

Quality Control Data Analysis

Dave Lorenz, Laura Medalie, and Jeffrey Martin

January 20, 2016

Abstract

These examples demonstrate some of the functions and statistical methods for the analysis of water-quality control data that are available in the `smwrQW` package. The examples illustrate many of the concepts discussed by Mueller and others (2015).

Contents

1	Introduction	2
2	Statistical Concepts	3
2.1	Confidence Interval of the Mean	3
2.2	Confidence Interval for the Median and other Percentiles	5
2.3	Confidence Interval for a Proportion	6
2.4	Relative Percent Difference	7
3	Initial Data Processing	8
4	Analysis of Blanks	11
5	Analysis of Spikes	13
6	Analysis of Replicates	14

1 Introduction

Most of the examples will use datasets supplied in the **smwrQW** package, but the Statistical Concepts section will use small, artificial datasets, with known properties to illustrate the use of the functions. The artificial data are class "lcens" rather than class "qw," but the statistical methods apply to both classes. The user should read the Working with Water-Quality Data vignette to become familiar with the management of water-quality data in the **smwrQW** package before attempting the examples in this vignette.

```
> # Load the smwrQW package
> library(smwrQW)
> # Generate normal data
> set.seed(132)
> Xn <- rnorm(26, mean=5)
> # And left-censor it at 2 levels
> Xc <- as.lcens(Xn, rep(c(4.5, 5.0), 13))
> # And log-normal data
> Xln <- rlnorm(26, mean=0.1, sd=0.3)
> # And left censor it at 2 levels
> Xlc <- as.lcens(Xln, rep(c(0.8, 1.2), 13))
> # Multiply censor the data
> Xmc <- censor(Xln, 0.8, 1.75)
```

2 Statistical Concepts

It is not possible, physically or financially, to measure all occurrences of every characteristic of interest in environmental studies. For some characteristics, any direct measurement is impossible. Thus, statistical methods are necessary to make estimates of these characteristics. Such estimates can be less than satisfying, and even the subject of disbelief or derision.

Water-quality data are complicated because the data are reported as censored values if the measured result is less than the reporting level, defined by the laboratory for each method and analyte. Censored values can be problematic for statistical analyses, particularly if some assumed value, such as zero or one-half the reporting level, is substituted for the censored result (see Helsel, 2012, for a detailed discussion). The methods in the **smwrQW** and described in this vignette provide statistically unbiased, or asymptotically unbiased, results with very little user adjustments and do not rely on simple substitution.

2.1 Confidence Interval of the Mean

The uncertainty of an inferential statistic often is indicated by reporting a range of values, referred to as a "confidence interval." Confidence intervals are constructed to contain an unknown characteristic of the population, such as the mean, median, standard deviation, or a percentile, with a specified probability. The width of the confidence interval is the uncertainty due to estimation of a population characteristic based on sample data. Note that the estimation of a confidence interval is appropriate for random samples from a single population.

The equations for computing confidence intervals are not reproduced in this vignette. Mueller and others (2015) and Helsel (2012) present and describe the equations for the user who desires more information about the actual computation.

The `censMean.CI` function can be used to compute the confidence interval for the sample mean. There are six computational methods for the `censMean.CI` function, each designed for different distributions and censoring. The "AMLE" method can be used for **uncensored data** or **left-censored data** that can be assumed to be normally distributed. The "MLE" method can be used for **multiply-censored data** that can be assumed to be normally distributed. The "ROS" method can be used for left-, multiply-, or uncensored data that are approximately normally distributed. The "ROS" method uses robust estimation and bootstrapping to relax the assumption of normality. The "log AMLE" method can be used for left-, or uncensored data that can be assumed to be log-normally distributed. The "log MLE" method can be used for multiply-censored data that can be assumed to be log-normally distributed. The "log ROS" method can be used for left-, multiply-, or uncensored data that are approximately log-normally distributed. The "log ROS" method uses robust estimation and bootstrapping to relax the assumption of normality.

The confidence interval for the mean of normally distributed, uncensored data can also be computed using the `t.test` function in base R as demonstrated in the code immediately following this paragraph. The code also illustrates the use of the `censMean.CI` in the **smwrQW** package. The method "AMLE" is used to compute the confidence interval. The adjusted maximum likelihood method, "AMLE," is first-order unbiased and replicates the results from `t.test`.

```
> # The t.test for confidence interval of the mean
> t.test(Xn, conf.level=.9)
```

One Sample t-test

```

data:  Xn
t = 27.638, df = 25, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 4.676905 5.293098
sample estimates:
mean of x
 4.985001

```

```

> # The adjusted maximum likelihood
> censMean.CI(Xn, method="AMLE", CI=.9)

```

```

      estimate      lcl      ucl  ci
mean 4.985001 4.676905 5.293098 0.9

```

```

> # Compare to the left-censored data
> censMean.CI(Xc, method="AMLE", CI=.9)

```

```

      estimate      lcl      ucl  ci
mean 4.886856 4.525555 5.248157 0.9

```

Most often, water-quality data can be assumed to be from a log-normal distribution. The `method` argument must be "log AMLE" for an unbiased estimate of uncensored or left0censored data, "log MLE" for an asymptotically unbiased estimate of any **censored data**, or "log ROS" for a robust estimate for any censored data. The confidence intervals of the mean of a log-normally distributed sample are not simply back-transformed confidence intervals of the mean of the logs, see Helsel (2012) for details. The code following this paragraph demonstrates the "log AMLE" method for both the uncensored and leftcensored log-normal data and uses "log MLE" and "log ROS" for the mutliply censored data.

```

> # The AMLE, uncensored
> censMean.CI(Xln, method="log AMLE", CI=.9)

```

```

      estimate      lcl      ucl  ci
mean  1.22684 1.121109 1.345951 0.9

```

```

> # The AMLE, left censored
> censMean.CI(Xlc, method="log AMLE", CI=.9)

```

```

      estimate      lcl      ucl  ci
mean 1.245633 1.146584 1.357304 0.9

```

```

> # MLE for multiply censored
> censMean.CI(Xmc, method="log MLE", CI=.9)

```

```

      estimate      lcl      ucl  ci
mean 1.225595 1.122493 1.338166 0.9

```

```
> # and ROS
> censMean.CI(Xmc, method="log ROS", CI=.9)
```

```
      estimate      lcl      ucl  ci
mean  1.22873 1.121265 1.352765 0.9
```

2.2 Confidence Interval for the Median and other Percentiles

The median and other percentiles are statistics of particular importance in QC analyses. Nonparametric confidence intervals on percentiles are calculated using the binomial probability function when applied to uncensored data (Mueller and others, 2015) and an approximation using the Kaplan-Meier method described by Helsel (2012) when the data are censored. In general, the confidence interval for the approximation is slightly smaller than that based on the binomial distribution, as demonstrated by the first part of the code following this paragraph. The `qtiles.CI` function in the **smwrStats** package computes confidence intervals for uncensored data and the `censQtiles.CI` function in the **smwrQW** package computes confidence intervals for left-censored data. The code following this paragraph demonstrates both functions.

```
> # Uncensored data, by default, 90% CI for median
> qtiles.CI(Xln)
```

```
      estimate      lcl      ucl      ci
50% 1.200256 1.054237 1.335974 0.9244813
```

```
> censQtiles.CI(Xln)
```

```
      estimate      lcl      ucl  ci
50% 1.200256 1.054609 1.289765 0.9
attr("minimum")
[1] 0.6258476
attr("maximum")
[1] 2.015399
```

```
> # Compare to censored data
> censQtiles.CI(Xlc)
```

```
      estimate      lcl      ucl  ci
50% 1.200256 1.054609 1.289765 0.9
attr("minimum")
[1] 0.8
attr("maximum")
[1] 2.015399
```

```
> # Compute the upper confidence interval for selected probabilities
> censQtiles.CI(Xlc, probs=c(.75, .9, .95), bound="upper")
```

```
      estimate lcl      ucl  ci
75% 1.497274  0 1.561259 0.9
```

```

90% 1.609559    0 1.681648 0.9
95% 1.681648    0 2.015399 0.9
attr("minimum")
[1] 0.8
attr("maximum")
[1] 2.015399

> # Compare to "filled in" values
> Xlc.f <- fillIn(Xlc, method="log ROS")
> qtiles.CI(Xlc.f, probs=c(.95), bound="upper")

      estimate  lcl ucl ci
95% 1.681648 -Inf  NA  NA

```

The maximum reported for the upper confidence level by the approximation method is the largest value in the data, which is included as an attribute of the returned value. There can be times when that is acceptable. However, if the user must know that the actual upper limit could be greater than the largest value, then the user can estimate all values using the `fillIn` function (assuming either a normal or log-normal distribution) and use the `qtiles.CI` function to determine if the upper level matches the largest value or exceeds it (returned as NA as in the example).

2.3 Confidence Interval for a Proportion

Proportions can be computed for a dataset by dividing the observations into groups, such as those less than or greater than a specified value. In water-quality analyses, proportions often are used to indicate the frequency of analyte detection or exceeding a specified threshold within the total number of observations in a sample dataset. The code following this paragraph demonstrates a simple application of determining the confidence interval of a proportion—for the largest detection limit. The code uses the `binom.test` rather than `prop.test` because it computes the exact test statistic. Note that this example assumes no missing values; if necessary they should be removed before the analysis.

```

> # What is the maximum detection limit?
> Xlc.max <- max(censorLevels(Xlc))
> Xlc.max

[1] 1.2

> # What percentage?
> percentile(Xlc, 1.2)

Percent >= 1.2
          50

> # Compute the confidence interval
> binom.test(sum(Xlc >= Xlc.max),
+           length(Xlc), percentile(Xlc, 1.2, percent=FALSE))

```

Exact binomial test

```
data:  sum(Xlc >= Xlc.max) and length(Xlc)
number of successes = 13, number of trials = 26, p-value = 1
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2992722 0.7007278
sample estimates:
probability of success
                0.5
```

2.4 Relative Percent Difference

The relative percent difference (RPD) between uncensored replicate-pair results is calculated using the following equation:

$$RPD = 100 * \frac{\text{larger value} - \text{smaller value}}{(\text{larger value} + \text{smaller value})/2}$$

The values for RPD range from 0 if the smaller value is equal to the larger value to 200 if the smaller value is 0.

For censored water-quality data, the results can be ambiguous—"less than" values represent an unquantified value between 0, the minimum possible concentration, and the reporting level. Thus the equation for RPD can give any value between 0 and 200, depending on what value is used for the "less than" value. The `qwRPD` function has three options for dealing with those censored values: return `NA`, the minimum possible difference, or the range of possible RPDs for any computation involving censored values. The first two are simple preview options and suitable for very small sample-replicate sizes. The last option returns the values as class "mcens" so that the mean and confidence interval for the RPD can be computed and is suitable for larger sample-replicate sizes. The documentation for `qwRPD` describes the details and gives an example.

3 Initial Data Processing

Jeff and Laura, I think some details on the preprocessing of the sample data. Thanks.

Bonn (2008) describes the fundamentals of the various reporting limits used by the U.S. Geological Survey National Water Quality Laboratory (NWQL). She describes the process of recensoring data at the long-term method detection limit (LT-MDL) for the purposes of analyzing low-level concentrations, which is important for quality control. This section describes that procedure for selected atrazine samples collected by the U.S. Geological Survey National Stream Water-Quality Assessment (NAWQA) program from October 1, 2001 through September 30, 2012. The procedure described would be applicable to any data retrieved from analysis by the NWQL and broadly applicable to data from other laboratories. Ryberg and others (2010) describe a procedure for lowering the effective detection limit for information-rich analyses where values less than the detection limit for the math can be quantified. That procedure could be extended to blanks or other cases of distinct patterns of those quantified values less than the detection level.

The first step in processing these data for the quality-control analysis is to recode the reporting level of the data to the detection limit. This is only necessary if the laboratory uses the quantitation level rather than the detection limit for censored values as described in Bonn (2008). The R code immediately following this paragraph illustrates how to extract the information and update both the reporting levels and the values in the dataset. The initial part of the R code recodes some data in each dataset so that the codes used in the different datasets agree. The user must be careful to review the processing carefully to verify that the detection levels are correct.

```
> # Get the sample and the reporting level data (supplied with smwrQW)
> data(atrazine.df)
> data(atrazine.dl)
> # Change the schedule code (sched) in the sample data to match the
> # LabCode in the reporting level data (easy for these data, others
> # could be more complicated).
> atrazine.df$LabCode <- sub("NWQL", "1", atrazine.df$sched)
> # The reporting level data do not report values for LabCode 12033 prior
> # to 2005-03-04, but that LabCode is recorded in the sample data.
> # LabCode 12033 is equivalent to 12003, so recode
> atrazine.df$LabCode[atriazine.df$LabCode == "12033" &
+   atrazine.df$Date < "2005-03-04"] <- "12003"
> # Extract the method code from the TestID in the reporting level data
> atrazine.dl$QW_METHOD_CD <- substring(atrazine.dl$TestID, 6)
> # For these data the reporting level codes agree, but in some case, the
> # reporting level data uses lower case.
> # The data are now ready for processing. They cannot be merged because
> # there is no exact match for dates. This requires a relatively slow
> # loop to get the 1 to 1 match.
> # First set up vector of detection limits
> DL <- numeric(nrow(atrazine.df))
> # Then loop through each row
> for(i in seq(DL)) {
+   # match method code
+   tmp.mc <- atrazine.df$QW_METHOD_CD[i] == atrazine.dl$QW_METHOD_CD
+   # match reporting level codes
+   tmp.rlc <- atrazine.df$RPT_LEV_CD[i] == atrazine.dl$ReportLevelCode
+   # and finally the dates
```



```

+     tmp.dt <- atrazine.df$Date[i] >= atrazine.dl$StartDate &
+     atrazine.df$Date[i] <= atrazine.dl$EndDate
+     # match lab code only if necessary--more than one detection limit
+     # This works when the LabCode is not recorded in the sample data or
+     # for different DLs among the LabCodes
+     tmp.sel <- tmp.mc & tmp.rlc & tmp.dt
+     if(sum(tmp.sel) > 1) {
+         # extract the value(s)
+         tmp.val <- unique(atrazine.dl[tmp.sel, "DetectionLevel"])
+         if(length(tmp.val) > 1L) {
+             tmp.sel <- tmp.sel & atrazine.df$LabCode[i] == atrazine.dl$LabCode
+             tmp.val <- atrazine.dl[tmp.sel, "DetectionLevel"]
+         }
+     } else { # found a single or none
+         tmp.val <- atrazine.dl[tmp.sel, "DetectionLevel"]
+     }
+     if(length(tmp.val)) {
+         DL[i] <- tmp.val
+     } else {
+         DL[i] <- NA
+     }
+ }
+ # clean up
+ rm(tmp.dt, tmp.mc, tmp.rlc, tmp.sel, tmp.val, i)
+ # Do the missing values make sense?
+ Miss <- which(is.na(DL))
+ atrazine.df[Miss, c("Date", "sched", "RPT_LEV_VA", "RPT_LEV_CD", "QW_METHOD_CD")]

      Date      sched RPT_LEV_VA RPT_LEV_CD QW_METHOD_CD
4964  2002-05-28 NWQL2001      0.05             GCM35
4966  2002-07-23 NWQL2001      0.05             GCM35
11838 2002-07-17 NWQL2001      0.05             GCM35
12563 2002-07-17 NWQL2001      0.05             GCM35
12565 2002-06-05 NWQL2001      0.05             GCM35

> # The values have no code for the reporting level code and the reporting
> # levels are different from those in the reporting level data--OK.
> #
> # Update the reporting level value and code where DL is not missing.
> atrazine.df$DL <- ifelse(is.na(DL), atrazine.df$RPT_LEV_VA, DL)
> atrazine.df$DL_CD <- ifelse(is.na(DL), atrazine.df$RPT_LEV_CD, "DL")
> # Update the censored values in RESULT_VA--this works only for left-censored
> atrazine.df$RESULT_VA <- ifelse(atrazine.df$REMARK_CD == "<",
+   atrazine.df$DL, atrazine.df$RESULT_VA)

```

The last step in processing these data for the quality-control analysis is to split the sample data into three separate datasets, one for each type of data and convert the atrazine data to class "qw" to facilitate processing. The R code following this paragraph splits the data, retaining only the necessary information and uses the function `convert2qw` to construct the dataset containing the atrazine data. The dataset are ready for quality-control analysis of blanks, and replicates

```

> # Subset the atrazine sample data and rename the columns
> # Blank data
> atrazine.oaq <- subset(atrazine.df, MEDIUM_CD == "OAQ",
+   select = c("staid", "Date", "MEDIUM_CD", "SAMP_TYPE_CD",
+     "RESULT_VA", "REMARK_CD", "DL", "DL_CD", "QW_METHOD_CD",
+     "Units", "PARAMETER_CD"))
> names(atrazine.oaq)[5:11] <- paste0("atrazine", c("", ".rmk",
+   ".rlv", ".rmt", ".mth", ".unt", ".pcd"))
> # Check if practcal to reset the DL for the blank data by method
> with(subset(atrazine.oaq, atrazine < atrazine.rlv),
+   table(atrazine, atrazine.mth))

      atrazine.mth
atrazine GCM35
      0.0035      1

> # there is only 1 value less than the detection limit, so the payback for
> # resetting the detection limits is small.
> # Convert the atrazine to "qw"
> atrazine.oaq <- convert2qw(atrazine.oaq, scheme="partial")
> #
> # Replicate data
> atrazine.wsq <- subset(atrazine.df, MEDIUM_CD == "WSQ",
+   select = c("staid", "Date", "MEDIUM_CD", "SAMP_TYPE_CD",
+     "RESULT_VA", "REMARK_CD", "DL", "DL_CD", "QW_METHOD_CD",
+     "Units", "PARAMETER_CD"))
> names(atrazine.wsq)[5:11] <- paste0("atrazine", c("", ".rmk",
+   ".rlv", ".rmt", ".mth", ".unt", ".pcd"))
> # Convert the atrazine to "qw"
> atrazine.wsq <- convert2qw(atrazine.wsq, scheme="partial")
> #
> # Envirionmental sample data
> atrazine.ws <- subset(atrazine.df, MEDIUM_CD == "WS",
+   select = c("staid", "Date", "MEDIUM_CD", "SAMP_TYPE_CD",
+     "RESULT_VA", "REMARK_CD", "DL", "DL_CD", "QW_METHOD_CD",
+     "Units", "PARAMETER_CD"))
> names(atrazine.ws)[5:11] <- paste0("atrazine", c("", ".rmk",
+   ".rlv", ".rmt", ".mth", ".unt", ".pcd"))
> # Convert the atrazine to "qw"
> atrazine.ws <- convert2qw(atrazine.ws, scheme="partial")

```

4 Analysis of Blanks

Blanks are used to estimate the positive bias that can be caused by extraneous contamination introduced into environmental samples during collection, processing, shipment, and laboratory analysis. Evaluation of data from field blanks depends on the inference space represented by the blanks. In general, there are two possibilities: (1) a single blank is prepared to represent potential sources of contamination that affect a specific, small set of environmental samples, or (2) multiple blanks are prepared periodically over time and space to represent potential sources of contamination that might affect a much larger set of environmental samples.

Out of a total of 654 values for atrazine concentrations in field blanks associated with surface-water sampling across the United States, 646 are censored at a consistent DL of 0.004 $\mu\text{g/L}$. In the processing of the data in the "Initial Data Processing" section, only one quantified value was recorded less than the DL. The 8 quantified concentrations that exceed the DL are shown in the code following this paragraph. The 8 quantified concentrations represent about 1.2 percent of the total number of blank samples.

```
> # Print the blank data exceeding the DL
> subset(atrazine.oaq, atrazine >= 0.004)
```

	staid	Date	MEDIUM_CD	SAMP_TYPE_CD	atrazine
187	03303280	2002-05-20	OAQ	2	0.010
212	03378500	2010-06-15	OAQ	2	0.009
235	03609750	2008-04-17	OAQ	2	EO.004
311	05532500	2003-06-19	OAQ	2	0.010
316	05572000	2004-05-20	OAQ	2	EO.005
321	05572000	2012-07-05	OAQ	2	0.006
381	06934500	2002-08-13	OAQ	2	0.250
388	07022000	2008-06-11	OAQ	2	EO.005

This example presents a technique for evaluating potential contamination by comparing the distribution of atrazine in environmental samples to the distribution of the 90-percent upper confidence limit for percentiles of concentration in associated field blanks (Mueller and others (2015)). The code following this paragraph demonstrates one way to create a graph that shows the largest values of the environmental samples compared to the field blanks. The range of the graph covers the largest 40 percent of the data. Different samples may require a different range.

```
> # Initial processing
> # The range of percentages for the environmental data
> Pct <- seq(60, 100, by=0.5)
> # The quantiles of the environmental data
> Env.Q <- quantile(atrazine.ws$atrazine, probs=Pct/100)
> # The range for the blanks, no reason to compute for probabilities
> # More than two times 1.2 percent smaller than 1
> # The confidence interval cannot be computed for 1, (100 percent)
> # set the largest percent to 99.9
> Pct.90 <- seq(97.6, 99.9, by=.1)
> # The upper 90 confidence limit of the blanks
> Bnk.90 <- censQtiles.CI(atrazine.oaq$atrazine, probs=Pct.90/100,
+   CI=0.9, bound="upper")
```

```

> # set up the graphics environment: setSweave used in vignettes
> setSweave("graph01", 6, 6)
> AA.pl <- xyPlot(Pct, Env.Q, Plot=list(name="Environmental Samples",
+   what="lines", color="orange", width="color"), yaxis.log=TRUE,
+   yaxis.range=c(0.001, 100), ytitle="Concentration, in micrograms per liter",
+   xaxis.range=c(60, 100), xtitle="Cumulative Percent of Distribtution")
> # Add minor ticks for detail on the x-axis
> addMinorTicks("x", AA.pl)
> # Add the detection limit and annotation
> refLine(horizontal=0.004, current=AA.pl)
> addAnnotation(65, 0.004, "Maximum Long-term Method Detection Limit", current=AA.pl)
> # Add the UCL--Bnk.90 is a matrix, extract the ucl column
> AA.pl <- addXY(Pct.90, Bnk.90[, "ucl"], Plot=list(name="90 Percent UCL for Blanks",
+   what="lines", color="red", width="color"), current=AA.pl)
> # And the explanation
> addExplanation(AA.pl, "ul")

```

graphics.off()

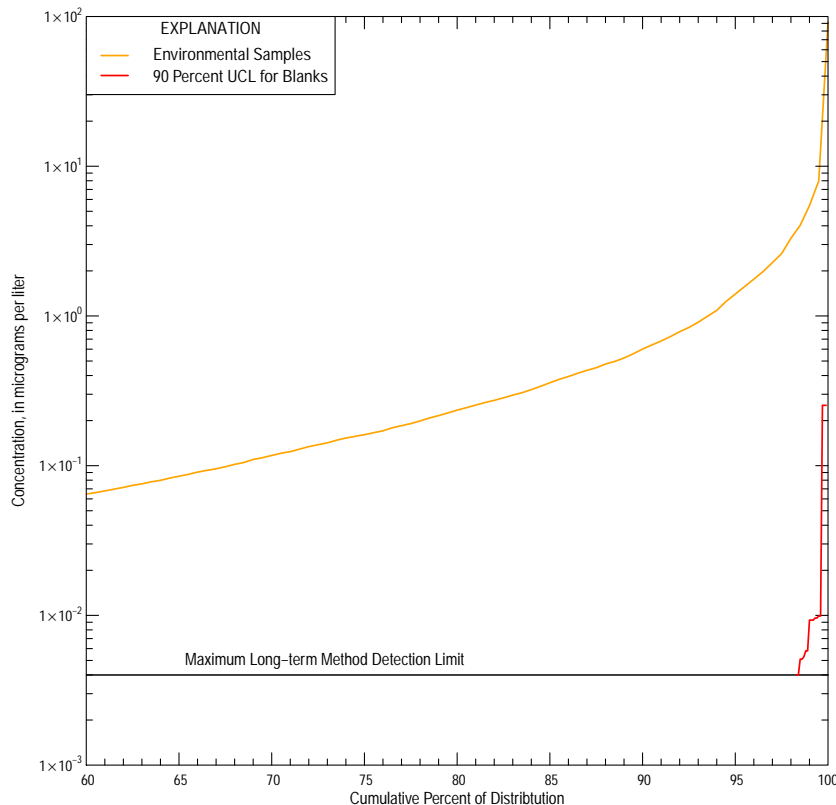


Figure 1. Comparison of environmental and blank data.

5 Analysis of Spikes

Spikes are used to estimate the positive or negative bias that can affect the measured results for environmental samples because of analyte degradation or problems with the analytical methods. This bias is estimated by determining the recovery of known concentrations of the analytes in the spiked sample. Calculation of recovery for matrix spikes requires a separate environmental sample to determine the background concentration of the analyte in the unspiked matrix. Recovery in the spiked matrix samples can be compared to some criteria or to typical recovery for the analytical method based on laboratory reagent spikes.

6 Analysis of Replicates

Replicates are used to measure variability, which is defined as the random error in independent measurements as the result of repeated application of the measurement process under identical conditions. Statistical evaluation of replicate variability is based on the standard deviation of measured values in the primary environmental sample and the replicate sample or samples. If only one set of a large number of replicates was collected, the standard deviation could be calculated directly; however, the general practice is to collect many sets of a small number of replicates under different conditions.

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] Helsel, D.R. 2012, Statistics for Censored Environmental Data Using Minitab and R: New York, Wiley, 324 p.
- [3] 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.
- [4] Mueller, D.K., Schertz, T.L., Martin, J.D., and Sandstrom, M.W., 2015, Design, analysis, and interpretation of field quality-control data for water-sampling projects: U.S. Geological Survey Techniques and Methods book 4, chap. C4, 54 p.
- [5] Ryberg, K.R., Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2010, Trends in pesticide concentrations in urban streams in the United States, 1992–2008: U.S. Geological Survey Scientific Investigations Report 2010–5139, 101 p. plus appendixes. Available at <http://pubs.usgs.gov/sir/2010/5139/>.

Glossary

censored data Any data containing or potentially containing censored values.

left-censored data Any data containing left-censored values.

multiply-censored data Any data containing right- or interval-censored values or a mixture of any types of censored values.

uncensored data Any data not containing any censored values.