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

July 27, 2020

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 model and makes the trend analysis method widely available for use by others. In addition, several enhancements to the 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 York, 2020).

The statistical methodology for the model is described in Vecchia and others (2008) and in the U.S. Geological Survey Open-File Report documenting this package (Ryberg and York, 2020). 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 York, 2020).

2 Input Data

The model needs two types of input data. The first is 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.

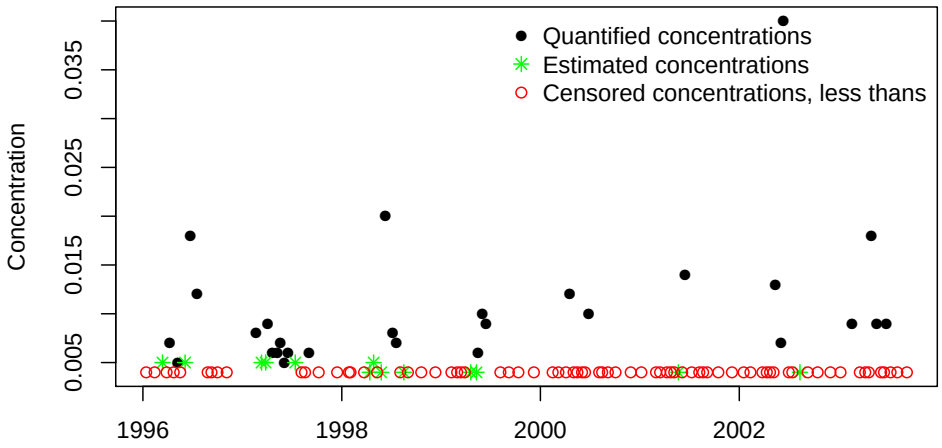
```
> options(width = 65)
> # load necessary packages, assuming they have already been installed on the system
> library(waterData)
> library(survival)
> library(seawaveQ)
> # load example data that comes with the package
> data(swData)
> # show first few rows of water-quality data for Missouri River 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
```

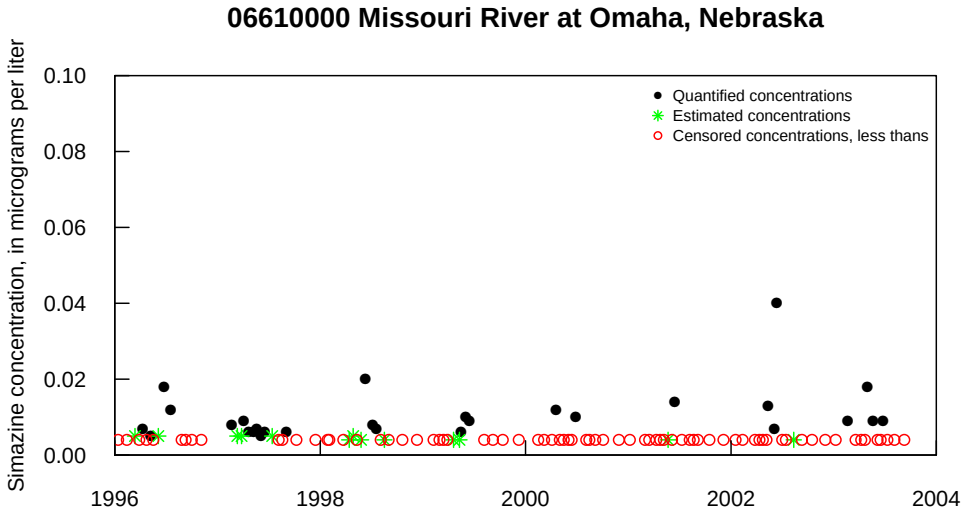
```
> # get a description of the data including definitions of the columns
> # by viewing the help documentation
> ?qwMoRivOmaha
```

Optionally, a function has been provided to plot concentration data. This function produces a scatter plot that indicates the censored, less than, values. The functions is *cenScatPlot* and an example of its use follows. Previous versions of the software included an option to generate box plots properly representing regression on order statistics using an implementation of a regression on order statistics designed for multiply-censored analytical-chemistry data (Helsel, 2005). This relied on the R package **NADA** (Lee, 2017); however, long-term maintenance of this package has been an issue (oral commun., R Core Team, 2020) and the functionality was removed. Users may investigate the availability of **NADA** and use it separately to produce box plots.

```
> # 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, Nebraska",
+             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, xaxs = "i", xaxt = "n",
+             xlim = c(as.Date("1996-01-01"), as.Date("2004-01-01")))
> 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)
```



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 and others (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 SEAWAVE-Q Model

One can now fit the model using the data explored and combined in the previous code examples. The following code fits three different models (with differing continuous ancillary variables) for two pesticides in the data set. The pesticides are 04035, simazine; 04037, prometon; 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)
> myfitLinearTrend <- fitswavecav(cdat = modMoRivOmaha, cavdat = cqwMoRivOmaha,
+                                 tanm = "myfitLinearTrend",
+                                 pnames = c("04035", "04037", "04041"),
+                                 yrstart = 1995, yrend = 2003, tndbeg = 1995,
+                                 tndend = 2003, iwcav = c("flowa30", "flowa1"),
+                                 dcol = "dates", qwcols = c("R", "P"), mclass = 1)
```

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, the fifth element the summary statistics for the predicted concentrations; and the sixth element provides trend results expressed as trends in percent or trends in concentration units; 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
> myfitLinearTrend[[1]]

  pname mclass jmod hlife   cmaxt     scl    loglik cxmatintcpt
1 04035      1    3     1 0.48087 0.27199 -32.79163    -2.49349
2 04037      1    8     4 0.42896 0.19970  -9.96412    -2.31618
3 04041      1    4     3 0.48087 0.42405 -43.05521    -2.26965
  cxmatwavest cxmattndlin cxmatflowa30 cxmatflowa1 sexmatintcpt
1     0.55437    -0.02793      0.05386     2.10427      0.04910
2     0.27310    -0.00733      0.09402     0.34218      0.02550
3     2.17692    -0.24887     -0.02829     2.98901      0.08534
  sexmatwavest sexmattndlin sexmatflowa30 sexmatflowa1
1      0.11172      0.02112       0.30492      0.46888
2      0.07609      0.01202       0.16905      0.26856
3      0.25938      0.03683       0.51375      0.78007
  pvalxmattndlin
1        0.18598
2        0.54224
3        0.00000

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

[[1]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
              Value Std. Error      z       p
xmatintcpt  -2.4935     0.0491 -50.78 < 2e-16
```

```
xmatwavest   0.5544     0.1117   4.96 7.0e-07
xmattndlin  -0.0279     0.0211  -1.32    0.19
xmatflowa30  0.0539     0.3049   0.18    0.86
xmatflowa1   2.1043     0.4689   4.49 7.2e-06
Log(scale)  -1.3020     0.1205 -10.80 < 2e-16

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", control = list(maxiter = 100))
              Value Std. Error      z       p
xmatintcpt  -2.31618    0.02550 -90.84 < 2e-16
xmatwavest   0.27310    0.07609   3.59 0.00033
xmattndlin  -0.00733    0.01202  -0.61 0.54224
xmatflowa30  0.09402    0.16905   0.56 0.57811
xmatflowa1   0.34218    0.26856   1.27 0.20261
Log(scale)  -1.61096    0.08315 -19.37 < 2e-16

Scale= 0.2

Gaussian distribution
Loglik(model)= -10   Loglik(intercept only)= -20.2
        Chisq= 20.54 on 4 degrees of freedom, p= 0.00039
Number of Newton-Raphson Iterations: 4
n= 115


[[3]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
             Value Std. Error      z       p
xmatintcpt  -2.2697     0.0853 -26.59 < 2e-16
xmatwavest   2.1769     0.2594   8.39 < 2e-16
```

```
xmattndlin  -0.2489     0.0368  -6.76 1.4e-11
xmatflowa30 -0.0283     0.5138  -0.06 0.95609
xmatflowa1   2.9890     0.7801   3.83 0.00013
Log(scale)  -0.8579     0.1036  -8.28 < 2e-16

Scale= 0.424

Gaussian distribution
Loglik(model)= -43.1   Loglik(intercept only)= -103.7
        Chisq= 121.33 on 4 degrees of freedom, p= 2.8e-25
Number of Newton-Raphson Iterations: 6
n= 115
```

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

```
   dectime R04035 P04035 R04037 P04037 R04041 P04041
1 1996.034      <  0.004         0.024      <  0.008
2 1996.117      <  0.004         0.005      <  0.008
3 1996.201         0.005         0.004      <  0.008
4 1996.242      <  0.004         0.005         0.009
5 1996.273         0.007         0.006         0.014
6 1996.311      <  0.004      <  0.004         0.012
```

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

```
   dectime      P04035      P04037      P04041
1 1995.831 0.002184579 0.004008740 0.007085318
2 1995.833 0.002223730 0.004021064 0.007255036
3 1995.835 0.002256099 0.004022765 0.007295828
4 1995.837 0.002301522 0.004028182 0.007390105
5 1995.840 0.002361534 0.004037500 0.007548559
6 1995.843 0.002297257 0.004011441 0.007149901
```

```
> # get the summary of predicted concentrations
> myfitLinearTrend[[5]]
```

```
          analysis pname predMeanConc predQ10 predQ25 predQ50
1 myfitLinearTrend 04035      0.00271 0.00117 0.00150 0.00203
2 myfitLinearTrend 04037      0.00441 0.00353 0.00378 0.00415
3 myfitLinearTrend 04041      0.01576 0.00013 0.00043 0.00218
  predQ75 predQ90
1 0.00350 0.00535
2 0.00481 0.00581
3 0.00831 0.04093
```

```
> # get summary of trend results
> myfitLinearTrend[[6]]
```

```
  pname mclass alpha baseConc ctndPpor cuciPpor clciPpor
1 04035      1   0.1   0.0042 -40.2194  13.3638 -68.4757
2 04037      1   0.1   0.0052 -12.6305  25.7561 -39.2997
3 04041      1   0.1   0.0531 -98.9790 -96.8834 -99.6655
  ctndOrigPORPercentBase cuciOrigPORPercentBase
1                -0.0017                 0.0006
2                -0.0007                 0.0013
3                -0.0526                -0.0515
  clciOrigPORPercentBase ctndlklhd
1                -0.0028    0.9070
2                -0.0020    0.7289
3                -0.0530    1.0000
```

The first element of the list, the data frame, contains information about each model including the pesticide analyzed; the model class (a class 1 model incorporates a linear trend, a class 2 model incorporates restricted cubic splines); 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, fifth, and sixth elements of the list are provided for user-generated plots and further user analysis (see R help for more details).

```
> attributes(myfitLinearTrend[[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"

> myfitLinearTrend[[2]][[1]]$n

[1] 115

> myfitLinearTrend[[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
```

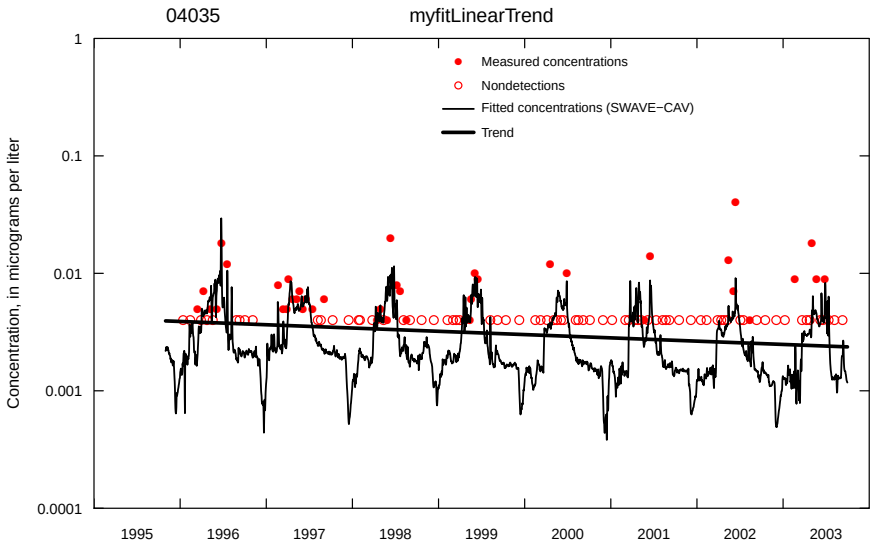
The text file for the first of the pesticides modeled above is inserted here as an example. Users may run the model fitting code themselves and view the resulting text files for all three pesticide models. The results for all three are too long to include in this vignette.

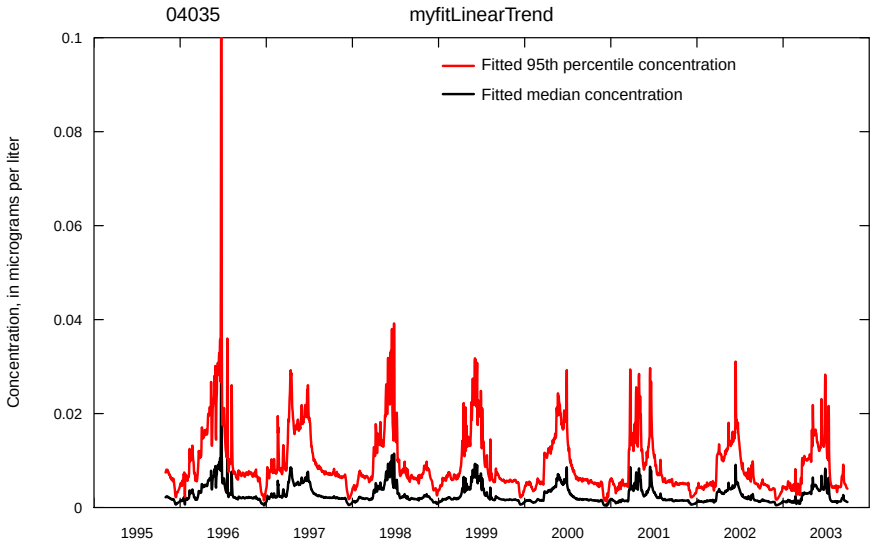
```
Monday 27 Jul 2020 16:00:51 PM CDT
R version 3.5.1 (2018-07-02)
seawaveQ version 2.0.0
x86_64-apple-darwin15.6.0 (64-bit)

Final model survreg results for 04035
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
             Value Std. Error      z       p
xmatintcpt  -2.4935     0.0491 -50.78 < 2e-16
xmatwavest   0.5544     0.1117   4.96 7.0e-07
xmattndlin  -0.0279     0.0211  -1.32    0.19
xmatflowa30  0.0539     0.3049   0.18    0.86
xmatflowa1   2.1043     0.4689   4.49 7.2e-06
Log(scale)  -1.3020     0.1205 -10.80 < 2e-16

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

Generalized r-squared is:  0.39
AIC (Akaike's An Information Criterion) is:  77.58
BIC (Bayesian Information Criterion) is:  94.05
Model class is 1
Pulse input function is 3
Half life is 1
Seasonal value of the maximum concentration is 0.48.
```

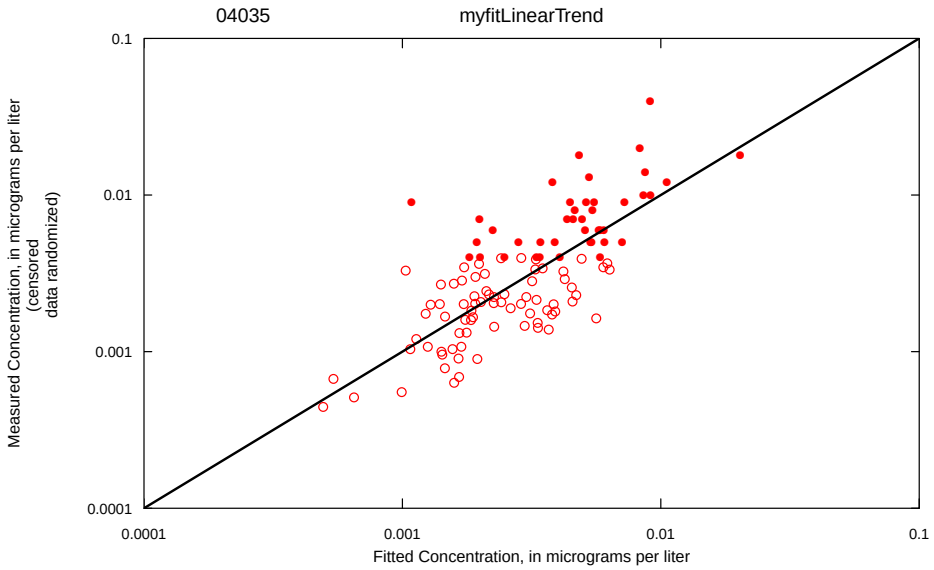
```
Monday 27 Jul 2020 16:00:51 PM CDT
R version 3.5.1 (2018-07-02)
seawaveQ version 2.0.0
x86_64-apple-darwin15.6.0 (64-bit)

Final model survreg results for 04037
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
              Value Std. Error      z       p
xmatintcpt  -2.31618    0.02550 -90.84 < 2e-16
xmatwavest   0.27310    0.07609   3.59 0.00033
xmattndlin  -0.00733    0.01202  -0.61 0.54224
xmatflowa30  0.09402    0.16905   0.56 0.57811
xmatflowa1   0.34218    0.26856   1.27 0.20261
Log(scale)  -1.61096    0.08315 -19.37 < 2e-16

Scale= 0.2

Gaussian distribution
Loglik(model)= -10   Loglik(intercept only)= -20.2
Chisq= 20.54 on 4 degrees of freedom, p= 0.00039
Number of Newton-Raphson Iterations: 4
n= 115

Generalized r-squared is:  0.16
AIC (Akaike's An Information Criterion) is:  31.93
BIC (Bayesian Information Criterion) is:  48.4
Model class is 1
Pulse input function is 8
Half life is 4
Seasonal value of the maximum concentration is 0.43.


Monday 27 Jul 2020 16:00:51 PM CDT
R version 3.5.1 (2018-07-02)
seawaveQ version 2.0.0
x86_64-apple-darwin15.6.0 (64-bit)

Final model survreg results for 04041
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
            Value Std. Error      z       p
```

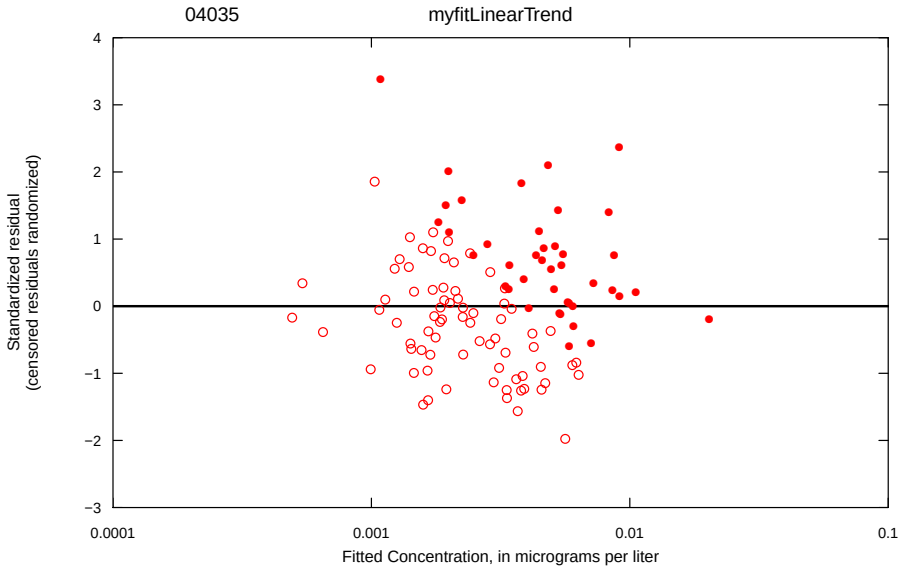
```
xmatintcpt  -2.2697     0.0853 -26.59 < 2e-16
xmatwavest   2.1769     0.2594   8.39 < 2e-16
xmattndlin  -0.2489     0.0368  -6.76 1.4e-11
xmatflowa30 -0.0283     0.5138  -0.06 0.95609
xmatflowa1   2.9890     0.7801   3.83 0.00013
Log(scale)  -0.8579     0.1036  -8.28 < 2e-16

Scale= 0.424

Gaussian distribution
Loglik(model)= -43.1   Loglik(intercept only)= -103.7
Chisq= 121.33 on 4 degrees of freedom, p= 2.8e-25
Number of Newton-Raphson Iterations: 6
n= 115

Generalized r-squared is:  0.65
AIC (Akaike's An Information Criterion) is:  98.11
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 0.48.
```

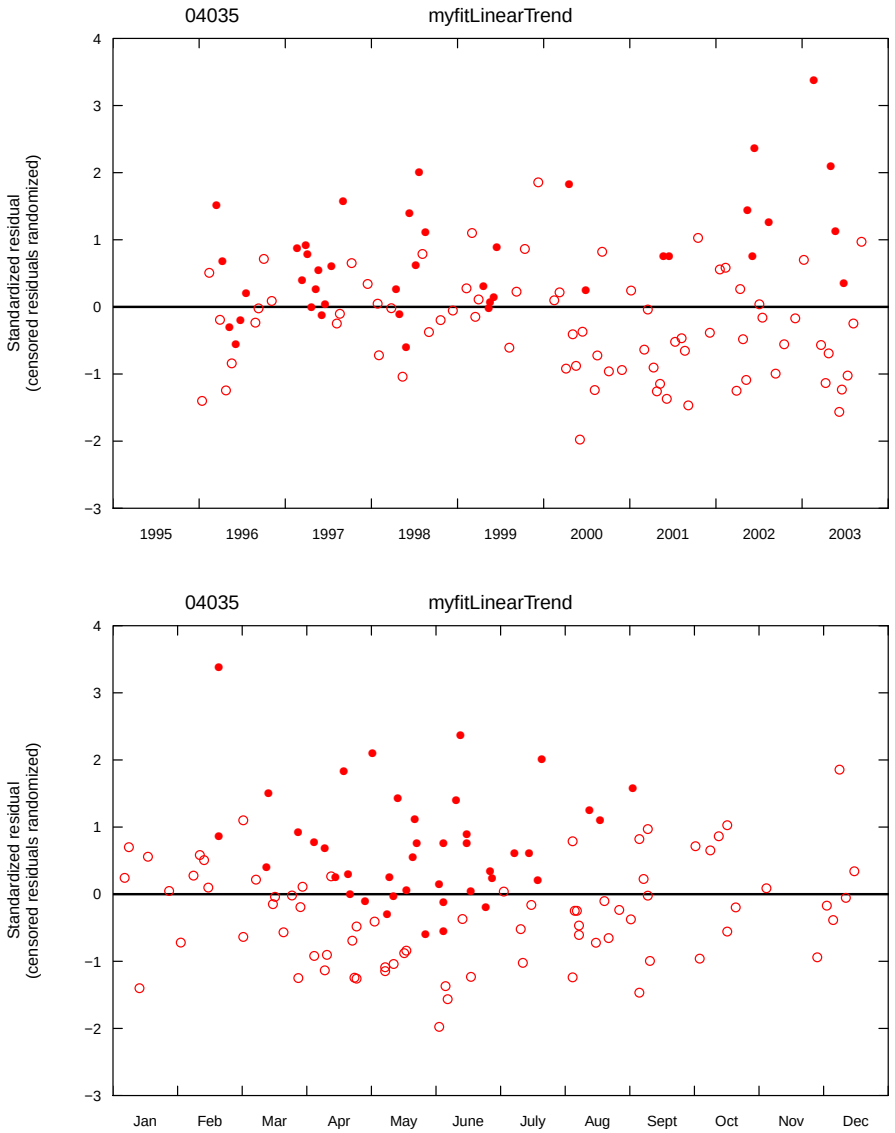
The plots written to a pdf file for the first pesticide, 04035, simazine are included below. The plots for all pesticides are too numerous to include here. Users are encouraged to run the code themselves and examine all of the plots.


04035
myfitLinearTrend
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
myfitLinearTrend
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
myfitLinearTrend
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
myfitLinearTrend
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
myfitLinearTrend
Standardized residual
(censored residuals randomized)
4
3
2
1
0
−1
−2
−3
1995
1996
1997
1998
1999
2000
2001
2002
2003
04035
myfitLinearTrend
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 Restricted Cubic Splines Option

The original SEAWAVE-Q model assumes a linear trend over a given trend period. Linearity is a common, useful assumption in trend analysis, but becomes problematic as trend periods become longer because of the potential for abrupt, or gradual, non-linear changes. Examples include cancellation of some or all of the uses of a particular pesticide, phase-outs of a pesticide, or changes in usage. As additional data are collected, longer trend periods may be calculated and a strict linear trend may be inappropriate for many site-pesticide combinations. Restricted cubic splines (transformations of the time variable) were incorporated into the SEAWAVE-Q model to allow flexibility over time. The range of values of the time variable is split up, with "knots" defining the end of one segment and the start of the next. Separate curves are then fit to each segment (Harrell, 2010, 2016). Overall, the splines are defined so that the resulting fitted curve is smooth and continuous, passing through each knot. Other studies have shown that 3 to 5 knots typically are sufficient for most models (Korn and Graubard, 1999). For small sample sizes (< 30, a sample size insufficient for a pesticide trend model), 3 knots can be used; 4 knots are sufficient for most models and represent a compromise between overfitting and model flexibility (Harrell, 2010; Croxford, 2016). If the sample size is large (>100, as is often the case for pesticide trend modeling) or the relation changes quickly over the trend period, 5 to 7 knots could be used, although 5 are usually sufficient (Stone, 1986; Croxford, 2016).

6 Fitting the SEAWAVE-Q Model with Restricted Cubic Splines

The following code fits a model with restricted cubic splines instead of a linear trend for one pesticide in the dataset. The pesticide is 04035, simazine. 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
> myfitRCS <- fitswavecav(cdat = modMoRivOmaha, cavdat = cqwMoRivOmaha,
+                         tanm = "myfitRCS", pnames = c("04035"), yrstart = 1995,
+                         yrend = 2003, tndbeg = 1995, tndend = 2003,
+                         iwcav = c("flowa30", "flowa1"), dcol = "dates",
+                         qwcols = c("R", "P"), mclass = 2, numk = 4)
```

7 Model Output

```
> # get the first element of the list for each model/constituent combination
> # the data frame with information about each model/constituent combination
> myfitRCS[[1]]
  pname mclass jmod hlife   cmaxt     scl    loglik cxmatintcpt
1 04035      2    3     1 0.48087 0.26702 -30.34875    -2.28508
  cxmatwavest cxmattndlin cxmattndlin' cxmattndlin''
1     0.58486     0.05451     -0.42262        1.2513
  cxmatflowa30 cxmatflowa1 sexmatintcpt sexmatwavest
1     -0.08385     2.07417      0.13916      0.11368
  sexmattndlin sexmattndlin' sexmattndlin'' sexmatflowa30
1      0.07055       0.23925         0.6444        0.3118
  sexmatflowa1 pvalxmattndlin pvalxmattndlin' pvalxmattndlin''
1      0.47422        0.43966         0.07732          0.05216
> # get the second element of the list for each model/constituent combination
> # the survival regression summary for each model/constituent combination
> myfitRCS[[2]]
[[1]]

Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
              Value Std. Error      z       p
xmatintcpt   -2.2851     0.1392 -16.42 < 2e-16
```

```
xmatwavest    0.5849     0.1137   5.14 2.7e-07
xmattndlin    0.0545     0.0705   0.77   0.440
xmattndlin'  -0.4226     0.2392  -1.77   0.077
xmattndlin''  1.2513     0.6444   1.94   0.052
xmatflowa30  -0.0839     0.3118  -0.27   0.788
xmatflowa1    2.0742     0.4742   4.37 1.2e-05
Log(scale)   -1.3204     0.1201 -10.99 < 2e-16

Scale= 0.267

Gaussian distribution
Loglik(model)= -30.3   Loglik(intercept only)= -60.9
        Chisq= 61.04 on 6 degrees of freedom, p= 2.8e-11
Number of Newton-Raphson Iterations: 6
n= 115
```

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

```
   dectime R04035 P04035
1 1996.034      <  0.004
2 1996.117      <  0.004
3 1996.201         0.005
4 1996.242      <  0.004
5 1996.273         0.007
6 1996.311      <  0.004
```

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

```
    dectime      P04035
30 1995.831 0.001729834
31 1995.833 0.001760735
32 1995.835 0.001785889
33 1995.837 0.001821616
34 1995.840 0.001868708
35 1995.843 0.001819211
```

```
> # get the summary of predicted concentrations
> myfitRCS[[5]]
```

```
  analysis pname predMeanConc predQ10 predQ25 predQ50 predQ75
1 myfitRCS 04035      0.00269 0.00108 0.00134 0.00205 0.00332
  predQ90
1 0.00542
```

```
> # get summary of trend results
> myfitRCS[[6]]
```

```
  pname mclass baseConc endConc rcsctndPpor
1 04035      2   0.0035  0.0033     -4.2617
  rcsctndOrigPORPercentBase pvalrcstnd ctndlklhd
1                     -1e-04         NA        NA
```

> attributes(myfitRCS[[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"
```

> myfitRCS[[2]][[1]]$n

```
[1] 115
```

> myfitRCS[[2]][[1]]$table

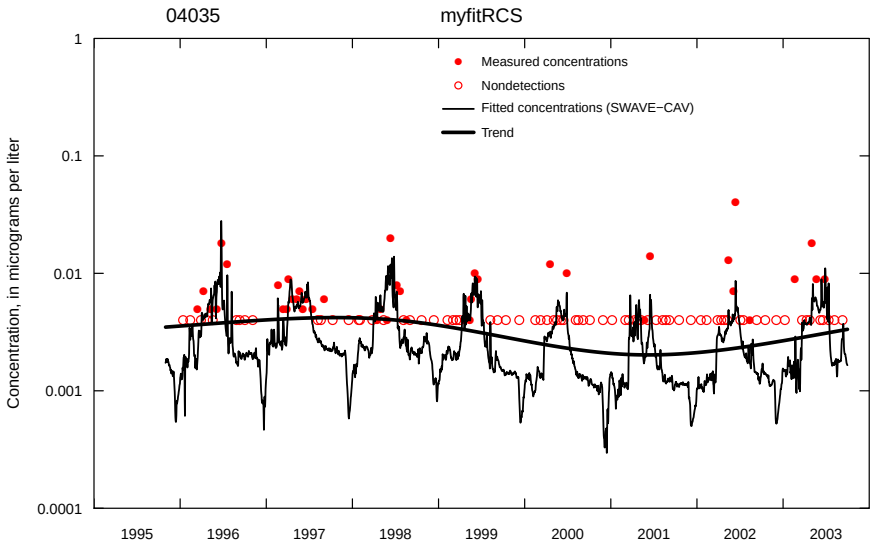
```
                  Value Std. Error           z            p
xmatintcpt   -2.28508179 0.13916276 -16.4202108 1.370896e-60
xmatwavest    0.58485595 0.11367880   5.1448112 2.677900e-07
xmattndlin    0.05451475 0.07054532   0.7727622 4.396631e-01
xmattndlin'  -0.42261789 0.23924903  -1.7664352 7.732287e-02
xmattndlin''  1.25130280 0.64439682   1.9418203 5.215886e-02
xmatflowa30  -0.08385015 0.31179820  -0.2689244 7.879878e-01
xmatflowa1    2.07416526 0.47421631   4.3738801 1.220575e-05
Log(scale)   -1.32043054 0.12012033 -10.9925647 4.149640e-28
```

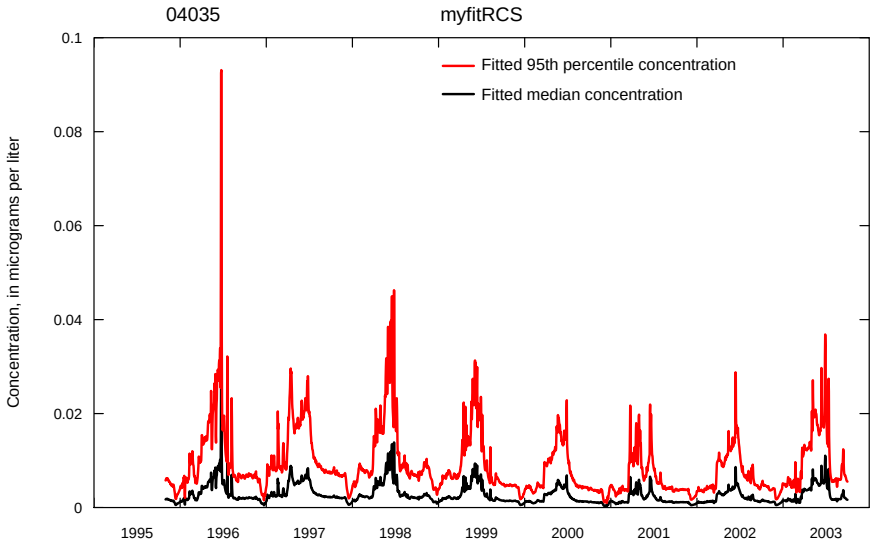
The text output file is inserted here as an example.

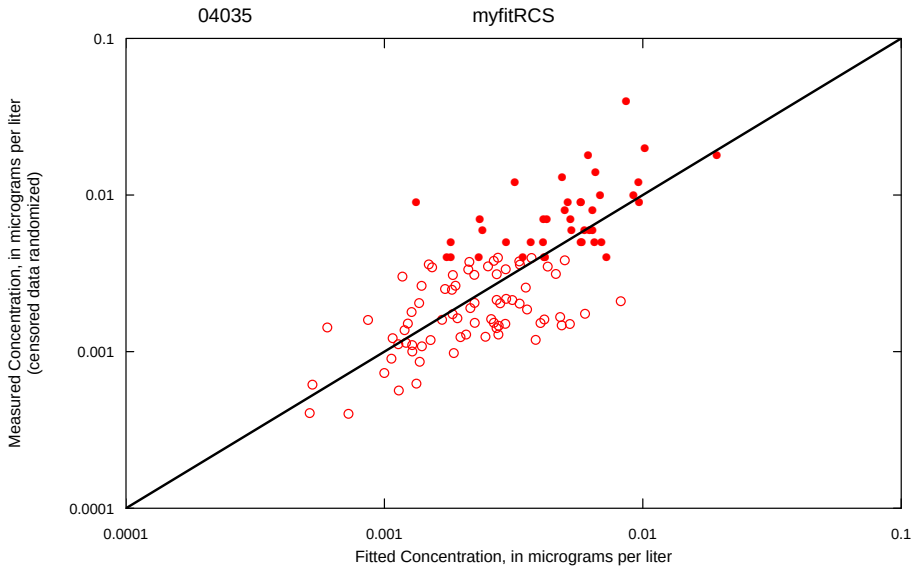
```
Monday 27 Jul 2020 16:00:52 PM CDT
R version 3.5.1 (2018-07-02)
seawaveQ version 2.0.0
x86_64-apple-darwin15.6.0 (64-bit)

Final model survreg results for 04035
Call:
survreg(formula = Surv(time = clogtmp, time2 = indcen, type = "left") ~
    xmat - 1, dist = "gaussian", control = list(maxiter = 100))
              Value Std. Error      z       p
xmatintcpt   -2.2851     0.1392 -16.42 < 2e-16
xmatwavest    0.5849     0.1137   5.14 2.7e-07
```

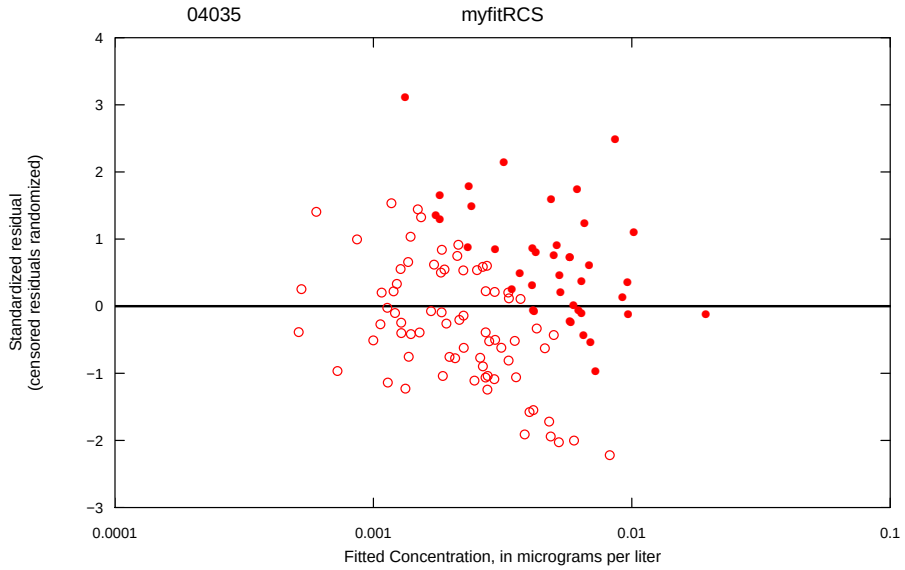
```
xmattndlin    0.0545     0.0705   0.77   0.440
xmattndlin'  -0.4226     0.2392  -1.77   0.077
xmattndlin''  1.2513     0.6444   1.94   0.052
xmatflowa30  -0.0839     0.3118  -0.27   0.788
xmatflowa1    2.0742     0.4742   4.37 1.2e-05
Log(scale)   -1.3204     0.1201 -10.99 < 2e-16

Scale= 0.267

Gaussian distribution
Loglik(model)= -30.3   Loglik(intercept only)= -60.9
Chisq= 61.04 on 6 degrees of freedom, p= 2.8e-11
Number of Newton-Raphson Iterations: 6
n= 115

Generalized r-squared is:  0.41
AIC (Akaike's An Information Criterion) is:  76.7
BIC (Bayesian Information Criterion) is:  98.66
Model class is 2
Pulse input function is 3
Half life is 1
Seasonal value of the maximum concentration is 0.48.
```

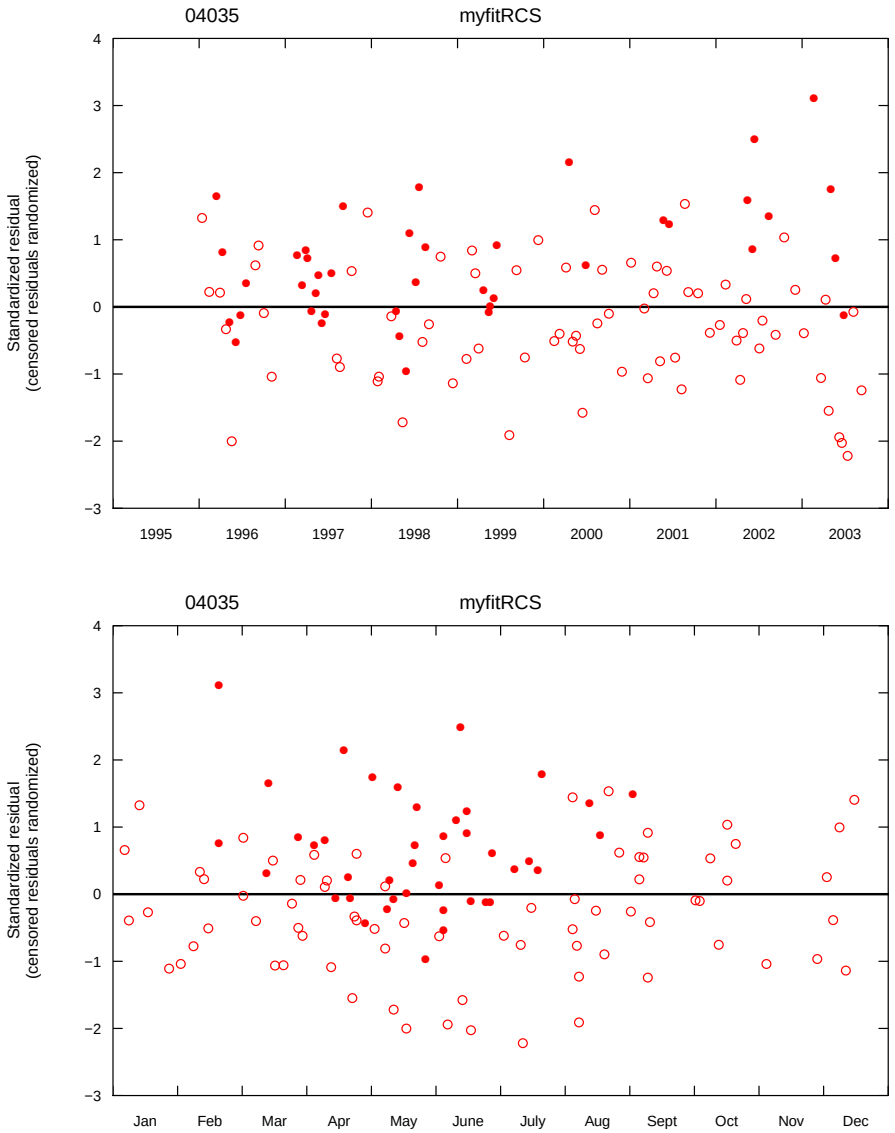
The plots written to a pdf file are included below.


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

8 Bootstrapping for Significance with Restricted Cubic Splines

The trend component of a model using restricted cubic splines is more complex than with previous implementations of the model. Because the model now as curvilinear components, the trend can be expressed a change in concentration from the beginning of the curvilinear trend line to the end of the curvilinear trend line (see Ryberg and York, 2020, for more details). In some cases, where the addition of restricted cubic splines has little change to the model, this trend may be the same as that of the linear model; however, sometimes the trend can be dramatically different because restricted cubic splines can respond to abrupt or gradual changes in use or regulation that cannot be modeled linearly. In addition, the determination of statistical significance is different for the model with restricted cubic splines. In order to determine if the trends in pesticide concentration calculated using restricted cubic splines are substantially different from a pattern produced by chance alone, we added functionality to perform a bootstrap analysis. Bootstrapping creates a distribution by resampling the dataset. Normal bootstrapping eliminates all trends and produces trend-free data, while block bootstrapping preservers any autocorrelation detected in the sample if the sample is serially dependent (Önöz and Bayazit, 2012). To ensure that block bootstrapping preserves autocorrelation, blocks are often overlapped and are wrapped around from the end to the start of the series (Canty, 2002). The bootstrap analysis in seawaveQ uses a variation of block bootstrapping and an attained significance level similar to the method used in Hirsch and Ryberg (2012). However, Hirsch and Ryberg used a random length block with block length geometrically distributed with a mean length of 20 years to replicate hydrologic conditions that might have climatic persistence of variable length. In our case, we did not want to randomize the seasonal patterns in concentration trends, because we would then be testing the significance of the seasonal wave in the model rather than the trend. Therefore, our block units are entire years. The significance calculation is based on an attained level of significance method described in Hirsch and Ryberg (2012). For each bootstrap replicate, the resampled time series of pesticide concentrations is regressed against the variables in the SEAWAVE-Q model. For iteration j, mj is the estimate of the trend. The mjs form a distribution of possible trend values across the bootstrap resampled dataset. The test statistic is M, is the estimated trend with the observations from the original data (data in their original order). The null hypothesis is that the expected value of M is zero (no trend). Using 1,000 iterations of the bootstrap, we compute the attained, two-sided significance level. The p-value is the fraction of the iterations in which $|mj|$ is greater than or equal to $|M|$. If few of the trends from the randomized data (the mjs) are larger than the original trend value, the trend is statistically significant, depending on the analyst's choice of significance level. The seawaveQ results report the attained p-value so that one may choose to compare it to any significance level.

The bootstrap process is too time consuming to include in an R vignette. However, the function call would be something like the following:

```
> # myfitRCSTrend <- fitswavecav(cdat = modMoRivOmaha, cavdat = cqwMoRivOmaha,
> #                              tanm = "myfitRCSTrend",
> #                              pnames = c("04035", "04037", "04041"),
> #                              yrstart = 1995, yrend = 2003, tndbeg = 1995,
> #                              tndend = 2003, iwcav = c("flowa30", "flowa1"),
> #                              dcol = "dates", qwcols = c("R", "P"), mclass = 2,
> #                              numk = 4, bootRCS = TRUE, nboot = 1000)
```

9 References Cited

Canty, A.J., 2002, Resampling methods in R—The boot package: R News, v. 2, no. 3, p. 2–7.

Croxford, R., 2016, Restricted cubic spline regression—A brief introduction: SAS Institute Inc., Paper 5621.

Harrell, F.E., Jr., 2010, Regression modeling strategies—With applications to linear models, logistic regression, and survival analysis: New York, Springer-Verlag, 568 p.

Harrell, F.E., Jr., 2016, rms: Regression modeling strategies; R package version 5.0-1: accessed August 14, 2018, at https://CRAN.R-project.org/package=rms.

Helsel, D.R., 2005, Nondetects and data analysis: New York, John Wiley and Sons, 250 p.

Hirsch, R.M., and Ryberg, K.R., 2012, Has the magnitude of floods across the USA changed with CO2 levels?: Hydrological Sciences Journal, v. 57, no. 1, p. 1–9, https://doi.org/10.1080/02626667.2011.621895.

Korn, E.L., and Graubard, B.I., 1999, Appendix C—Restricted cubic regression splines *in* Korn, E.L, and Graubard, B.I., Analysis of health surveys: New York, John Wiley and Sons, p. 345–346, https://doi.org/10.1002/9781118032619.

Lee, Lopaka, 2017, NADA—Nondetects and data analysis for environmental data: R package version 1.6-1, https://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.]

Önöz, B., and Bayazit, M., 2012, Block bootstrap for Mann–Kendall trend test of serially depend data: Hydrological Processes v. 26, no. 23, p. 3552–3560, https://doi.org/10.1002/hyp.8438.

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 York, B.C., 2020, 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 2020–1082, 25 p., with 4 appendixes, https://doi.org/10.3133/ofr20201082.

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 Inves-

tigations Report 2010–5139; 101 p., accessed May 1, 2012, at http://pubs.usgs.gov/sir/2010/5139/.

Stone, C.J., 1986, Comment—Generalized additive models: Statistical Science, v. 1, no. 3, p. 312–314.

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. 9096–9102, 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, p. 1308–1324, accessed May 1, 2012, at http://dx.doi.org/10.1111/j.1752-1688.2008.00225.x.