Vignette for seawaveQ—An R Package Providing a Model and Utilities for Analyzing Trends in Chemical Concentrations in Streams with a Seasonal Wave (seawave) and Adjustment for Streamflow (Q) and Other Ancillary Variables

Karen R. Ryberg and Aldo V. Vecchia

December 30, 2013

Contents

1 Introduction

This R package, **seawaveQ**, is designed for fitting a parametric regression model for assessing variability and trends in pesticide concentration in streams and was developed by Vecchia and others (2008), and subsequently refined and referred to as the "seawave-Q" model in several trend analyses (Ryberg and others, 2010; Sullivan and others, 2009; Vecchia and others, 2009). In these publications, "seawave-Q" stands for seasonal wave (seawave) with adjustment for streamflow (Q). The model was developed to "handle a number of difficulties often found in pesticide data, such as strong seasonality in response to use patterns, high numbers of concentrations below laboratory reporting levels (RLs), complex relations between streamflow and concentration, and intermittent or changing sampling frequencies (both inter-annually and intra-annually)" (Vecchia and others, 2008). This R package provides a standardized methodology for fitting the seawaveQ model and makes the trend analysis method widely available for use by others. In addition, several enhancements to the seawaveQ model have been included as well as utility functions for working with chemical concentration data. The enhancements and utilities include procedures for preparing and summarizing input data; flexibility to include other explanatory variables besides streamflow; graphical methods for assessing model fit; and plotting routines that may be used for pesticide and other chemical concentration data. A flow chart showing how the various function in the package work together is shown in figure 1 of the U.S. Geological Survey Open-File Report documenting this package (Ryberg and Vecchia, 2013).

The statistical methodology for the seawaveQ model is described in Vecchia and others (2008) and in the U.S. Geological Survey Open-File Report documenting this package (Ryberg and Vecchia, 2013). Users new to this model should read both of those documents before applying the model to their own data. An important part of the model and the output shown below is the seasonal wave. The seasonal wave is a periodic (period of 1 year) solution to a differential equation (Vecchia and others, 2008) that has a pulse input function, a seasonal shift that determines the time at which the seasonal wave reaches its maximum, and a model half-life (see appendix 3. Visualizations of the Seasonal Wave; Ryberg and Vecchia, 2013).

2 Input Data

The seawaveQ model needs two types of input data. The first is the the water-quality sample data including dates, the concentration data, and qualification codes, indicating which values are censored (less than a laboratory reporting level). The second type of data is the continuous ancillary data used in the model, such as streamflow anomalies (Ryberg and Vecchia, 2012). These ancillary data also are used to produce a continuous estimate of pesticide concentration. Examples of the necessary format of these two datasets are provided and documented in the package. The following code shows how to access the example data.

```
> options(width=65)
> # load waterData package, assuming it has already been installed on the system
> library(seawaveQ)
> # load example data that comes with the package
> data(swData)
> # show first few rows of water-quality data for Missouri River at Omaha, Nebr.
> head(qwMoRivOmaha)
```

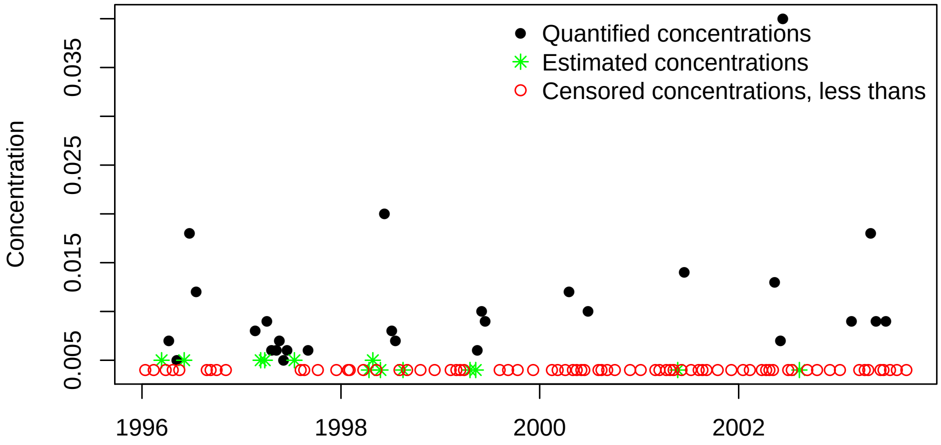
```
     staid      dates times R04035 P04035 R04037 P04037 R04041
1 06610000 1996-01-13  1130      <  0.004      _  0.024      <
2 06610000 1996-02-13  1200      <  0.004      E  0.005      <
3 06610000 1996-03-13  1000      E  0.005      E  0.004      <
4 06610000 1996-03-28  1030      <  0.004      E  0.005      _
5 06610000 1996-04-09  1100      _  0.007      E  0.006      _
6 06610000 1996-04-23  1000      <  0.004      <  0.004      _
  P04041 R39415 P39415 R46342 P46342 R82630 P82630 R82661 P82661
1  0.008      _  0.006      <  0.003      _  0.029      <  0.003
2  0.008      _  0.200      <  0.003      <  0.007      <  0.003
3  0.008      _  0.026      <  0.003      <  0.007      <  0.003
4  0.009      _  0.026      <  0.003      <  0.007      <  0.003
```

```
5  0.014       _  0.075      E  0.003      <  0.007      <  0.003
6  0.012       _  0.040      <  0.003      <  0.007      <  0.003
  R82668 P82668
1       _  0.008
2       _  0.007
3       E  0.004
4       _  0.008
5       _  0.009
6       <  0.002
```

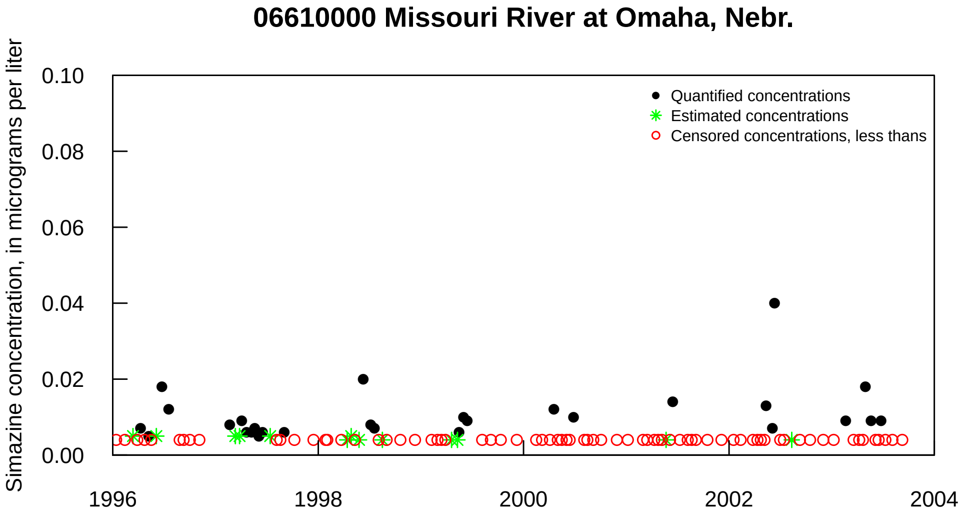
```
> # get a description of the data including definitions of the columns
> # by viewing the help documentation
> ?qwMoRivOmaha
```

Optionally, functions have been provided to plot concentration data. These functions produce scatter plots and box plots that indicate or take into account the censored, less than, values. The functions are *cenScatPlot* and *rosBoxPlot* and examples of their use follow. The box plots are generated using the function *ros*, regression on order statistics, in the R package **NADA** (Lee, 2012). It is an implementation of a regression on order statistics designed for multiply-censored analytical-chemistry data (Helsel, 2005).

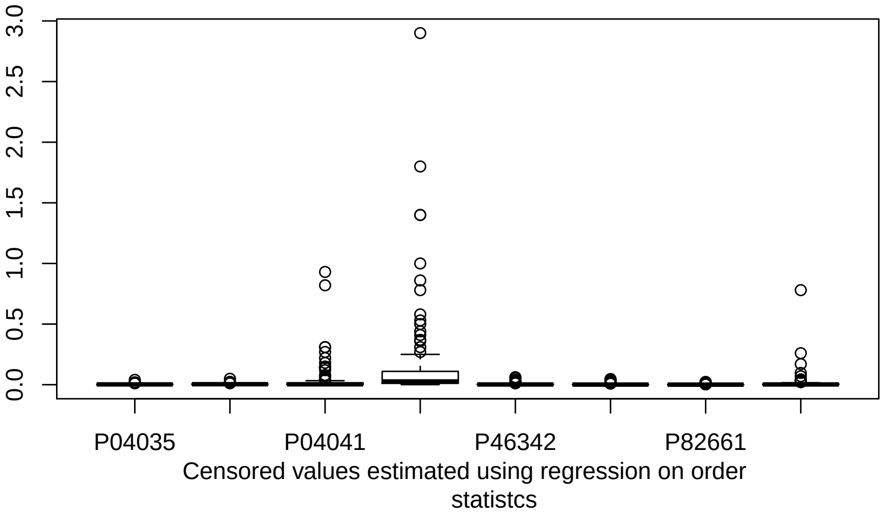
```
> # scatter plot showing quantified, estimated, and censored  values
> cenScatPlot(qwMoRivOmaha, pname="04035")
```



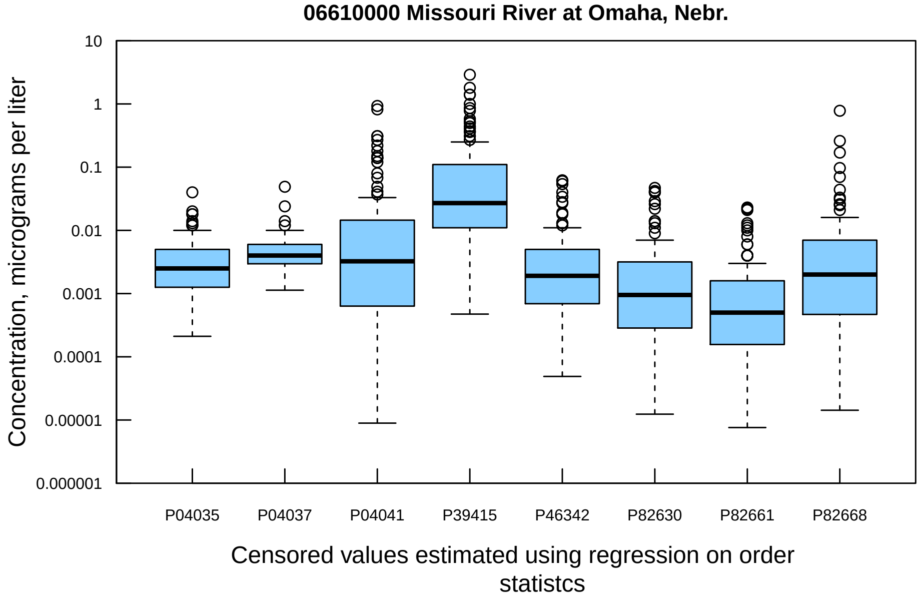
```
> # scatter plot with many additional plotting arguments
> # these options provide a plot closer to the plotting standards
> # of the U.S. Geological Survey, however, these plots may not
> # meet all U.S. Geological Survey publication requirements
> par(las=1, tcl=0.5)
> cenScatPlot(qwMoRivOmaha, pname="04035",
+                        site="06610000 Missouri River at Omaha, Nebr.",
+                        ylabel="Simazine concentration, in micrograms per liter",
+                        legcex=0.7, qwcols=c("R", "P"), ylim=c(0,0.1), yaxs="i",
+                        cex.lab=0.9, cex.axis=0.9, xlim=c(as.Date("1996-01-01"),
+                        as.Date("2004-01-01")), xaxs="i", xaxt="n")
> axdates <- c("1996-01-01", "1998-01-01", "2000-01-01",
+                        "2002-01-01", "2004-01-01")
> axis(1, as.Date(axdates),
+                        labels=c("1996", "1998", "2000", "2002", "2004"), cex.axis=0.9)
```



```
> # simple box plots of water-quality concentrations
> rosBoxPlot(qwMoRivOmaha, qwcols=c("R", "P"))
```



```
> # same boxplot function with many additional plotting arguments
> rosBoxPlot(qwMoRivOmaha, site="06610000 Missouri River at Omaha, Nebr.",
+                        log="y", yaxt="n", ylim=c(0.000001, 10), qwcols=c("R", "P"),
+                        ylab=c("Concentration, micrograms per liter"), col="skyblue1",
+                        cex.axis=0.7, cex.sub=0.8,
+                        par(tcl=0.5, las=1, yaxs="i", mgp=c(3,0.5,0), mar=c(5,5,2,2),
+                        cex.main=0.9))
> axis(2, at=c(0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10),
+ labels=c("0.000001", "0.00001", "0.0001", "0.001", "0.01",
+ "0.1", "1", "10"), cex.axis=0.7)
```



The second data set needed is the one containing the continuous ancillary data for building the model that describes pesticide concentrations.

```
> data(swData)
> # show last few rows of water-quality data for Missouri River at Omaha, Nebr.
> tail(cqwMoRivOmaha)
```

```
        staid      dates dflow     flowa30      flowa1 dsed
2917 06610000 2003-09-25 29200 -0.09433312 -0.00527612  176
2918 06610000 2003-09-26 28800 -0.09268448 -0.01291513  177
2919 06610000 2003-09-27 28700 -0.09102975 -0.01608045  181
2920 06610000 2003-09-28 28700 -0.08926147 -0.01784872  184
2921 06610000 2003-09-29 28600 -0.08754373 -0.02108233  189
2922 06610000 2003-09-30 28700 -0.08577546 -0.02133474  201
         seda30       seda1
2917 -0.1546322 -0.10766231
```

```
2918 -0.1566163 -0.10321762
2919 -0.1579888 -0.09213983
2920 -0.1588294 -0.08416002
2921 -0.1586754 -0.07267005
2922 -0.1569162 -0.04769499
```

```
> # get a description of the data including definitions of the columns
> # by viewing the help documentation
> ?cqwMoRivOmaha
```

In this case, the continuous ancillary data includes daily streamflow (dflow) and daily sediment concentration (dsed), as well as the 30-day and 1-day streamflow (flowa30 and flowa1) and sediment (seda30 and seda1) anomalies. [The anomalies were calculated using the **waterData** package for R (Ryberg and Vecchia, 2012).]

In order to build a model using one or more of these ancillary variables as explanatory variables for pesticide concentration, the continuous ancillary variables need to be associated with the water-quality samples. The function *combineData* will combine water-quality sample data and continuous (daily) ancillary variables and drop unnecessary columns. One needs to specify the water-quality sample data, the continuous ancillary data, and the columns representing the station identifier (staid), the sample date, the qualification code (<, E) columns, and the concentration columns as shown in the following code. See Oblinger Childress (1999) for an explanation of the qualification codes used by the U.S. Geological Survey.

```
> data(swData)
> MoRivOmaha<-combineData(qwdat=qwMoRivOmaha, cqwdat=cqwMoRivOmaha,
+ qwcols=c("staid", "dates", "R", "P"))
> # view combined data set
> head(MoRivOmaha)
```

```
     staid      dates R04035 P04035 R04037 P04037 R04041 P04041
1 06610000 1996-01-13      <  0.004      _  0.024      <  0.008
2 06610000 1996-02-13      <  0.004      E  0.005      <  0.008
3 06610000 1996-03-13      E  0.005      E  0.004      <  0.008
4 06610000 1996-03-28      <  0.004      E  0.005      _  0.009
5 06610000 1996-04-09      _  0.007      E  0.006      _  0.014
6 06610000 1996-04-23      <  0.004      <  0.004      _  0.012
  R39415 P39415 R46342 P46342 R82630 P82630 R82661 P82661 R82668
1      _  0.006      <  0.003      _  0.029      <  0.003      _
2      _  0.200      <  0.003      <  0.007      <  0.003      _
3      _  0.026      <  0.003      <  0.007      <  0.003      E
4      _  0.026      <  0.003      <  0.007      <  0.003      _
5      _  0.075      E  0.003      <  0.007      <  0.003      _
6      _  0.040      <  0.003      <  0.007      <  0.003      <
  P82668 dflow      flowa30      flowa1 dsed      seda30
1  0.008 25800 -0.111771936 -0.041600453  255  0.04313266
```

```
2  0.007 30500 -0.155914620  0.075222364  312  0.02706313
3  0.004 32600 -0.043752697 -0.008021798  236 -0.02856792
4  0.008 42400 -0.004315925  0.066689687  609  0.01503934
5  0.009 50300  0.073100169  0.063475721  528  0.13734452
6  0.002 48800  0.126711034 -0.003283307  368  0.23175763
        seda1
1 -0.14439969
2 -0.04071575
3 -0.10632729
4  0.26177074
5  0.07748219
6 -0.17371703
```

3 Fitting the seawaveQ Model

One can now fit the seawaveQ model using the data explored and combined in the previous code examples. The following code fits three different seawaveQ models (with differing continuous ancillary variables) for two pesticides in the data set. The pesticides are 04035, simazine, and 04041, cyanazine. See the help documentation for further information about the function arguments shown and additional arguments.

```
> data(swData)
> # associate continuous water-quality data with each sample
> # combineData does this for you
> modMoRivOmaha<-combineData(qwdat=qwMoRivOmaha, cqwdat=cqwMoRivOmaha)
> # then fit model(s)
> myfit1 <- fitswavecav(cdat=modMoRivOmaha, cavdat=cqwMoRivOmaha,
+ tanm="myfit1", pnames=c("04035", "04041"), yrstart=1995,
+ yrend=2003, tndbeg=1995, tndend=2003, iwcav=c("flowa30", "flowa1"),
+ dcol="dates", qwcols=c("R","P"))
> myfit2 <- fitswavecav(cdat=modMoRivOmaha, cavdat=cqwMoRivOmaha,
+ tanm="myfit2", pnames=c("04035", "04041"), yrstart=1995,
+ yrend=2003, tndbeg=1995, tndend=2003, iwcav=c("seda30", "seda1"),
+ dcol="dates", qwcols=c("R","P"))
> myfit3 <- fitswavecav(cdat=modMoRivOmaha, cavdat=cqwMoRivOmaha,
+ tanm="myfit3", pnames=c("04035", "04041"), yrstart=1995,
+ yrend=2003, tndbeg=1995, tndend=2003, iwcav=c("flowa30", "flowa1",
+ "seda30", "seda1"), dcol="dates", qwcols=c("R","P"))
```

4 Model Output

The model fitting process finds the best pulse input function and model half-life for the concentration data and uses survival regression to fit a regression model. Three types of output are provided: (1) a list, the first element being a data frame with information about the model and its parameters, the second element being the survival regression summary, the third element the observed concentration (censored and uncensored), the fourth element the concentrations predicted by the model, and the fifth element the summary statistics for the predicted concentrations; (2) text files showing a summary of the survival regression results, like the second element of the list, but with additional measures of model quality and information about the R session; and (3) a pdf file of plots showing the model, trend, and diagnostic plots. The data frame results for the three models for simazine and cyanazine are shown below.

```
> # get the first element of the list for each model/constituent combination
> # the data frame with information about each model/constituent combination
> myfit1[[1]]

  pname mclass jmod hlife   cmaxt     scl    loglik     cint
1 04035      1    3     1 0.48087 0.27199 -32.79163 -2.49349
2 04041      1    4     3 0.48087 0.42405 -43.05521 -2.26965
    cwave     ctnd cflowa30 cflowa1   seint  sewave   setnd
1 0.55437 -0.02793  0.05386 2.10427 0.04910 0.11172 0.02112
2 2.17692 -0.24887 -0.02829 2.98901 0.08534 0.25938 0.03683
  seflowa30 seflowa1 pvaltnd
1   0.30492  0.46888 0.18598
2   0.51375  0.78007 0.00000

> myfit2[[1]]

  pname mclass jmod hlife   cmaxt     scl    loglik     cint
1 04035      1    5     3 0.48087 0.24677 -24.32398 -2.68378
2 04041      1    3     2 0.48087 0.40826 -39.64286 -2.12752
    cwave     ctnd cseda30  cseda1   seint  sewave   setnd
1 0.79650 -0.02405 0.25361 0.59703 0.06852 0.18671 0.01585
2 1.58343 -0.22011 1.13733 0.63650 0.07514 0.18937 0.02931
  seseda30 seseda1 pvaltnd
1  0.17293 0.10253 0.12907
2  0.28745 0.18127 0.00000

> myfit3[[1]]

  pname mclass jmod hlife   cmaxt     scl    loglik     cint
1 04035      1    5     3 0.48087 0.24621 -23.83644 -2.67838
2 04041      1    3     2 0.48087 0.38981 -37.63645 -2.07403
    cwave     ctnd cflowa30  cflowa1 cseda30  cseda1   seint
1 0.76533 -0.01817  0.17914 -0.36126 0.22732 0.66650 0.06724
2 1.71880 -0.24112 -0.61646  1.58972 1.03880 0.29012 0.07836
```

```
   sewave   setnd seflowa30 seflowa1 seseda30 seseda1 pvaltnd
1 0.18443 0.01918   0.30546  0.64210  0.20464 0.15225 0.34346
2 0.19990 0.03359   0.49387  1.11121  0.30870 0.26845 0.00000
```

```
> # get the second element of the list for each model/constituent combination
> # the survival regression summary for each model/constituent combination
> myfit1[[2]]
```

```
[[1]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~ 
    xmat - 1, dist = "gaussian")
              Value Std. Error       z        p
xmatintcpt  -2.4935     0.0491 -50.782 0.00e+00
xmatwavest   0.5544     0.1117   4.962 6.97e-07
xmattndlin  -0.0279     0.0211  -1.323 1.86e-01
xmatflowa30  0.0539     0.3049   0.177 8.60e-01
xmatflowa1   2.1043     0.4689   4.488 7.20e-06
Log(scale)  -1.3020     0.1205 -10.804 3.31e-27

Scale= 0.272 

Gaussian distribution
Loglik(model)= -32.8   Loglik(intercept only)= -60.9
        Chisq= 56.16 on 4 degrees of freedom, p= 1.9e-11 
Number of Newton-Raphson Iterations: 5 
n= 115 


[[2]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~ 
    xmat - 1, dist = "gaussian")
              Value Std. Error        z         p
xmatintcpt  -2.2697     0.0853 -26.5946 7.85e-156
xmatwavest   2.1769     0.2594   8.3929  4.74e-17
xmattndlin  -0.2489     0.0368  -6.7576  1.40e-11
xmatflowa30 -0.0283     0.5138  -0.0551  9.56e-01
xmatflowa1   2.9890     0.7801   3.8317  1.27e-04
Log(scale)  -0.8579     0.1036  -8.2828  1.20e-16

Scale= 0.424 

Gaussian distribution
```

```
Loglik(model)= -43.1   Loglik(intercept only)= -103.7
        Chisq= 121.33 on 4 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 6
n= 115
```

```
> myfit2[[2]]
```

```
[[1]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
             Value Std. Error      z        p
xmatintcpt -2.6838     0.0685 -39.17 0.00e+00
xmatwavest  0.7965     0.1867   4.27 1.99e-05
xmattndlin -0.0241     0.0158  -1.52 1.29e-01
xmatseda30  0.2536     0.1729   1.47 1.43e-01
xmatseda1   0.5970     0.1025   5.82 5.78e-09
Log(scale) -1.3993     0.1184 -11.82 3.11e-32

Scale= 0.247

Gaussian distribution
Loglik(model)= -24.3   Loglik(intercept only)= -60.9
        Chisq= 73.09 on 4 degrees of freedom, p= 5e-15
Number of Newton-Raphson Iterations: 6
n= 115


[[2]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
            Value Std. Error      z         p
xmatintcpt -2.128     0.0751 -28.31 2.31e-176
xmatwavest  1.583     0.1894   8.36  6.18e-17
xmattndlin -0.220     0.0293  -7.51  5.96e-14
xmatseda30  1.137     0.2874   3.96  7.60e-05
xmatseda1   0.636     0.1813   3.51  4.46e-04
Log(scale) -0.896     0.1027  -8.73  2.65e-18

Scale= 0.408

Gaussian distribution
Loglik(model)= -39.6   Loglik(intercept only)= -103.7
```

```
        Chisq= 128.16 on 4 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 6
n= 115

> myfit3[[2]]

[[1]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
              Value Std. Error       z        p
xmatintcpt  -2.6784     0.0672 -39.830 0.00e+00
xmatwavest   0.7653     0.1844   4.150 3.33e-05
xmattndlin  -0.0182     0.0192  -0.947 3.43e-01
xmatflowa30  0.1791     0.3055   0.586 5.58e-01
xmatflowa1  -0.3613     0.6421  -0.563 5.74e-01
xmatseda30   0.2273     0.2046   1.111 2.67e-01
xmatseda1    0.6665     0.1522   4.378 1.20e-05
Log(scale)  -1.4016     0.1188 -11.802 3.79e-32

Scale= 0.246

Gaussian distribution
Loglik(model)= -23.8   Loglik(intercept only)= -60.9
        Chisq= 74.07 on 6 degrees of freedom, p= 6e-14
Number of Newton-Raphson Iterations: 6
n= 115


[[2]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
             Value Std. Error      z         p
xmatintcpt  -2.074     0.0784 -26.47 2.27e-154
xmatwavest   1.719     0.1999   8.60  8.10e-18
xmattndlin  -0.241     0.0336  -7.18  7.02e-13
xmatflowa30 -0.616     0.4939  -1.25  2.12e-01
xmatflowa1   1.590     1.1112   1.43  1.53e-01
xmatseda30   1.039     0.3087   3.37  7.65e-04
xmatseda1    0.290     0.2684   1.08  2.80e-01
Log(scale)  -0.942     0.1033  -9.12  7.53e-20

Scale= 0.39
```

```
Gaussian distribution
Loglik(model)= -37.6   Loglik(intercept only)= -103.7
        Chisq= 132.17 on 6 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 6
n= 115
```

```
> # get the first few lines of the third element of the list
> head(myfit1[[3]])
```

```
   dectime R04035 P04035 R04041 P04041
1 1996.034      <  0.004      <  0.008
2 1996.117      <  0.004      <  0.008
3 1996.201         0.005      <  0.008
4 1996.242      <  0.004         0.009
5 1996.273         0.007         0.014
6 1996.311      <  0.004         0.012
```

```
> head(myfit2[[3]])
```

```
   dectime R04035 P04035 R04041 P04041
1 1996.034      <  0.004      <  0.008
2 1996.117      <  0.004      <  0.008
3 1996.201         0.005      <  0.008
4 1996.242      <  0.004         0.009
5 1996.273         0.007         0.014
6 1996.311      <  0.004         0.012
```

```
> head(myfit3[[3]])
```

```
   dectime R04035 P04035 R04041 P04041
1 1996.034      <  0.004      <  0.008
2 1996.117      <  0.004      <  0.008
3 1996.201         0.005      <  0.008
4 1996.242      <  0.004         0.009
5 1996.273         0.007         0.014
6 1996.311      <  0.004         0.012
```

```
> # get the first few lines of the fourth element of the list
> head(myfit1[[4]])
```

```
   dectime      P04035      P04041
1 1995.831 0.002184579 0.007085318
2 1995.833 0.002223730 0.007255036
3 1995.835 0.002256099 0.007295828
4 1995.837 0.002301522 0.007390105
5 1995.840 0.002361534 0.007548559
6 1995.843 0.002297257 0.007149901
```

```
> head(myfit2[[4]])
```

```
   dectime      P04035     P04041
1 1995.831 0.001539907 0.01452533
2 1995.833 0.001542766 0.01483602
3 1995.835 0.001543323 0.01511136
4 1995.837 0.001542208 0.01535307
5 1995.840 0.001513703 0.01517149
6 1995.843 0.001430205 0.01425638

> head(myfit3[[4]])

   dectime      P04035     P04041
1 1995.831 0.001708573 0.01084807
2 1995.833 0.001705451 0.01121241
3 1995.835 0.001701741 0.01149961
4 1995.837 0.001694584 0.01181777
5 1995.840 0.001653297 0.01198216
6 1995.843 0.001560628 0.01134960

> # get the summary of predicted concentrations
> myfit1[[5]]

  analysis pname predMeanConc predQ10 predQ25 predQ50 predQ75
1   myfit1 04035      0.00271 0.00117 0.00150 0.00203 0.00350
2   myfit1 04041      0.01576 0.00013 0.00043 0.00218 0.00831
  predQ90
1 0.00535
2 0.04093

> myfit2[[5]]

  analysis pname predMeanConc predQ10 predQ25 predQ50 predQ75
1   myfit2 04035      0.00257 0.00085 0.00113 0.00180 0.00339
2   myfit2 04041      0.01618 0.00018 0.00057 0.00237 0.00942
  predQ90
1 0.00528
2 0.03975

> myfit3[[5]]

  analysis pname predMeanConc predQ10 predQ25 predQ50 predQ75
1   myfit3 04035      0.00259 0.00093 0.00119 0.00179 0.00341
2   myfit3 04041      0.01678 0.00021 0.00060 0.00251 0.00895
  predQ90
1 0.00529
2 0.03893

>
```

The first element of the list, the data frame, contains information about each model including the pesticide analyzed; the model class (an option in **seawaveQ** that is not currently implemented but that will provide additional model options in the future); the choice of model or pulse input function, an integer 1 through 14; the model half-life in months, an integer, 1 to 4 months; the decimal season of maximum concentration; the scale factor from the survreg.object; the log-likelihood for the model; the coefficient for the model intercept; the coefficient for the seasonal wave; the coefficient for the trend component of the model; 0 or more values representing coefficients for the continuous ancillary variables; the standard error for the intercept; the standard error for the seasonal wave; the standard error for the trend; and 0 or more columns representing standard errors for the continuous ancillary variables.

The second element of the list is provided so that users could extract the attributes of the survival regression summary programmatically (rather than viewing them in the text file) and create their own summaries or plots of the results. The third, fourth, and fifth elements of the list are provided for user-generated plots and further user analysis.

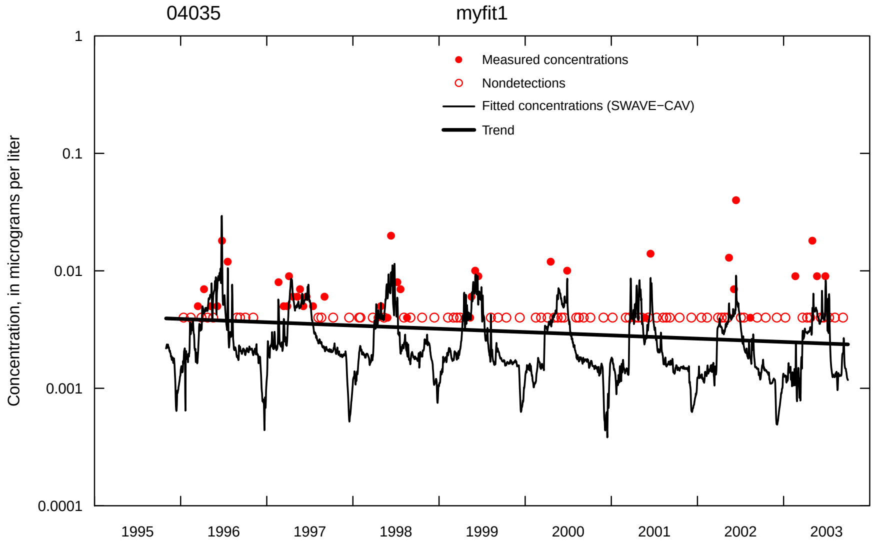
```
> attributes(myfit1[[2]][[1]])

$names
 [1] "call"         "df"           "loglik"       "iter"
 [5] "idf"          "scale"        "coefficients" "var"
 [9] "table"        "correlation"  "parms"        "n"
[13] "chi"          "robust"

$class
[1] "summary.survreg"

> myfit1[[2]][[1]]$n

[1] 115

> myfit1[[2]][[1]]$table

                  Value Std. Error           z            p
xmatintcpt  -2.49349164 0.04910211 -50.7817616 0.000000e+00
xmatwavest   0.55436953 0.11171616   4.9623039 6.966189e-07
xmattndlin  -0.02793350 0.02112089  -1.3225534 1.859839e-01
xmatflowa30  0.05385545 0.30491533   0.1766243 8.598035e-01
xmatflowa1   2.10426938 0.46888353   4.4878296 7.195245e-06
Log(scale)  -1.30197275 0.12051232 -10.8036489 3.307909e-27

>
```

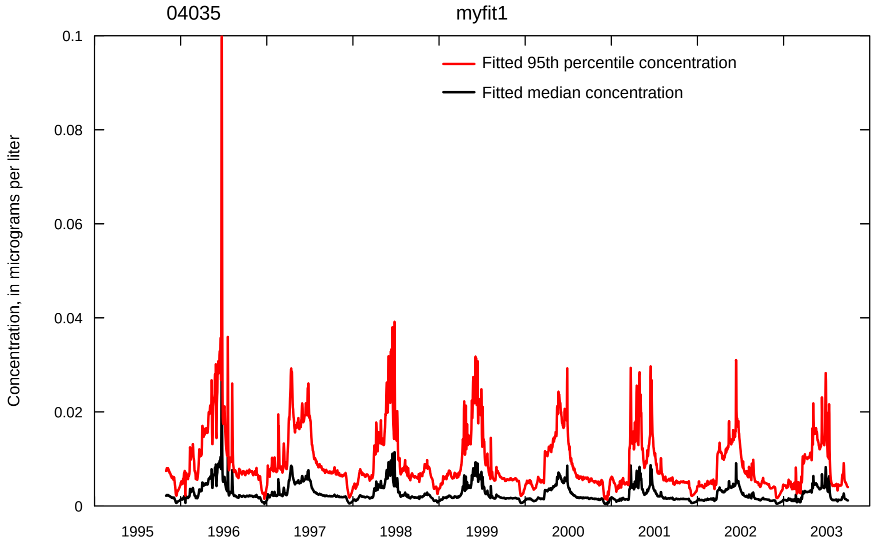
The text file for the first of the function calls above is inserted here as an example. Users may run the model fitting code themselves and view the resulting text files for all three models. The results for all three are too long to include in this vignette.

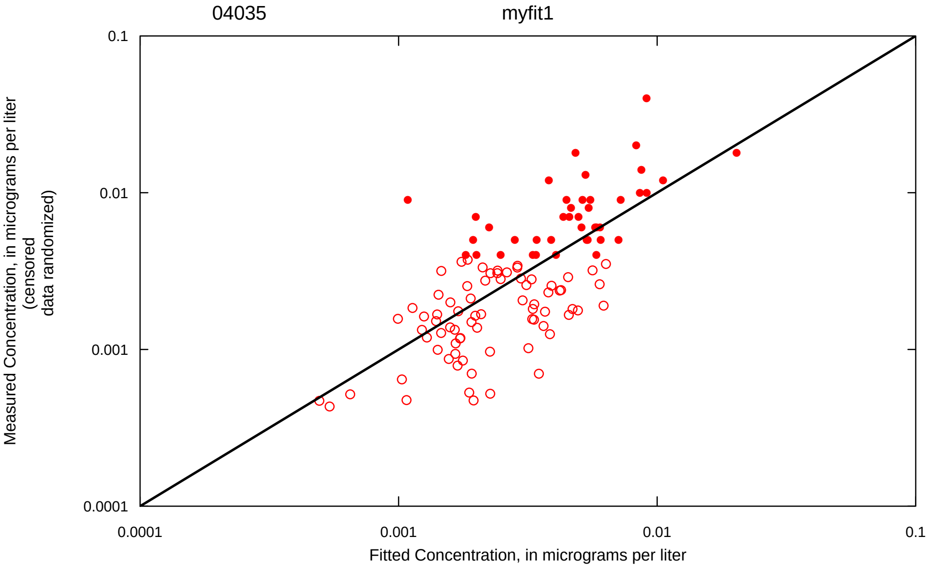
```
Monday 30 Dec 2013 14:39:55 PM CST
R version 3.0.2 (2013-09-25)
seawaveQ version 1.0.0
x86_64-apple-darwin10.8.0 (64-bit)

Final model survreg results for 04035
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
              Value Std. Error       z        p
xmatintcpt  -2.4935     0.0491 -50.782 0.00e+00
xmatwavest   0.5544     0.1117   4.962 6.97e-07
xmattndlin  -0.0279     0.0211  -1.323 1.86e-01
xmatflowa30  0.0539     0.3049   0.177 8.60e-01
xmatflowa1   2.1043     0.4689   4.488 7.20e-06
Log(scale)  -1.3020     0.1205 -10.804 3.31e-27

Scale= 0.272

Gaussian distribution
Loglik(model)= -32.8   Loglik(intercept only)= -60.9
Chisq= 56.16 on 4 degrees of freedom, p= 1.9e-11
Number of Newton-Raphson Iterations: 5
n= 115

AIC (Akaike's An Information Criterion) is:  77.58325
BIC (Bayesian Information Criterion) is:  94.05285
Model class is 1
Pulse input function is 3
Half life is 1
Seasonal value of the maximum concentration is 0.4808743.


Monday 30 Dec 2013 14:39:56 PM CST
R version 3.0.2 (2013-09-25)
seawaveQ version 1.0.0
x86_64-apple-darwin10.8.0 (64-bit)

Final model survreg results for 04041
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian")
              Value Std. Error        z         p
xmatintcpt  -2.2697     0.0853 -26.5946 7.85e-156
```

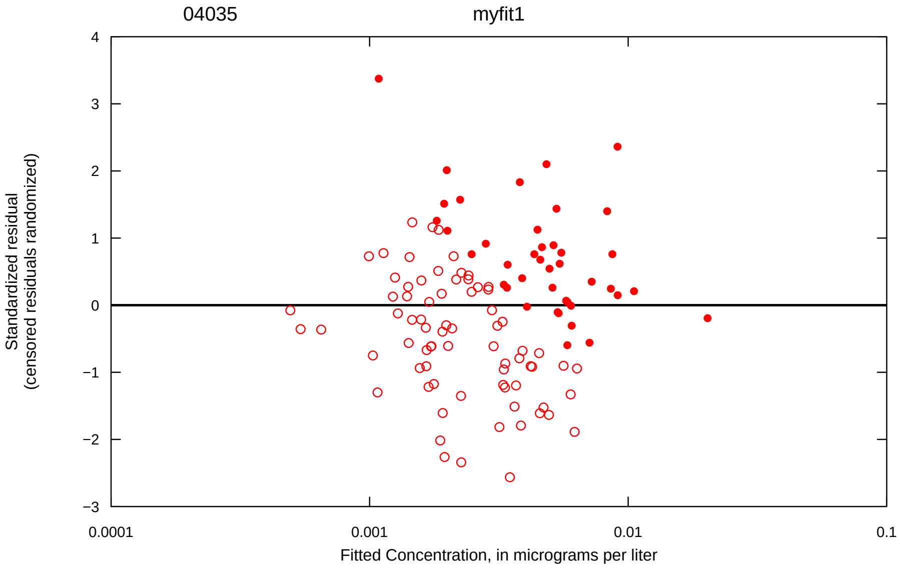
```
xmatwavest   2.1769     0.2594   8.3929  4.74e-17
xmattndlin  -0.2489     0.0368  -6.7576  1.40e-11
xmatflowa30 -0.0283     0.5138  -0.0551  9.56e-01
xmatflowa1   2.9890     0.7801   3.8317  1.27e-04
Log(scale)  -0.8579     0.1036  -8.2828  1.20e-16

Scale= 0.424

Gaussian distribution
Loglik(model)= -43.1   Loglik(intercept only)= -103.7
Chisq= 121.33 on 4 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 6
n= 115

AIC (Akaike's An Information Criterion) is:  98.11042
BIC (Bayesian Information Criterion) is:  114.58
Model class is 1
Pulse input function is 4
Half life is 3
Seasonal value of the maximum concentration is 0.4808743.
```

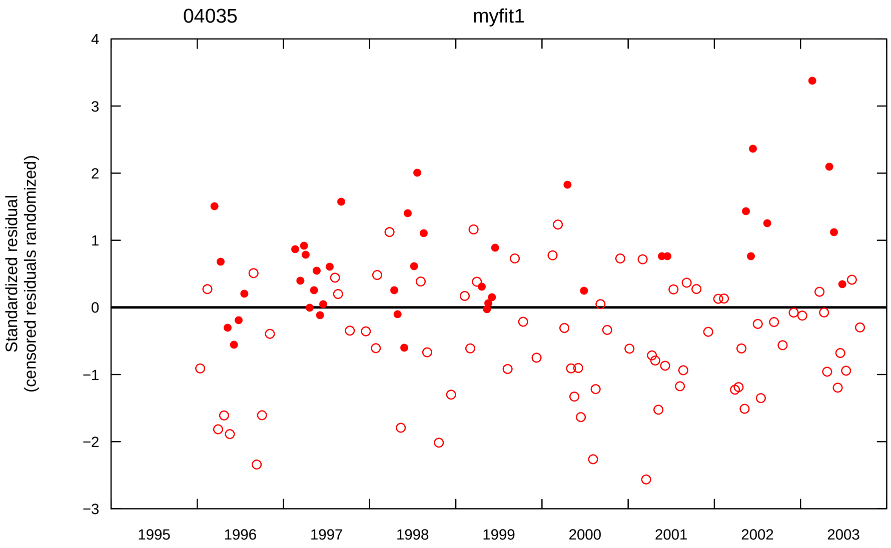
The plots written to a pdf file for the first pesticide, 04035, simazine, in the first model, myfit1, are included below. As with the text results, the plots for all three models and all pesticides are too numerous to include here. Users are encouraged to run the code themselves and examine all of the plots.

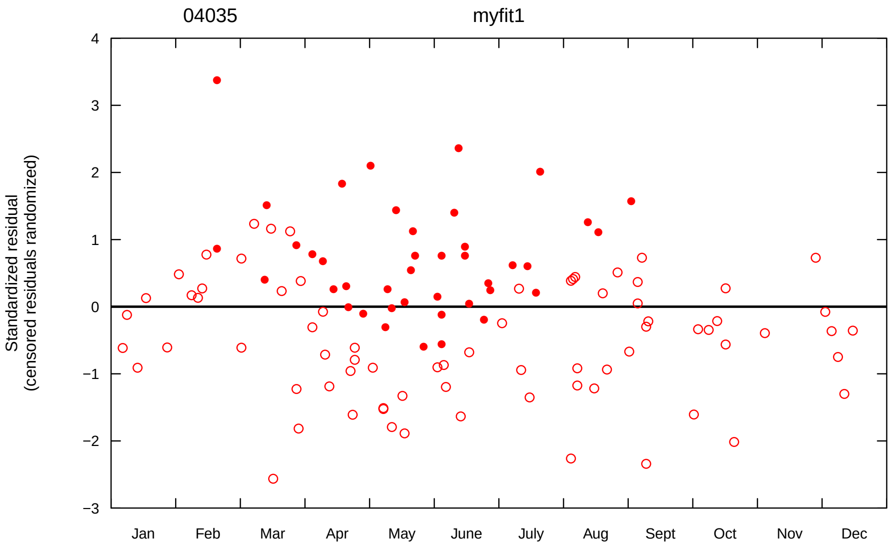

04035
myfit1
● Measured concentrations
○ Nondetections
Fitted concentrations (SWAVE−CAV)
Trend
Concentration, in micrograms per liter
1
0.1
0.01
0.001
0.0001
1995
1996
1997
1998
1999
2000
2001
2002
2003


04035
myfit1
Fitted 95th percentile concentration
Fitted median concentration
Concentration, in micrograms per liter
0.1
0.08
0.06
0.04
0.02
0
1995
1996
1997
1998
1999
2000
2001
2002
2003


04035
myfit1
Measured Concentration, in micrograms per liter (censored data randomized)
0.1
0.01
0.001
0.0001
0.0001
0.001
0.01
0.1
Fitted Concentration, in micrograms per liter


04035
myfit1
Standardized residual (censored residuals randomized)
4
3
2
1
0
−1
−2
−3
0.0001
0.001
0.01
0.1
Fitted Concentration, in micrograms per liter


04035
myfit1
Standardized residual
(censored residuals randomized)
4
3
2
1
0
−1
−2
−3
1995
1996
1997
1998
1999
2000
2001
2002
2003


04035
myfit1
Standardized residual
(censored residuals randomized)
4
3
2
1
0
−1
−2
−3
Jan
Feb
Mar
Apr
May
June
July
Aug
Sept
Oct
Nov
Dec

The plotting position used for representing censored values in the model plots (produced by the internal function *seawaveQPlots* that is further described in the package help documentation) is an important consideration for interpreting model fit. Plotting values obtained by using the censoring limit, or something smaller such as one-half of the censoring limit, produce plots that are difficult to interpret if there are a large number of censored values. Therefore, to make the plots more representative of diagnostic plots used for standard (non-censored) regression, a method for substituting randomized residuals in place of censored residuals was used. If a log-transformed concentration is censored at a particular limit, $logC < L$, then the residual for that concentration is censored as well, $logC - fitted(logC) < L - fitted(logC) = rescen)$. In that case, a randomized residual was generated from a conditional normal distribution, as shown in the following R code:

```
resran  <-  scl * qnorm(runif(1) * pnorm(rescen / scl))
```

where *scl* is the scale parameter from the survival regression model, *pnorm* is the R function for computing cumulative normal probabilities, *runif* is the R function for generating a random variable from the uniform distribution, and *qnorm* is the R function for computing quantiles of the normal distribution. Under the assumption that the model residuals are uncorrelated, normally distributed random variables with mean zero and standard deviation *scl*, the randomized residuals generated in this manner are an unbiased sample of the true (but unknown) residuals for the censored data. This is an application of the probability integral transform (Mood and others, 1974) to generate random variables from continuous distributions. The plotting position using a censored concentration is $fitted(logC) + resran$. Note that each time a new model fit is performed, a new set of randomized residuals is generated and thus the plotting positions for censored values can change.

5 References Cited

Helsel, D.R., 2005, Nondetects and data analysis: New York, John Wiley and Sons.

Lee, Lopaka, 2012, Nondetects and data analysis for environmental data: R package version 1.5-4, http://CRAN.R-project.org/package=NADA.

Mood, A.M., Graybill, F.A., and Boes, D.C., 1974, Introduction to the theory of statistics (3d ed.): New York, McGraw-Hill, Inc., 564 p.

Oblinger Childress, C.J., Foreman,W.T., Connor, B.F., and Maloney, T.J., 1999, New reporting procedures based on long-term method detection levels and some considerations for interpretations of water-quality data provided by the U.S. Geological Survey: U.S. Geological Survey Open-File Report 99–193, 19 p. (Also available at http://water.usgs.gov/owq/OFR_99-193/index.html.)

Ryberg, K.R. and Vecchia, A.V., 2012, waterData—An R package for retrieval, analysis, and anomaly calculation of daily hydrologic time series data, version 1.0: U.S. Geological Survey Open-File Report 2012–1168; 8 p., accessed March 1, 2013, at http://pubs.usgs.gov/of/2012/1168/.

Ryberg, K.R. and Vecchia, A.V., 2013, seawaveQ—An R package providing a model and utilities for analyzing trends in chemical concentrations in streams with a seasonal wave (seawave) and adjustment for streamflow (Q) and other ancillary variables: U.S. Geological Survey Open-File Report 2013–1255, 13 p., with 3 appendixes, http://dx.doi.org/10.3133/ofr20131255.

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., accessed May 1, 2012, at http://pubs.usgs.gov/sir/2010/5139/.

Sullivan, D.J., Vecchia, A.V., Lorenz, D.L., Gilliom, R.J., and Martin, J.D., 2009, Trends in pesticide concentrations in corn-belt streams, 1996–2006: U.S. Geological Survey Scientific Investigations Report 2009–5132; 75 p., accessed May 1, 2012, at http://pubs.usgs.gov/sir/2009/5132/.

Vecchia, A.V., Gilliom, R.J., Sullivan, D.J., Lorenz, D.L., and Martin, J.D., 2009, Trends in concentrations and use of agricultural herbicides for Corn Belt rivers, 1996–2006: Environmental Science and Technology, v. 43; p. 9,096-9,102, accessed May 1, 2012, at http://water.usgs.gov/nawqa/pubs/es902122j.pdf.

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5; pp. 1,308-1,324, accessed May 1, 2012, at http://dx.doi.org/10.1111/j.1752-1688.2008.00225.x.