplying water-quality data of class "qw" by a numeric value. It is typically used to change the units of the data. Use of the `multiply` function is not demonstrated in this section.

The code following this paragraph recomputes the values for dissolved organic nitrogen (NitrogenOrg), which is computed as dissolved Kjeldahl nitrogen (Kjeldahl.N.00623) minus dissolved ammonia (Ammonia.N). When Kjeldahl.N.00623 is uncensored and Ammonia.N is censored, then the result for NitrogenOrg is censored, see the results for the first line of executable code below. The value that is reported in the data retrieved from NWISweb (<0.7) is correct for descriptive purposes, but for statistical analysis, a more precise value is needed, one that puts the value within a range defined by the value for Kjeldahl.N.00623 (0.7) and the range of possible values for Ammonia.N (from 0 to 0.02). The value for analysis should be interval censored in the range from 0.68 to 0.70. That computation is on the second and following lines of executable code and the results shown below. Those data can now be used to produce statistical results that are unbiased, at least from the context of censoring.

```
> # Print an example of censored dissolved organic nitrogen
> LCobb.nuts[10, c("NitrogenOrg", "Ammonia.N", "Kjeldahl.N.00623")]

      NitrogenOrg Ammonia.N Kjeldahl.N.00623
17          <0.7      <0.02             0.7

> # Recompute censored dissolved organic nitrogen
> LCobb.nuts$NitrogenOrg <- with(LCobb.nuts, add(Kjeldahl.N.00623, -Ammonia.N,
+   analyte="Organic nitrogen", pcode="00607"))
> LCobb.nuts[10, c("NitrogenOrg", "Ammonia.N", "Kjeldahl.N.00623")]

      NitrogenOrg Ammonia.N Kjeldahl.N.00623
17    0.68-0.70      <0.02             0.7
```

The ratio of water-quality values can be computed using the `ratio` function, this is equivalent to division, which would normally be performed using the "/" operator. The `ratio` function is different from the other functions discussed in this section because the output is of class "mcens" rather than maintaining class "qw." The code following this paragraph demonstrates how to use the `ratio` function by computing the ratio of dissolved to whole-water phosphorus.

```
> # Subset the data to create a demonstration data set.
> LCobb.sub <- LCobb.nuts[1:20, c("Phosphorus.P", "Phosphorus_WW.P")]
> # Compute the ratio
> LCobb.sub$Ratio <- with(LCobb.sub, ratio(Phosphorus.P, Phosphorus_WW.P))
> # Print the results
> LCobb.sub

      Phosphorus.P Phosphorus_WW.P      Ratio
163             NA              NA    <NA>
```


1	0.37	0.37	1
18	0.01	0.10	0.1
35	0.03	0.11	0.272727272727273
12	<0.01	0.22	0 - 0.0454545454545455
29	0.35	0.70	0.5
6	0.03	0.15	0.2
23	0.06	0.10	0.6
40	0.29	0.65	0.446153846153846
17	0.12	0.23	0.521739130434783
33	0.03	0.18	0.166666666666667
50	0.10	0.16	0.625
67	0.05	0.06	0.833333333333333
71	0.07	0.11	0.636363636363636
76	0.10	0.11	0.909090909090909
80	0.25	0.35	0.714285714285714
14	0.22	0.27	0.814814814814815
31	0.07	0.19	0.368421052631579
48	0.01	0.12	0.0833333333333333
65	<0.01	0.07	0 - 0.142857142857143

4 Comparison Operations

Comparisons between uncensored values is very straightforward and all of the comparison operators in R "<," "<=," "==," "!=," ">," and ">=" give consistent results. Comparisons between censored values and between censored and uncensored values is not always straightforward. For example, it is clear the expression $4 > <2$ results in `TRUE`. But what about $2 > <4$? In that latter case, the correct result would be `NA` because it is not known whether 2 is greater than the actual value of <4 . All of the comparison operators in R work on data of class "qw" and the analysis classes "lcens" and "mcens." Furthermore, an additional comparison operator "%=%" that could be defined as "is in the range of" is defined for comparing censored values. The code following this paragraph demonstrates the results of the comparison operators.

```
> # Use the Ammonia data from 0531656290
> NH3 <- WFBC.NH3$Ammonia.N
> # Print the values, specifically calling print makes it more readable
> # The "n" following E 0.005 is a non-blank value qualifying code, verifying
> # that the E means the the value is greater than the detection limit, but
> # less than the reporting level
> print(NH3)

[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005    n
[8] 0.026      < 0.01

> # Equality and inequality:
> NH3 == 0.026

[1] NA NA FALSE NA NA FALSE FALSE TRUE FALSE

> NH3 != 0.026

[1] NA NA TRUE NA NA TRUE TRUE FALSE TRUE

> # Greater than and greater than or equal to
> NH3 > .1

[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE

> NH3 >= .1

[1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE

> # Less than and less than or equal to
> NH3 < 0.04

[1] TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE

> NH3 <= 0.04
```

```
[1] TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE
```

```
> # And the range checker
```

```
> NH3 %~=% 0.02
```

```
[1] TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE
```

5 Miscellaneous Manipulations

Occasionally, the same constituent will be represented by more than one column in a data set. This is common when multiple analytic methods are used to quantify a constituent. Often the user will want a single column representing the data for some particular analysis. The example following this paragraph demonstrates the use of the `qwCoalesce` function to create a single column of data from diverse sources. The `qwCoalesce` function acts like the `coalesce` function in the `smwrBase` package by selecting the first non-missing value in each row in the order specified in the arguments. All of the arguments to `qwCoalesce` must be of class "qw."

```
> # Retrieve alkalinity data for a couple of sites.
> sites <- c("01493112", "01632900")
> # The parameter codes represent the preferred order for computing alkalinity
> # according to NWIS
> PC <- c("39086", "29802", "39036", "00418", "39087", "29803", "29801", "00421")
> # Get the data
> Alk <- importNWISqw(sites, PC, begin.date="2011-01-01", end.date="2013-12-31",
+   use.pnames=TRUE)
> # Note only parameter codes 29801 and 39086 were retrieved for these sites
> # Compute alkalinity
> Alk <- transform(Alk, Alk=as.numeric(qwCoalesce(P39086, P29801)))
> # Print the first 10 rows of the data
> head(Alk, 10)
```

	site_no	sample_dt	sample_tm	tzone_cd	medium_cd	sample_end_dt	sample_end_tm	P29801
45	01493112	2011-07-27	12:30	EST	WS			NA
36	01632900	2011-08-02	10:00	EDT	WS			NA
32	01493112	2011-09-08	11:00	EST	WS			25
47	01493112	2011-09-23	11:30	EST	WS			NA
38	01493112	2011-10-12	10:45	EST	WS			NA
39	01493112	2011-11-15	12:45	EST	WS			NA
50	01493112	2011-12-07	10:15	EST	WS			NA
40	01493112	2011-12-12	11:15	EST	WS			NA
3	01632900	2011-12-19	12:00	EST	WS			220
41	01493112	2011-12-20	10:00	EST	WS			NA
	P39086	Alk						
45	33	33.0						
36	210	214.0						
32	NA	25.2						
47	24	24.0						
38	24	24.0						
39	23	23.0						
50	23	23.0						
40	23	23.0						
3	210	210.0						
41	22	22.0						

The `summary` function has a method for class "qw" that conforms to the expected output for the `summary` function method for class "data.frame"—a vector of length 6 that provides a very simply summary of the data. The example immediately following this paragraph demonstrates the output

in the context of the data frame. Also shown is the output from `str`. The `summary` function has a method for class "qw" has an additional argument, `details`, that returns a list with more detailed information. The example also demonstrates this output.

```
> # Print the summary information for WFBC.NH3.
> # Nobs is the number of non-missing values
> summary(WFBC.NH3)
```

site_no	sample_dt	sample_tm	tzone_cd
Length:9	Min. :2005-10-26	Length:9	Length:9
Class :character	1st Qu.:2006-03-28	Class :character	Class :character
Mode :character	Median :2006-05-17	Mode :character	Mode :character
	Mean :2006-04-17		
	3rd Qu.:2006-06-17		
	Max. :2006-08-22		
medium_cd	sample_end_dt	sample_end_tm	Ammonia.N
Length:9	Min. :2006-03-31	Length:9	Length :9
Class :character	1st Qu.:2006-03-31	Class :character	Nobs :9
Mode :character	Median :2006-05-01	Mode :character	Analyte:Ammonia and ammonium
	Mean :2006-05-07		Censor :left
	3rd Qu.:2006-06-19		Units :mg/l as N
	Max. :2006-06-19		Method :many

```
> # Compare to str
> str(WFBC.NH3)
```

```
'data.frame':      9 obs. of  8 variables:
 $ site_no      : chr  "0531656290" "0531656290" "0531656290" "0531656290" ...
 $ sample_dt    : Date, format: "2005-10-26" "2005-10-26" ...
 $ sample_tm    : chr  "14:00" "15:45" "16:00" "10:17" ...
 $ tzone_cd     : chr  "CDT" "CDT" "CST" "CDT" ...
 $ medium_cd    : chr  "WS" "WS" "WS" "WS" ...
 $ sample_end_dt: Date, format: NA NA ...
 $ sample_end_tm: chr  "" "" "16:00" "08:18" ...
 $ Ammonia.N    : qw, left censoring: "<0.040" "<0.040" ...
```

```
> # More details can be extracted from the summary of the qw data
> summary(NH3, details=TRUE)
```

```
$Length
[1] 9
```

```
$Nobs
[1] 9
```

```
$Analyte
[1] "Ammonia and ammonium"
```

```
$Censor
```

```
[1] "left"
```

```
$Units
```

```
[1] "mg/l as N"
```

```
$Method
```

```
[1] "CL037" "00048"
```

```
$RL.method
```

```
[1] "LRL" "IRL"
```

```
> # In this output, the units, and all of the analytical methods and reporting  
> # methods are returned, rather than "many" in the previous call to summary.
```

Data of class "qw" can be subsetted, much like any other vector. The example code following this paragraph demonstrates a few simple cases. In addition to subsetting using `[]`, the `subset` function can be used to extract data based on the meta data. The example code also demonstrates that capability. Because of the meta data, individual values cannot be updated using `[]`, except being set to NA. The example code demonstrates setting a single value to NA, note the additional requirement to treat the value as a matrix rather than as a vector as the extraction does.

```
> # Print NH3.
```

```
> print(NH3)
```

```
[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005      n  
[8] 0.026      < 0.01
```

```
> # select the first 3 values
```

```
> print(NH3[1:3])
```

```
[1] < 0.04      < 0.04      0.1
```

```
> # skip the first value
```

```
> print(NH3[-1])
```

```
[1] < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005      n      0.026  
[8] < 0.01
```

```
> # extract using a logical vector
```

```
> print(NH3[WFBC.NH3$sample_dt < "2006-01-01"])
```

```
[1] < 0.04      < 0.04
```

```
> # Use subset to extract the data associated with an analytical method
```

```
> print(subset(NH3, analyte.method == "CL037"))
```

```
[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04
```

```

> # Make a temporary copy of NH3 to demonstrate assignment
> Tmp <- NH3
> # Must treat as a matrix to set the necessary meta data to NA
> Tmp[2,] <- NA
> print(Tmp)

[1] < 0.04      NA      0.1      < 0.04      < 0.04      0.12      E 0.005    n
[8] 0.026      < 0.01

```

```

> rm(Tmp)

```

6 Conversions for Analysis

The "qw" class is useful for storing water-quality data because it retains meta information that help to understand the data. But the data must be converted to another type for analysis. All of the analytic functions in **smwrQW** convert data of class "qw" to class "lcens" or class "mcens" depending on the censoring. If the data are uncensored or strictly left-censored, then the data are converted to class "lcens" for analysis. Any other censoring requires conversion to class "mcens." The process of conversion uses the numeric data, the remark code information and the reporting level information. A thorough discussion of the conversion is in Lorenz (2016).

Plotting the data can help understand the data and how it will be interpreted for analysis. The example code immediately following this paragraph creates a plot of the ammonia data that has been used in previous examples. An additional argument that can be very useful is `yaxis.log` that can be set to `TRUE` to draw the data on a logarithmic scale. The graph shows uncensored data as solid filled circles and censored data as open circles. The reporting level is shown by the colored horizontal lines—the color of those lines changes with the analytic method, but no key is available for those colors. The y-axis caption is derived from the characteristic name and the units. The x-axis is simply the index number, sequential from 1 to the number of observations.

```
> # Set up the graphics environment, the equivalent call for an on screen
> # device would be setPage("square")
> setSweave("graph01", 6 ,6)
> # Plot the data
> plot(NH3, set.up=FALSE)
> # Required call to close PDF output graphics
> graphics.off()
```

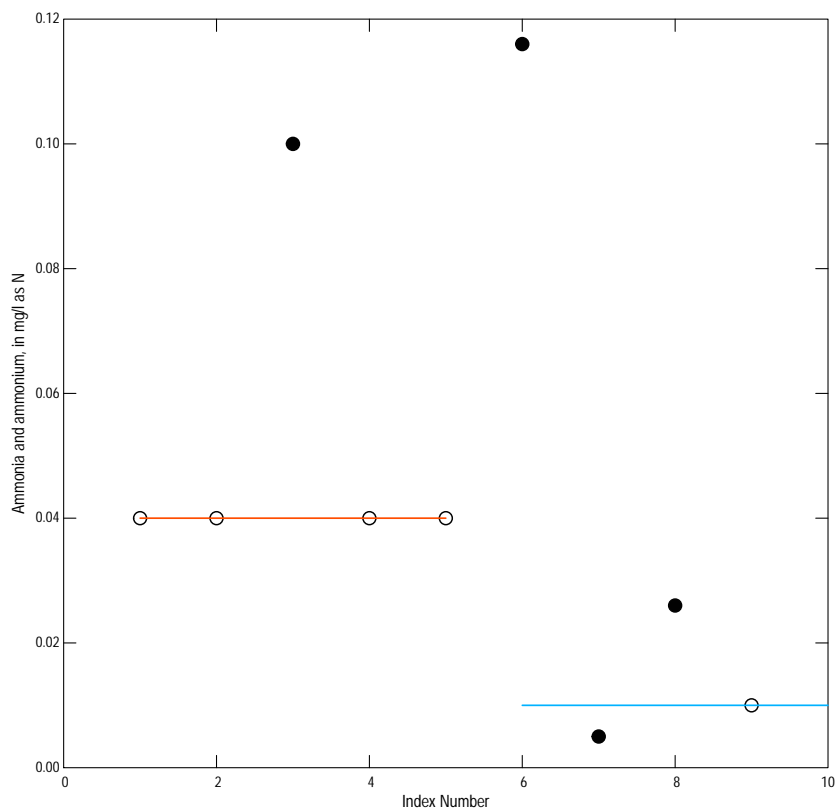



Figure 1. The default graph of water-quality data.

Figure 1 can be used to describe the how left-censored data are created (for either class "lcens" or "mcens"). Bonn (2008) provides an excellent description of the issues related to reporting censored data at the quantitation limit but reporting uncensored data using a detection limit and describes two methods for recensoring the data—censor everything at the quantitation limit or change the quantitation limit to the detection limit. The term reporting limit used in **smwrQW** refers to either the detection limit or the quantitation limit.

The seventh and ninth values in **NH3** are examples of reporting uncensored data at the detection limit and censored data at the quantitation limit, respectively. To provide unbiased statistical analysis, the conversion of the data to either "lcens" or "mcens" censors everything at the reporting level recorded with the data. The code immediately following this paragraph illustrates the manual conversion of the **NH3** data.

```
> # The raw data--the n following the E 0.005 value is a qualification code
> # indicating that the reported value is less than the reporting level
> print(NH3)
```

[1]	< 0.04	< 0.04	0.1	< 0.04	< 0.04	0.12	E 0.005	n
[8]	0.026	< 0.01						

```
> # And converted to lcens
> print(as.lcens(NH3))
```

```
[1] <0.04 <0.04 0.1 <0.04 <0.04 0.116 <0.01 0.026 <0.01
```

```
> # And converted to mcens
> print(as.mcens(NH3))
```

```
[1] <0.04 <0.04 0.1 <0.04 <0.04 0.116 <0.01 0.026 <0.01
```

If the user wants to use the detection limit rather than the quantitation limit, then the user must look up the detection limit for each analytic method, which may vary over time. For the NH3 data, The detection limit for method 00048 for the period of time covered by the last four values in NH3 was 0.005. The code following this paragraph demonstrates how to recode those data. The code uses the `as.data.frame` to convert the data in NH3; the function `convertFqw` can be used to expand any or all columns of class "qw" in a dataset.

```
> # Create a dataset that can be easily manipulated
> NH3.df <- as.data.frame(NH3, expand=TRUE)
> print(NH3.df)
```

	NH3.va1	NH3.va2	NH3.rmk	NH3.vqc	NH3.rlv	NH3.rmt	NH3.unt	NH3.mth	NH3.nam
1	0.000	0.040	<		0.04	LRL mg/l as N	CL037	Ammonia and ammonium	
2	0.000	0.040	<		0.04	LRL mg/l as N	CL037	Ammonia and ammonium	
3	0.100	0.100			0.04	LRL mg/l as N	CL037	Ammonia and ammonium	
4	0.000	0.040	<		0.04	LRL mg/l as N	CL037	Ammonia and ammonium	
5	0.000	0.040	<		0.04	LRL mg/l as N	CL037	Ammonia and ammonium	
6	0.116	0.116			0.01	IRL mg/l as N	00048	Ammonia and ammonium	
7	0.005	0.005	E	n	0.01	IRL mg/l as N	00048	Ammonia and ammonium	
8	0.026	0.026			0.01	IRL mg/l as N	00048	Ammonia and ammonium	
9	0.000	0.010	<		0.01	IRL mg/l as N	00048	Ammonia and ammonium	
NH3.pcd									
1	00608								
2	00608								
3	00608								
4	00608								
5	00608								
6	00608								
7	00608								
8	00608								
9	00608								

```
> # reset the reporting level, column suffix .rlv
> NH3.df$NH3.rlv[NH3.df$NH3.mth == "00048"] <- 0.005
> # and recensor the data in column suffix .va2
> # Must be careful not to recensor elevated censored values
> NH3.df$NH3.va2[NH3.df$NH3.mth == "00048" & NH3.df$NH3.rmk == "<" &
+   NH3.df$NH3.va2 == 0.01] <- 0.005
> # Print to verify
> print(NH3.df)
```

	NH3.va1	NH3.va2	NH3.rmk	NH3.vqc	NH3.rlv	NH3.rmt	NH3.unt	NH3.mth	NH3.nam
1	0.000	0.040	<		0.040	LRL mg/l as N	CL037	Ammonia and ammonium	
2	0.000	0.040	<		0.040	LRL mg/l as N	CL037	Ammonia and ammonium	
3	0.100	0.100			0.040	LRL mg/l as N	CL037	Ammonia and ammonium	
4	0.000	0.040	<		0.040	LRL mg/l as N	CL037	Ammonia and ammonium	
5	0.000	0.040	<		0.040	LRL mg/l as N	CL037	Ammonia and ammonium	
6	0.116	0.116			0.005	IRL mg/l as N	00048	Ammonia and ammonium	
7	0.005	0.005	E	n	0.005	IRL mg/l as N	00048	Ammonia and ammonium	
8	0.026	0.026			0.005	IRL mg/l as N	00048	Ammonia and ammonium	
9	0.000	0.005	<		0.005	IRL mg/l as N	00048	Ammonia and ammonium	

	NH3.pcd
1	00608
2	00608
3	00608
4	00608
5	00608
6	00608
7	00608
8	00608
9	00608

```
> # And convert back to water-quality data (as a data.frame)
> NH3.df <- convert2qw(NH3.df, "qw")
> print(NH3.df)
```

	NH3
1	<0.040
2	<0.040
3	0.100
4	<0.040
5	<0.040
6	0.120
7	EO.005
8	0.026
9	<0.005

```
> # verify the conversion
> print(as.lcens(NH3.df$NH3))
```

```
[1] <0.04 <0.04 0.1 <0.04 <0.04 0.116 0.005 0.026 <0.005
```

Censored methods for some kinds of analyses do not exist. For those cases, any but right-censored data of class "qw" can be converted to numeric values using the `as.numeric` function. That function uses an approach based on simple substitution to estimate values for left- and interval-censored data. The mid range is used for interval censored values. Left-censored values are converted sequentially from the smallest value up to the largest censored value—one half the reporting level is used for the smallest left-censored value, then the mean of all values less than the reporting level is used to substitute for each larger reporting level until the largest value is computed. The code following this paragraph illustrates the use of `as.numeric`. In the conversion to numeric, the data are first processed as for any other analysis described previously in this

section. For the NH3 data, the seventh and ninth values are first censored at 0.01 and converted to numeric values at one-half the reporting level or 0.005. The first, second, fourth and fifth values are censored at 0.04, so their numeric value is the mean of 0.005, 0.026, and 0.005 or 0.012. The output from the last line of code shows that conversion.

```
> # Print the data
> print(NH3)

[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005    n
[8] 0.026      < 0.01

> # And convert to numeric values
> print(as.numeric(NH3))

[1] 0.012 0.012 0.100 0.012 0.012 0.116 0.005 0.026 0.005
```

There is one final conversion function for data of class "qw" called `qw2mcens`. That function converts the data to class "mcens" but treats the less-than values as interval-censored values between 0 and the reporting level. It does not convert quantified data less than the reporting level. It is illustrated in the example code following this paragraph. Interval data are printed as a hyphenated range.

```
> # Print the data
> print(NH3)

[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005    n
[8] 0.026      < 0.01

> # And convert to numeric values
> print(qw2mcens(NH3))

[1] 0 - 0.04 0 - 0.04 0.1      0 - 0.04 0 - 0.04 0.116    0.005    0.026    0 - 0.01
```

7 Mathematical Transformations

A few math functions are supported for transforming water-quality data. Most commonly these will be the power transforms `log`, `log10`, `exp`, and `sqrt`. For the conversion, if the water-quality data are uncensored, then the data are converted to numeric data and the function applied to those values. If the data are left-censored, then the data are converted to class "lcens" and the function applied. For any other censoring, the data are converted to class "mcens" and the function applied. The **smwrQW** package also has a function called `pow` the does a power transform for any arbitrary exponent greater than 0. By default, data of class "qw" are converted using the `qw2mcens` function, but the output type can be controlled by the `out` argument. The data computed by `pow` are scaled by dividing by the exponent. The code following this paragraph provides some examples.

```
> # Print the data
> print(NH3)

[1] < 0.04      < 0.04      0.1      < 0.04      < 0.04      0.12      E 0.005      n
[8] 0.026      < 0.01

> # Compute the natural log, data converted to lcens
> print(log(NH3))

[1] <-3.219 <-3.219 -2.303 <-3.219 <-3.219 -2.154 <-4.605 -3.65 <-4.605

> # Compute the square root, data converted to lcens
> print(sqrt(NH3))

[1] <0.2 <0.2 0.3162 <0.2 <0.2 0.3406 <0.1 0.1612 <0.1

> # The default output from pow, using an exponent to mimic the square root
> print(pow(NH3, .5))

[1] 0 - 0.4 0 - 0.4 0.6325 0 - 0.4 0 - 0.4 0.6812 0.1414 0.3225 0 - 0.2

> # And forced to lcens
> print(pow(NH3, .5, out="lcens"))

[1] <0.4 <0.4 0.6325 <0.4 <0.4 0.6812 <0.2 0.3225 <0.2

> # Note that the last 2 are simply twice the result using sqrt
```

8 Other Applications

This vignette summarizes working with water-quality data of class "qw." It concentrates only on manipulating the data. Other vignettes illustrate the analysis or graphing data. Those vignettes typically use the data classes designed for analysis, "lcens" and "mcens" but the methods can be used directly with data of class "qw." The vignette "Quality Control Data Analysis" also uses data of class "qw" in its presentation of some tools for processing QA/QC data.

References

- [1] Bonn, B.A., 2008, Using the U.S. Geological Survey National Water Quality Laboratory LT-MDL to evaluate and analyze data: U.S. Geological Survey Open-File Report 2008-1227, 73p.
- [2] Lorenz, D.L., 2016, smwrQW—an R package for managing and analyzing water-quality data, version 1.0.0: U.S. Geological Survey Open File Report 2016-XXXX, XX p.