Application 3: Analysis of an Censored Constituent using a Seasonal Model

Dave Lorenz

July 20, 2015

This application illustrates the "7-parameter model," a predefined model that has been shown to perform well for loading analyses of constituents in large (> 100 square miles) watersheds (Cohn and others, 1992). In addition, the application illustrates the use of LOADEST when portions of the calibration data set are subject to censoring.

As in the previous example, a constituent with a seasonal loading pattern is considered here. In this case, constituent concentrations are assumed to vary in a continuous manner, as opposed to the abrupt changes considered in Application 2. Several of the predefined models (models 4 and 6–9; Section 3.2.2, table 7) use a first-order Fourier series (sine and cosine terms) to consider seasonality. In this application, the 7-parameter model (model 9) is developed for nutrient loading on the Potomac River. The 7-parameter model is given by:

$$\log(Load_i) = \alpha_0 + \alpha_1 lnQ_i + \alpha_2 lnQ_i^2 + \alpha_3 cT_i + \alpha_4 cT_i^2 + \alpha_5 \sin(2\pi dT_i) + \alpha_6 \cos(2\pi dT_i) + \epsilon_i, \quad (1)$$

where lnQ_i is the centered log of flow, cT_i is centered decimal time, and dT_i is the decimal time for observation i. Within the model, explanatory variables one and two account for the dependence on flow, explanatory variables three and four account for the time trend, and explanatory variables five and six are a first-order Fourier series to account for seasonal variability.

The load regression model for orthophosphate data collected near USGS gaging station 01646580 on the Potomac River uses equation 1. The retrieved dataset includes 237 observations of concentrations collected from 2002 to 2010; many of the observations are below the laboratory detection limit, resulting in a censored data set. The flow data will be from USGS gaging station 01646502, located just upstream from the water-quality gage.

1 Retrieve and Build the Datasets

Instead of relying on a packaged dataset, this example will retrieve data from NWISweb. You must be connected to the Internet in order to replicate the results in this example.

The first step is to retrieve the water-quality and flow data. The water-quality data are retrieved using the `importNWISqw` function, which requires the station identifier and the parameter code. It also accepts beginning and ending dates. The flow data are retrieved using the `readNWIS` function, which requires only the station identifier and also accepts beginning and ending dates as well as other arguments not used. The `renCol` function simply renames the flow column so that it is more readable by humans.

```
> # Load the rloadest package, which requires the USGSwsQW and
> # other packages that contain the necessary functions
> library(dataRetrieval)
> library(rloadest)
> app3.qw <- importNWISqw("01646580", params="00660",
+     begin.date="2001-10-01", end.date="2010-09-30")
> app3.flow <- renameNWISColumns(readNWISdv("01646502", "00060",
+     startDate="2001-10-01", endDate="2010-09-30"))
```

The second step is to merge the flow data with the water-quality data to produce a calibration dataset. The function `mergeQ` extracts the flow data from the flow dataset and merges the daily flow with the sample date in the water-quality dataset. For this analysis, we assume that a sample on any given day represents a valid estimate of the mean daily concentration. It requires that the names of the dates column match between the two datasets; the column `sample_dt` in the water-quality data set is renamed to `Date` to match the date column in the flow dataset. A further requirement of `mergeQ` is that there are no replicate samples taken on the same day. In general, the concentration values with a day agree very well, for this example, simply delete the duplicated days. For other cases, it may be better to compute a mean-daily concentration.

```
> # There are duplicated samples in this dataset. Print them
> subset(app3.qw, sample_dt %in%
+   app3.qw[duplicated(app3.qw$sample_dt), "sample_dt"])

     site_no  sample_dt sample_tm tzone_cd medium_cd sample_end_dt sample_end_tm OrthoPhosphate.PO4
67  01646580 2004-07-08     09:45      EST        WS                                         E0.012
68  01646580 2004-07-08     09:50      EST        WS                                         E0.009
105 01646580 2006-03-09     10:30      EST        WS                                         <0.037
106 01646580 2006-03-09     11:30      EST        WS                                         <0.037
153 01646580 2008-02-05     10:15      EST        WS                                         E0.017
154 01646580 2008-02-05     10:20      EST        WS                                         E0.018
156 01646580 2008-03-04     10:15      EST        WS                                         <0.018
157 01646580 2008-03-04     10:20      EST        WS                                         <0.018
159 01646580 2008-04-02     10:15      EST        WS                                          0.021
160 01646580 2008-04-02     10:20      EST        WS                                         E0.015
162 01646580 2008-05-06     09:15      EST        WS                                          0.071
163 01646580 2008-05-06     09:20      EST        WS                                          0.070
165 01646580 2008-05-15     08:45      EST        WS                                          0.070
166 01646580 2008-05-15     08:50      EST        WS                                          0.075
168 01646580 2008-06-04     08:45      EST        WS                                          0.043
169 01646580 2008-06-04     08:50      EST        WS                                          0.040
```

```
170 01646580 2008-06-10     09:45      EST        WS                                          0.091
171 01646580 2008-06-10     09:50      EST        WS                                          0.089
173 01646580 2008-07-01     10:15      EST        WS                                          0.047
174 01646580 2008-07-01     10:20      EST        WS                                          0.052
177 01646580 2008-08-04     09:15      EST        WS                                          0.078
178 01646580 2008-08-04     09:20      EST        WS                                          0.083
183 01646580 2008-10-01     09:15      EST        WS                                          0.210
184 01646580 2008-10-01     09:20      EST        WS                                          0.220
186 01646580 2008-12-03     10:15      EST        WS                                         <0.025
187 01646580 2008-12-03     10:20      EST        WS                                         <0.025
189 01646580 2009-01-12     10:45      EST        WS                                          0.044
190 01646580 2009-01-12     10:50      EST        WS                                          0.055
191 01646580 2009-03-03     09:45      EST        WS                                         <0.025
192 01646580 2009-03-03     09:50      EST        WS                                         <0.025
193 01646580 2009-04-01     09:15      EST        WS                                         E0.020
194 01646580 2009-04-01     09:20      EST        WS                                         E0.013
196 01646580 2009-05-05     09:45      EST        WS                                          0.110
197 01646580 2009-05-05     09:50      EST        WS                                          0.098
199 01646580 2009-05-14     09:45      EST        WS                                          0.091
200 01646580 2009-05-14     09:50      EST        WS                                          0.095
202 01646580 2009-06-03     09:45      EST        WS                                          0.074
203 01646580 2009-06-03     09:50      EST        WS                                          0.073
204 01646580 2009-06-15     08:45      EST        WS                                         E0.016
205 01646580 2009-06-15     08:50      EST        WS                                         E0.017
207 01646580 2009-07-01     09:15      EST        WS                                          0.035
208 01646580 2009-07-01     09:20      EST        WS                                          0.035
210 01646580 2009-08-11     10:45      EST        WS                                          0.120
211 01646580 2009-08-11     10:50      EST        WS                                          0.120
215 01646580 2009-10-06     09:45      EST        WS                                         <0.025
216 01646580 2009-10-06     09:50      EST        WS                                         <0.025
219 01646580 2010-03-04     10:15      EST        WS                                          0.036
220 01646580 2010-03-04     10:25      EST        WS                                          0.035
221 01646580 2010-04-06     09:15      EST        WS                                          0.028
222 01646580 2010-04-06     09:25      EST        WS                                          0.030
223 01646580 2010-05-04     09:30      EST        WS                                         <0.025
224 01646580 2010-05-04     09:40      EST        WS                                         <0.025
225 01646580 2010-05-11     09:15      EST        WS                                         <0.025
226 01646580 2010-05-11     09:25      EST        WS                                         <0.025
```

```
> # Remove the duplicates
> app3.qw <- subset(app3.qw, !duplicated(sample_dt))
> # Now change the date column name and merge
> names(app3.qw)[2] <- "Date"
> # Supress the plot in this merge
> app3.calib <- mergeQ(app3.qw, FLOW="Flow", DATES="Date",
+                      Qdata=app3.flow, Plot=FALSE)
```

2 Build the Model

The loadReg function is used to build the rating-curve model for constituent load estimation. The basic form of the call to loadReg is similar to the call to lm in that it requires a formula and data source. The response variable in the formula is the constituent concentration, which is converted to load per day (flux) based on the units of concentration and the units of flow. The conc.units, flow.units, and load.units arguments to loadReg define the conversion. For these data, the concentration units (conc.units) are ”mg/L” (as orthophosphate) and are known within the column so do not need to be specified, the flow units are ”cfs” (the default), and the load units for the model are ”kilograms.” Two additional pieces of information are required for loadReg—the names of the flow column and the dates column. A final option, the station identifier, can also be specified.

```
> # Create and print the load model.
> app3.lr <- loadReg(OrthoPhosphate.PO4 ~ model(9), data = app3.calib,
+   flow = "Flow", dates = "Date",
+   station="Potomac River at Chain Bridge, at Washington, DC")
> print(app3.lr)

*** Load Estimation ***

Station: Potomac River at Chain Bridge, at Washington, DC
Constituent: OrthoPhosphate.PO4

           Number of Observations: 210
Number of Uncensored Observations: 179
           Center of Decimal Time: 2006.299
                  Center of ln(Q): 9.3227
                 Period of record: 2001-10-30 to 2010-09-02

Selected Load Model:
--------------------

OrthoPhosphate.PO4 ~ model(9)

Model coefficients:
             Estimate Std. Error z-score p-value
(Intercept)  7.049643    0.08472 83.2130  0.0000
lnQ          1.462419    0.06069 24.0961  0.0000
lnQ2        -0.007454    0.03762 -0.1981  0.8233
DECTIME     -0.082608    0.02164 -3.8165  0.0001
DECTIME2     0.014610    0.01017  1.4364  0.1403
sin.DECTIME -0.636680    0.09101 -6.9959  0.0000
cos.DECTIME -0.296216    0.08009 -3.6985  0.0002

AMLE Regression Statistics
Residual variance: 0.5126
Generalized R-squared: 76.89 percent
G-squared: 307.6 on 6 degrees of freedom
P-value: <0.0001
Prob. Plot Corr. Coeff. (PPCC):
  r = 0.99
  p-value = 0.0117
Serial Correlation of Residuals: 0.3576

Variance Inflation Factors:
              VIF
lnQ         1.620
lnQ2        1.139
DECTIME     1.045
DECTIME2    1.160
sin.DECTIME 1.539
cos.DECTIME 1.091

Comparison of Observed and Estimated Loads
------------------------------------------
      Summary Stats: Loads in kg/d
---------------------------------------------
      Min 25%  50%  75%   90%   95%   Max
Est 137.0 382 1260 3100 10400 17500 96900
Obs  50.8 262  868 3310  9030 17300 66100

Bias Diagnostics
----------------
 Bp: 19.08 percent
PLR: 1.191
  E: 0.5439
```

A few details from the printed report deserve mention–the second order flow and decimal time terms have p-values that are greater than 0.05 and may not be necessary; the p-value of the PPCC test is less that 0.05, which suggests a lack of normality; the serial correlation of the residuals is 0.3576, which is quite large; and Bp is relatively large at 19.08.

3 Diagnostic Plots

Figure 1 shows the AMLE 1:1 line as a dashed line and the solid line is a LOWESS smooth curve. The LOWESS curve indicates a good fit.

```
> # setSweave is required for the vignette.
> setSweave("app3_01", 5, 5)
> plot(app3.lr, which=1, set.up=FALSE)
> graphics.off()
```

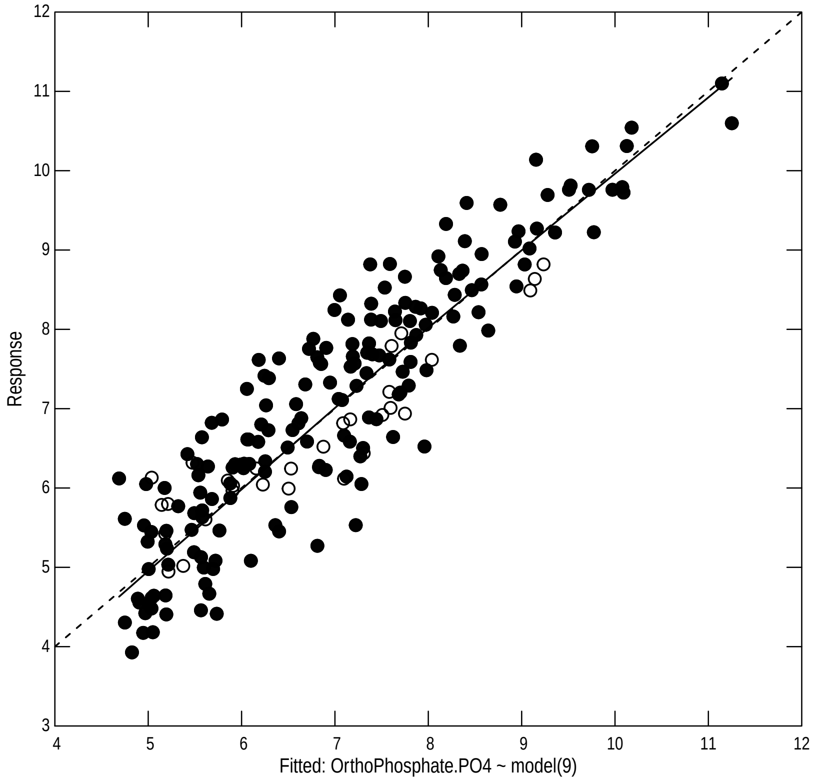

Figure 1. The rating-curve regression model.

Figure 2 is a scale-location (S-L) graph that is a useful graph for assessing heteroscedasticity of the residuals. The horizontal dashed line is the expected value of the square root of the absolute value of the residuals and the solid line is the LOWESS smooth. In this case, only 1 of the seven largest residuals is above the expected value line, which suggests in decreasing variance as the estimated load increases.

```
> # setSweave is required for the vignette.
> setSweave("app3_02", 5, 5)
> plot(app3.lr, which=3, set.up=FALSE)
> graphics.off()
```

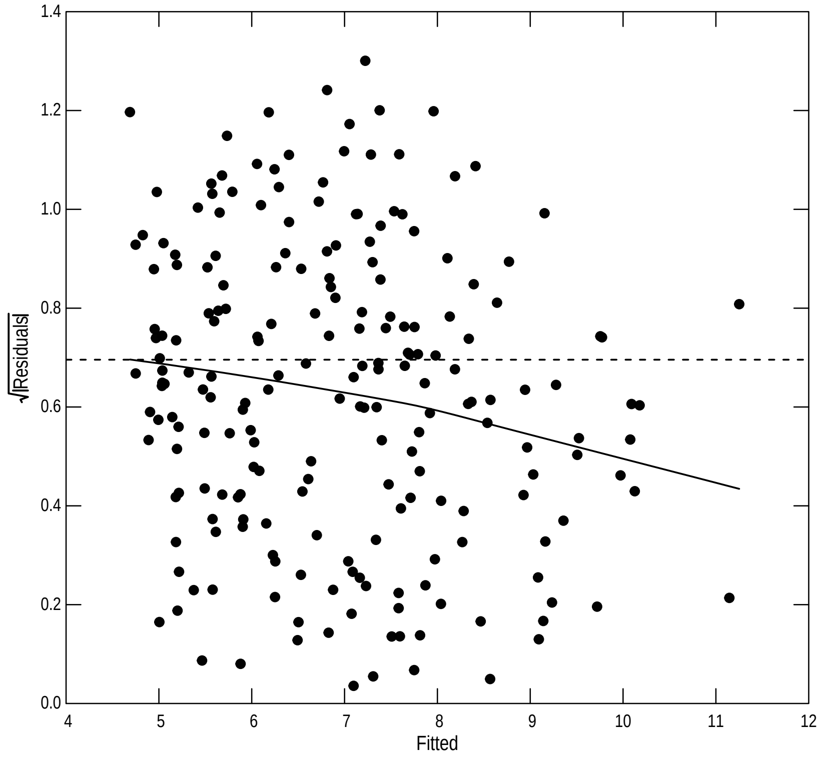

Figure 2. The scale-location graph for the regression model.

The correlogram in figure 3 is a adaptation of the correlogram from time-series analysis, which deals with regular samples. The horizontal dashed line is the zero value and the solid line is a kernel smooth rather than a LOWESS line. The kernel smooth gives a better fit in this case. The solid line should be very close to the horizontal line. In this case, there is a suggestion of a long-term lack of fit because the solid line is above the horizontal line for a 1-year lag.

```
> # setSweave is required for the vignette.
> setSweave("app3_03", 5, 5)
> plot(app3.lr, which=4, set.up=FALSE)
> graphics.off()
```

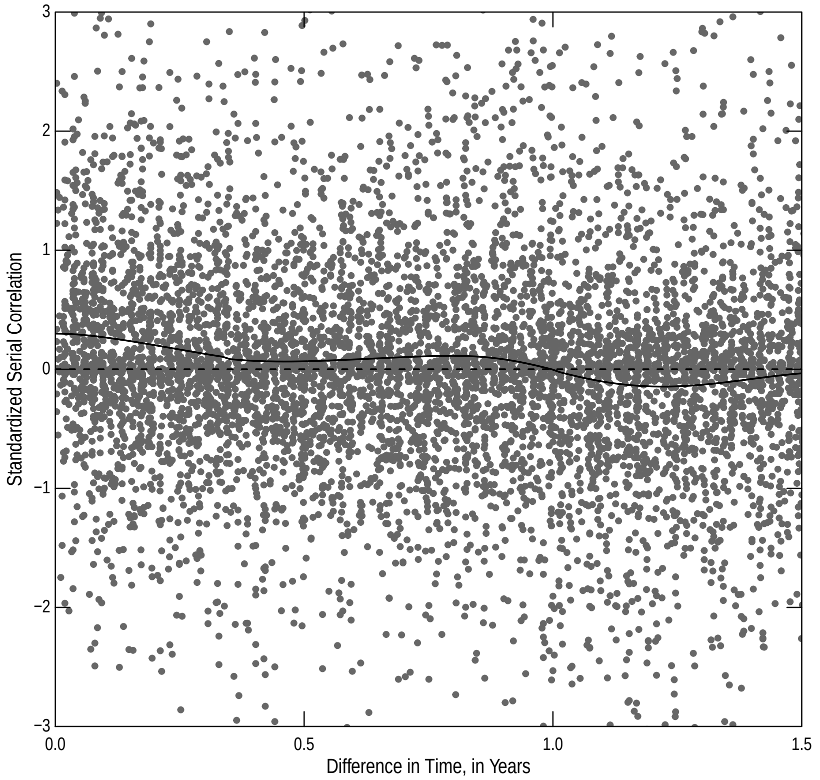

Figure 3. The correlogram from the regression model.

Figure 4 shows the q-normal plot of the residuals. The visual appearance of figure 4 confirms the results of the PPCC test in the printed output—the largest residuals trail off the line.

```
> # setSweave is required for the vignette.
> setSweave("app3_04", 5, 5)
> plot(app3.lr, which=5, set.up=FALSE)
> graphics.off()
```

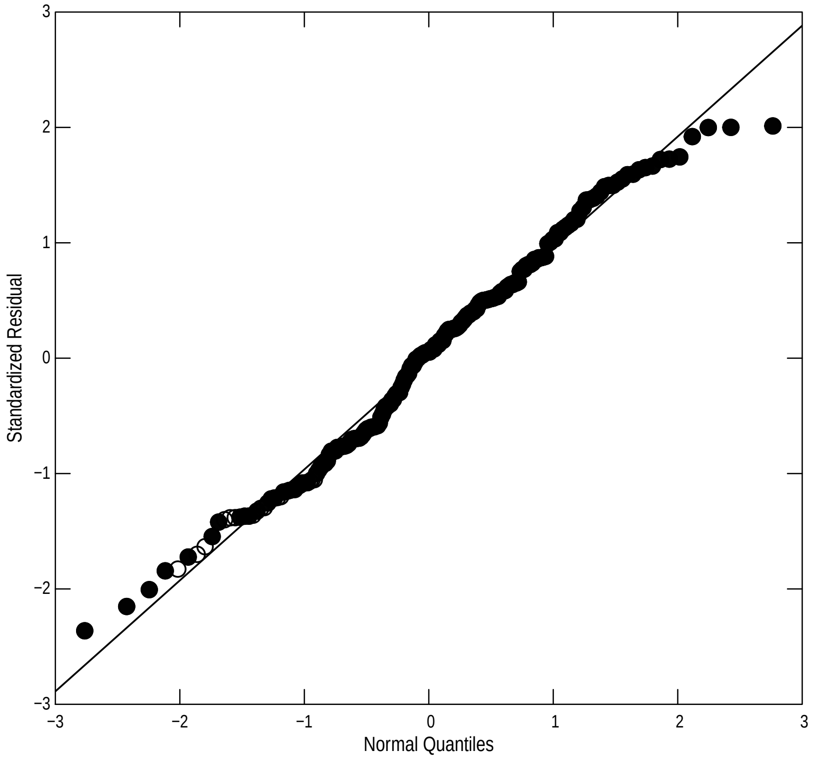

Figure 4. The Q-normal plot of the residuals.

Figure 5 shows the partial residual plot for decimal time (DECTIME). This one was selected because of the long-term lack of fit over time suggested by figure 3. The dashed line is the linear fit and the solid line is the LOWESS smooth. In this case, the LOWESS smooth does follow the fitted line, but there is a distinct pattern in the left part of the graph–most of the residuals are above the line up to a DECTIME value of about -3 and then most residuals are below the line to about -1, after which the residuals are fairly well behaved.

```
> # setSweave is required for the vignette.
> setSweave("app3_05", 5, 5)
> plot(app3.lr, which="DECTIME", set.up=FALSE)
> graphics.off()
```

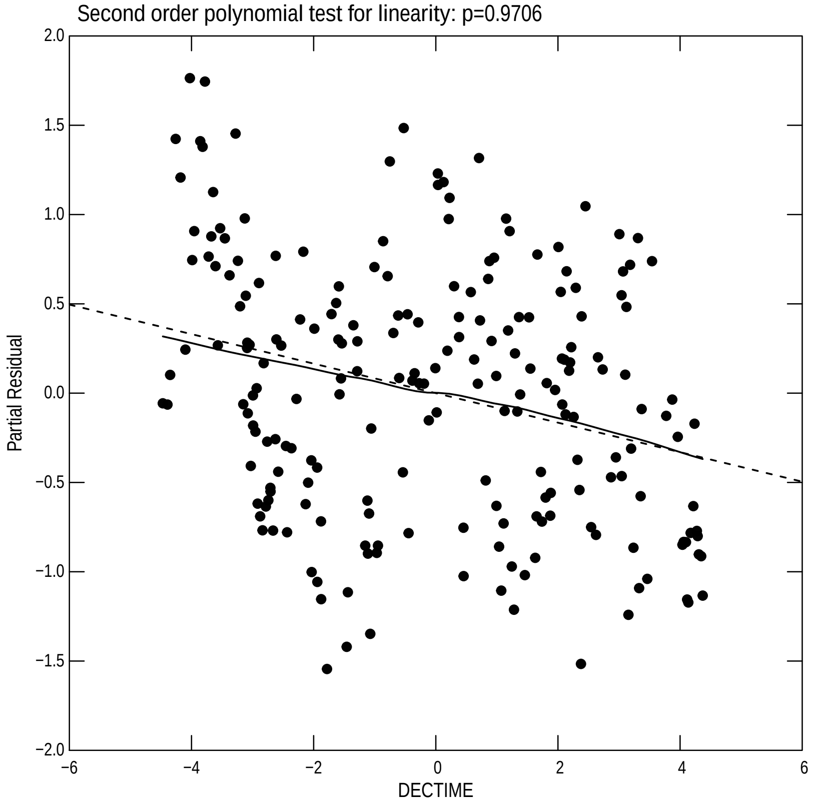

Figure 5. The partial residual plot for decimal time.

4 Further Diagnostics

Figure 5 suggested a distinct pattern in the residuals in the early part of the record. Figure 6 replots the residuals on a date axis, so that it will be easier to relate to the date. The second call to `refLine` adds vertical lines at the water years.

```
> # setSweave is required for the vignette.
> setSweave("app3_06", 5, 5)
> timePlot(app3.calib$Date, residuals(app3.lr),
+          Plot=list(what="points"), ytitle="Residuals")
> refLine(horizontal=0)
> refLine(vertical=as.Date("2001-10-01") + years(0:9))
> graphics.off()
```

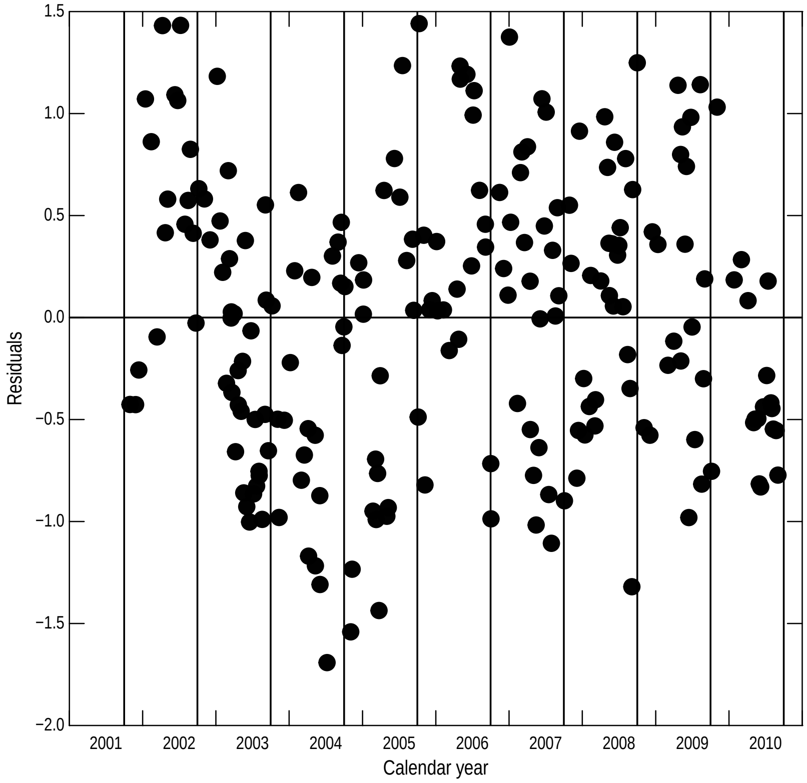

Figure 6. Model residuals by date.

The residuals for water-year 2002 are mostly greater than 0; those for watery-year 2003 are trending down and those for water-year 2004 are mostly less than 0. This raises the question about whether more persistent flow patterns affect the relation between flow and concentration. The code immediately below computes the average (first line) and water-year average (second line) flow. The pattern of water-year average flows closely matches the pattern of the residuals.

```
> mean(app3.flow$Flow)
```

```
[1] 12656.48
```

```
> with(app3.flow, tapply(Flow, waterYear(Date), mean))
```

```
     2002      2003      2004      2005      2006      2007      2008      2009      2010 
 4661.712 23446.740 18394.617 12782.082  9337.562 11086.959 11035.164  9488.548 13663.671 
```

5 Modeling Flow Anomalies

Vecchia and others (2008) describe an approach for breaking down stream flow into what they call anomalies–long- to intermediate-term deviations from average flow and the residual high-frequency variation or daily residuals. That approach can be very useful in cases such as this where there is a strong relation between flow and concentration, but relatively persistent patterns of flow are not captured.

The first step in modeling flow anomalies required retrieving data for a longer period of time. We'll retrieve data from two years prior to the start of the sampling record that we are working with. The additional two years of record were selected because the first one year would be all missing values and one additional year to establish a pattern going into 2001. Then we'll construct a single 1-year anomaly, which seems to make sense from figure 6. This 6-parameter anomaly model is given by:

$$\log(Load_i) = \alpha_0 + \alpha_1 A1yr_i + \alpha_2 HFV_i + \alpha_3 dT_i + \alpha_4 \sin(2\pi dT_i) + \alpha_5 \cos(2\pi dT_i) + \epsilon_i, \qquad (2)$$

where $A1yr_i$ is the 1-year anomaly of the log of flow, HFV_i is remaining high-frequency variation in the log of flow, and dT_i is the decimal time for observation i.

```
> app3.anom <- renameNWISColumns(readNWISdv("01646502", "00060",
+   startDate="1999-10-01", endDate="2010-09-30"))
> app3.anom <- cbind(app3.anom, anomalies(log(app3.anom$Flow),
+   a1yr=365))
> # The head would show missing values for a1yr and HFV
> tail(app3.anom)
```

```
     agency_cd  site_no       Date Flow_cd Flow        a1yr         HFV
4013      USGS 01646502 2010-09-25       A 1550 0.011527670 -1.58411007
4014      USGS 01646502 2010-09-26       A 1430 0.010749096 -1.66391198
4015      USGS 01646502 2010-09-27       A 2020 0.009923879 -1.31766370
4016      USGS 01646502 2010-09-28       A 1770 0.009010897 -1.44886868
4017      USGS 01646502 2010-09-29       A 1900 0.008259015 -1.37724246
4018      USGS 01646502 2010-09-30       A 7080 0.010839955 -0.06440338
```

The next step is to merge the flow and anomaly data with the water-quality data.

```
> # Supress the plot in this merge and overwrite app3.calib
> app3.calib <- mergeQ(app3.qw, FLOW=c("Flow", "a1yr", "HFV"),
+                      DATES="Date",
+                      Qdata=app3.anom, Plot=FALSE)
```

The final step is to construct the model. Note that flow is not a necessary part of the model because it is represented by the anomalies, linear time is represented by `dectime(Date)`, and the seasonal components by `fourier(Date)`. Note also that decimal time is not centered, but could be by using the `center` function.

```
> app3.lra <- loadReg(OrthoPhosphate.PO4 ~ a1yr + HFV + dectime(Date)
+                     + fourier(Date),
+                     data = app3.calib,
+                    flow = "Flow", dates = "Date",
+                    station="Potomac River at Chain Bridge, at Washington, DC")
> print(app3.lra)
```

```
*** Load Estimation ***

Station: Potomac River at Chain Bridge, at Washington, DC
Constituent: OrthoPhosphate.PO4

           Number of Observations: 210
Number of Uncensored Observations: 179
           Center of Decimal Time: 2006.299
                  Center of ln(Q): 9.3227
                 Period of record: 2001-10-30 to 2010-09-02

Selected Load Model:
--------------------

OrthoPhosphate.PO4 ~ a1yr + HFV + dectime(Date) + fourier(Date)

Model coefficients:
                       Estimate Std. Error z-score p-value
(Intercept)           201.10035   38.05116   5.285       0
a1yr                    0.69175    0.11572   5.978       0
HFV                     1.54017    0.05256  29.303       0
dectime(Date)          -0.09696    0.01897  -5.112       0
fourier(Date)sin(k=1)  -0.69938    0.07869  -8.888       0
fourier(Date)cos(k=1)  -0.33704    0.07024  -4.798       0

AMLE Regression Statistics
Residual variance: 0.4021
Generalized R-squared: 81.6 percent
G-squared: 355.5 on 5 degrees of freedom
P-value: <0.0001
Prob. Plot Corr. Coeff. (PPCC):
  r = 0.9924
  p-value = 0.0402
Serial Correlation of Residuals: 0.1914

Variance Inflation Factors:
                        VIF
a1yr                  1.043
HFV                   1.546
dectime(Date)         1.048
fourier(Date)sin(k=1) 1.453
fourier(Date)cos(k=1) 1.082

Comparison of Observed and Estimated Loads
------------------------------------------
      Summary Stats: Loads in kg/d
---------------------------------------------
      Min 25%  50%  75%   90%   95%   Max
Est 104.0 380 1070 2770 10600 25100 75200
Obs  50.8 262  868 3310  9030 17300 66100
```

```
Bias Diagnostics
----------------
 Bp: 22.63 percent
PLR: 1.226
  E: 0.6723
```

The residual variance is much smaller than the original model, 0.4021 rather than 0.5126. The PPCC p-value is still less than 0.05, but much closer to 0.05. The serial correlation of the residuals is much smaller than the original 0.1914 rather than 0.3576. But the Bp statistic is a bit larger 22.63 percent rather than 19.08. In spite of the larger Bp statistic, the Nash-Sutcliffe statistic (E) is larger 0.6723 rather than 0.5439. All of this suggests a better model. Review some of the diagnostic plots.

Figure 7 shows the AMLE 1:1 line as a dashed line and the solid line is a LOWESS smooth curve. The LOWESS curve indicates a good fit.

```
> # setSweave is required for the vignette.
> setSweave("app3_07", 5, 5)
> plot(app3.lra, which=1, set.up=FALSE)
> graphics.off()
```

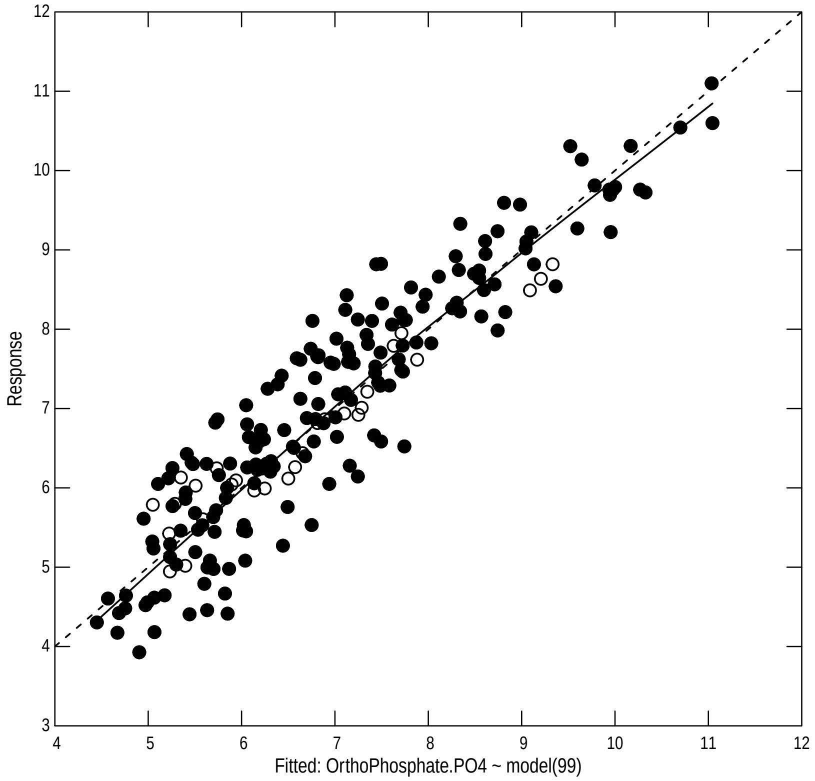

Figure 7. The revised rating-curve regression model.

Figure 8 shows the S-L graph, which indicates some decrease in variance for larger fitted values than for smaller.

```
> # setSweave is required for the vignette.
> setSweave("app3_08", 5, 5)
> plot(app3.lra, which=3, set.up=FALSE)
> graphics.off()
```

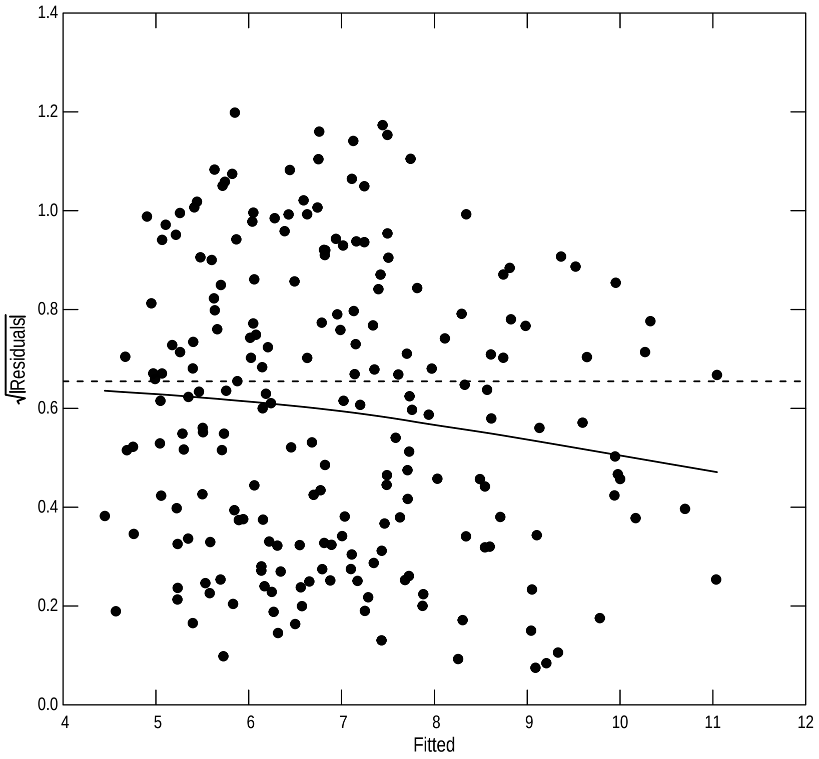

Figure 8. The scale-location graph for the revised regression model.

The correlogram in figure 9 shows more variability than one would like, but no distinct long-term or seasonal patterns.

```
> # setSweave is required for the vignette.
> setSweave("app3_09", 5, 5)
> plot(app3.lra, which=4, set.up=FALSE)
> graphics.off()
```

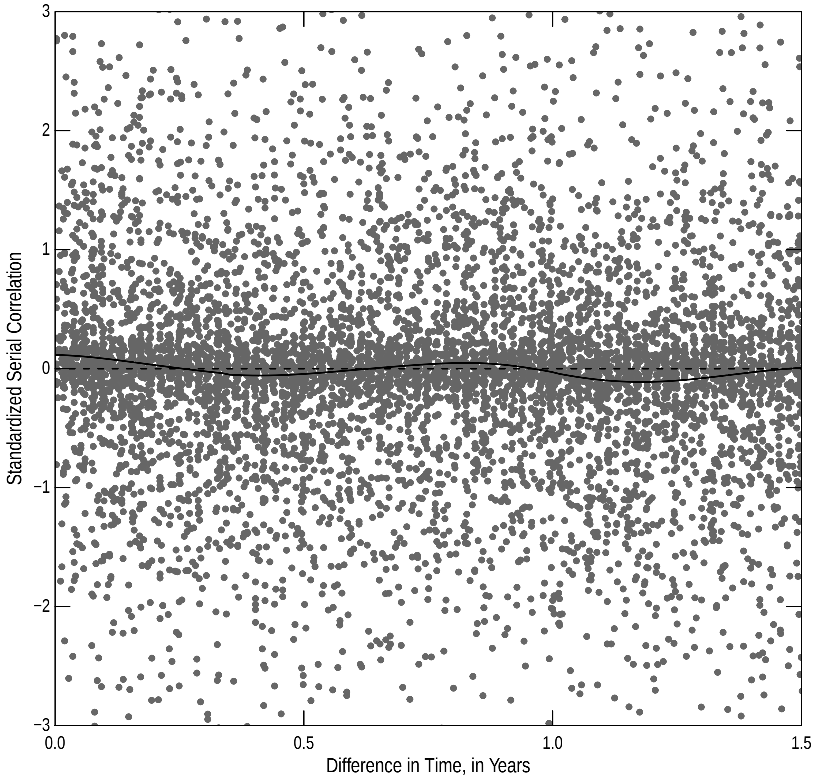

Figure 9. The correlogram from the revised regression model.

Figure 10 shows the q-normal plot of the residuals. The largest residuals trail off the line for this analysis but not quite as much as in the original model.

```
> # setSweave is required for the vignette.
> setSweave("app3_10", 5, 5)
> plot(app3.lra, which=5, set.up=FALSE)
> graphics.off()
```

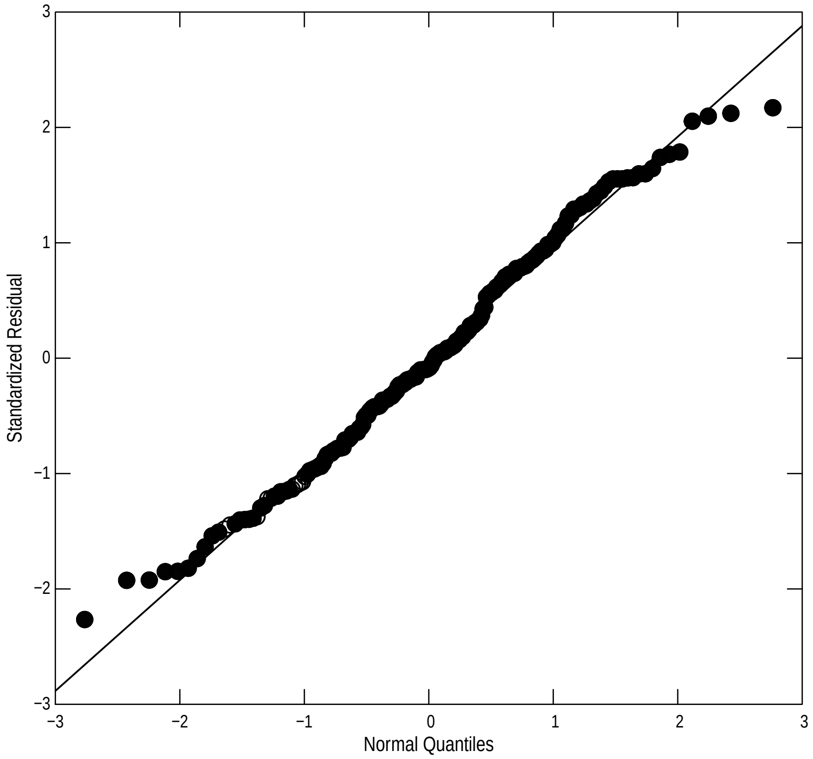

Figure 10. The Q-normal plot of the residuals from the revised model.

Figure 11 shows the partial residual plot for decimal time. This one was selected because of the long-term lack of fit over time suggested by figure 3. The dashed line is the linear fit and the solid line is the LOWESS smooth. In this case, the LOWESS smooth does follow the fitted line, but there is a distinct pattern in the left part of the graph–most of the residuals are above the line up to a DECTIME value of about -3 and then most residuals are below the line to about -1, after which the residuals are fairly well behaved.

```
> # setSweave is required for the vignette.
> setSweave("app3_11", 5, 5)
> plot(app3.lra, which="dectime(Date)", set.up=FALSE)
> graphics.off()
```

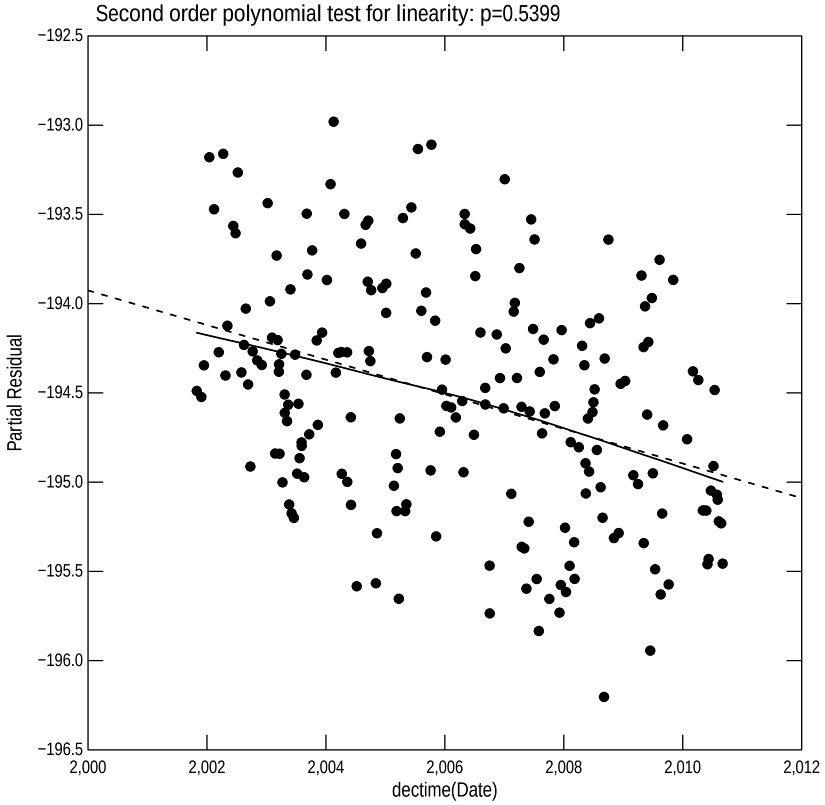

Figure 11. The partial residual plot for decimal time.

6 Load Estimates

Because we used anomalies in the regression model, we must be very careful to use the same anomalies in the estimation data. The data that were retrieved to compute the anomalies include dates outside of the calibration period, so must be subsetted to the calibration period. We'll compute load estimates for the water years 2002 through 2010

```
> app3.est <- subset(app3.anom, Date > as.Date("2001-09-30"))
> predLoad(app3.lra, newdata = app3.est, by="water year",
+          print=TRUE)

----------------------------------------------------------------------
Constituent Output File Part IIa: Estimation (test for extrapolation)
 Load Estimates for 2001-10-01 to 2010-09-30
----------------------------------------------------------------------

Streamflow Summary Statistics
-------------------------------------------

WARNING: The maximum estimation data set steamflow exceeds the maximum
calibration data set streamflow. Load estimates require extrapolation.


--------------------------------------------------------------
Constituent Output File Part IIb: Estimation (Load Estimates)
 Load Estimates for 2001-10-01 to 2010-09-30
----------------------------------------------------------------

Flux Estimates, in kg/d, using AMLE
----------------------------------------

   Period Ndays     Flux  Std.Err      SEP       L95       U95
1 WY 2002   365 1220.648 164.5140 193.7133  884.9994  1642.238
2 WY 2003   365 8037.721 837.9254 999.7867 6256.4054 10168.877
3 WY 2004   366 2671.495 315.5778 403.1849 1968.4296  3544.931
4 WY 2005   365 1591.845 124.5184 175.5853 1275.4593  1962.831
5 WY 2006   365 1570.207 101.3389 179.1237 1248.4155  1949.573
6 WY 2007   365 1537.921 111.4383 165.1147 1239.7052  1886.133
7 WY 2008   366 1786.773 163.4341 238.2895 1365.2750  2297.535
8 WY 2009   365 1214.415 108.0001 156.0877  937.2396  1547.987
9 WY 2010   365 1423.184 169.7617 220.0813 1040.5768  1901.012
```