Using EGRET Data in a rloadest Model

Dave Lorenz

July 20, 2015

This example illustrates how to set up and use data retrieved and processed for an EGRET (Hirsch and De Cicco, 2015) in a rloadest load model. EGRET includes the statistical algorithm Weighted Regressions on Time, Discharge, and Season (WRTDS) that can compute loads and concentrations. WRTDS uses locally weighted regression on linear time, linear flow (discharge), and the first-order sine and cosine terms to model constituent concentrations and fluxes over time and through the range for flow.

This example uses the processed data supplied in the EGRET package, but any data retrieved and processed by the *readNWISDaily*, *readNWISSample*, *readNWISInfo* and *mergeReport* functions in EGRET can be used. The sullied data are nitrate plus nitrite data collected in the Choptank River near Greensboro, Maryland (USGS site identifier 01491000).

```
> # Load the necessary packages and the data
> library(survival) # required for Surv
> library(rloadest)
> library(EGRET)
> # Get the QW and daily flow data
> Chop.QW <- Choptank_eList$Sample
> Chop.Q <- Choptank_eList$Daily
```

1 Compute the Initial rloadest Model

The 7-parameter model (model number 9) is a typical model for relatively long-term records, longer than about 7 years and can be a good starting point for building a good model. The water-quality data in the Sample dataset for EGRET is stored in four columns—the minimum value, maximum value, an indicator of censoring, and the average value. That format can be converted to a valid response variable for *loadReg* using either *as.mcens* or *Surv*; *Surv* is preferred because if the data are uncensored or left-censored, then the "AMLE" method is used rather that the "MLE" method, which is always used with a response variable of class "mcens."

```
> # Compute the 7-parameter model.
> Chop.lr <- loadReg(Surv(ConcLow, ConcHigh, type="interval2") ~ model(9),
+   data=Chop.QW, flow="Q", dates="Date", conc.units="mg/L",
+   flow.units="cms", station="Choptank")
```

One of the first steps in assessing the fit is to look at the diagnostic plots for the linearity of the overall fit and each explanatory variable. The overall fit (figure 1) looks linear, but there are three low outliers and a tendency to larger scatter at larger predicted values

```
> # Plot the overall fit
> setSweave("graph01", 6, 6)
> plot(Chop.lr, which=1, set.up=FALSE)
> dev.off()

null device
          1
```

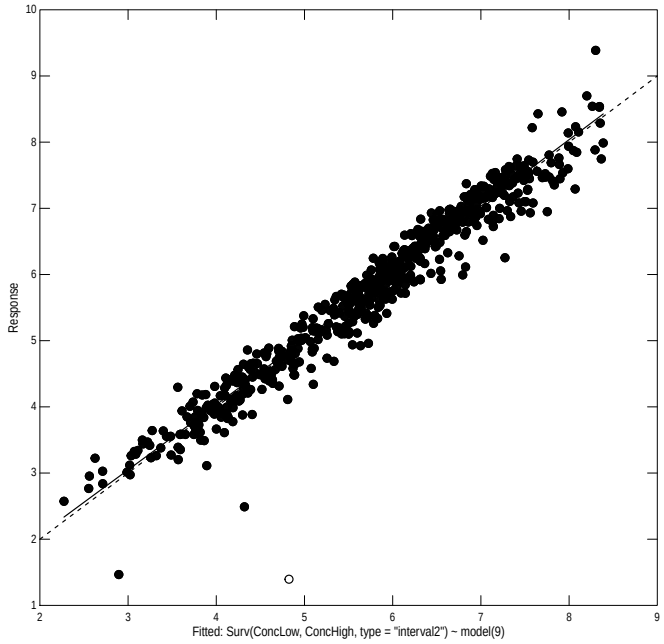

Figure 1. The overall fit.

The linearity of the explanatory variables is shown in figure 2. The partial residual plots for flow (lnQ and lnQ2) show nonlinearity in the second order (lnQ2). The partial residual plots for time (DECTIME and DECTIME2) show no nonlinearity, but the second-order term (DECTIME2) shows no trend and can therefore be removed from the model. The partial residual plots for seasonality (DECTIME and DECTIME2) show nonlinearity in both terms, suggesting the need for higher order seasonal terms.

```
> # Plot the explanatory variable fits
> setSweave("graph02", 6, 9)
> AA.lo <- setLayout(num.rows=3, num.cols=2)
> # Flow and flow squared
> setGraph(1, AA.lo)
> plot(Chop.lr, which="lnQ", set.up=FALSE)
> setGraph(2, AA.lo)
> plot(Chop.lr, which="lnQ2", set.up=FALSE)
> # Time and time squared
> setGraph(3, AA.lo)
> plot(Chop.lr, which="DECTIME", set.up=FALSE)
```

```
> setGraph(4, AA.lo)
> plot(Chop.lr, which="DECTIME2", set.up=FALSE)
> # Seasonality
> setGraph(5, AA.lo)
> plot(Chop.lr, which="sin.DECTIME", set.up=FALSE)
> setGraph(6, AA.lo)
> plot(Chop.lr, which="cos.DECTIME", set.up=FALSE)
> dev.off()

null device
          1
```

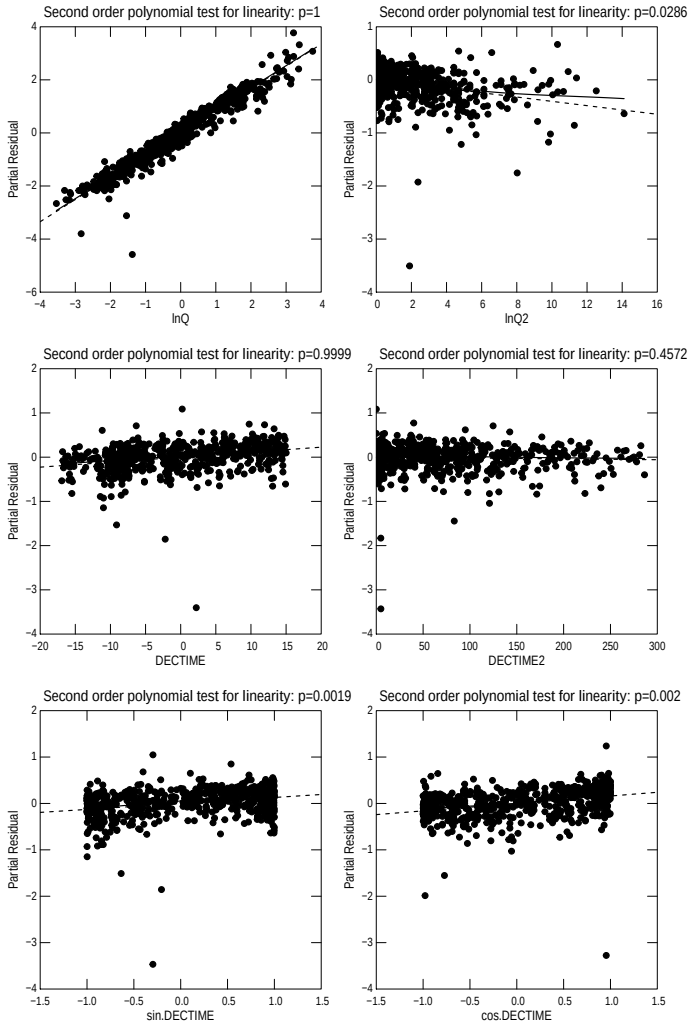

Figure 2. The linearity of the explanatory variables.

Figure 3 shows the relation between concentration and flow. The relation is not quadratic, but it appears that there is a distinct change at about 10 cubic meters per second. That relation can be modeled using piecewise linear, or segmented, terms.

```
> # Plot tconcentration and flow
> setSweave("graph03", 6, 6)
> # Use the average concentration (only one censored value)
```

```
> with(Chop.QW, xyPlot(Q, ConcAve, yaxis.log=TRUE, xaxis.log=TRUE))
> dev.off()

null device
          1
```

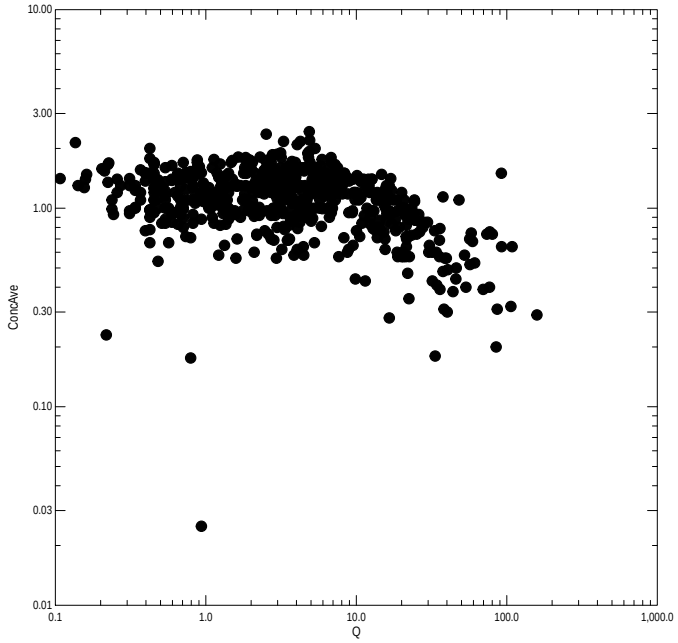

Figure 3. The relation between concentration and flow.

2 Construct the Modified rloadest Model

The *segLoadReg* can be used to build a piecewise linear model. It relies on the segmented package, which cannot model censored data to identify the breakpoints. For the first step censored values will be approximated by simple substitution; for the final model, the censored values are restored. One other quirk of *segLoadReg* is that the response term must be a variable, it cannot be constructed using *Surv* or any other function. Therefore, the breakpoint for this model will be identified using ConcAve, but the final model will be built using the censoring information.

```
> # Compute the breakpoint--the seg term must be the first term on
> # the right-hand side.
> Chop.lr <- segLoadReg(ConcAve ~ seg(LogQ, 1) + DecYear +
+     fourier(DecYear, 2),
+   data=Chop.QW, flow="Q", dates="Date", conc.units="mg/L",
+   flow.units="cms", station="Choptank")

Segmented regression results:
 AIC:
     lm      sm
405.344 334.227
Breakpoints (psi):
     Initial  Est. St.Err
psi1   1.211 1.994 0.1532

> # From the printed output, the breakpoint is 1.994 in natural log units,
> # about 7.4 cms
> # Compute and print the final model
> Chop.lr <- loadReg(Surv(ConcLow, ConcHigh, type="interval2") ~
+     segment(LogQ, 1.994) + DecYear + fourier(DecYear, 2),
+   data=Chop.QW, flow="Q", dates="Date", conc.units="mg/L",
+   flow.units="cms", station="Choptank")
> print(Chop.lr)

*** Load Estimation ***

Station: Choptank
Constituent: Surv(ConcLow, ConcHigh, type = "interval2")

           Number of Observations: 606
Number of Uncensored Observations: 605
           Center of Decimal Time: 1996.735
                  Center of ln(Q): 1.3098
```

Period of record: 1979-10-24 to 2011-09-29

Selected Load Model:

```
Surv(ConcLow, ConcHigh, type = "interval2") ~ segment(LogQ, 1.994) +
    DecYear + fourier(DecYear, 2)
```

Model coefficients:

	Estimate	Std. Error	z-score	p-value
(Intercept)	-17.886811	2.938636	-6.08677	0e+00
segment(LogQ, 1.994)X	0.948716	0.018697	50.74174	0e+00
segment(LogQ, 1.994)U1	-0.338732	0.040304	-8.40445	0e+00
segment(LogQ, 1.994)P1	0.001323	0.050014	0.02646	1e+00
DecYear	0.011303	0.001473	7.67280	0e+00
fourier(DecYear, 2)sin(k=1)	0.113551	0.021330	5.32351	0e+00
fourier(DecYear, 2)cos(k=1)	0.143342	0.018775	7.63469	0e+00
fourier(DecYear, 2)sin(k=2)	0.057079	0.017424	3.27584	1e-03
fourier(DecYear, 2)cos(k=2)	0.060241	0.017818	3.38097	7e-04

AMLE Regression Statistics
Residual variance: 0.09136
Generalized R-squared: 94.75 percent
G-squared: 1786 on 8 degrees of freedom
P-value: <0.0001
Prob. Plot Corr. Coeff. (PPCC):
  r = 0.9608
  p-value = 0
Serial Correlation of Residuals: 0.2493

Variance Inflation Factors:

	VIF
segment(LogQ, 1.994)X	4.647
segment(LogQ, 1.994)U1	3.490
segment(LogQ, 1.994)P1	3.373
DecYear	1.032
fourier(DecYear, 2)sin(k=1)	1.499
fourier(DecYear, 2)cos(k=1)	1.165
fourier(DecYear, 2)sin(k=2)	1.048
fourier(DecYear, 2)cos(k=2)	1.010

Comparison of Observed and Estimated Loads

Summary Stats: Loads in kg/d

Min	25%	50%	75%	90%	95%	Max

```
Est 12.70 122 364 920 1660 2090  4350
Obs  2.02 116 352 911 1670 1990 11900

Bias Diagnostics
----------------
 Bp: -0.4405 percent
PLR: 0.9956
  E: 0.7596
```

This segmented model has three variables— with names ending in X, U1, and P1. The coefficient for the variable ending in X is the slope for the variable less that the breakpoint, the coefficient for the variable ending in U1 is the change in slope above the breakpoint, and the coefficient for the variable ending in P1 should always be close to 0.

No partial residual plot indicates any nonlinearity. Figure 4 shows 5 selected partial residual plots.

```
> # Plot the explanatory variable fits
> setSweave("graph04", 6, 9)
> AA.lo <- setLayout(num.rows=3, num.cols=2)
> # Segmented flow
> setGraph(1, AA.lo)
> plot(Chop.lr, which="segment(LogQ, 1.994)X", set.up=FALSE)
> setGraph(2, AA.lo)
> plot(Chop.lr, which="segment(LogQ, 1.994)U1", set.up=FALSE)
> # Time
> setGraph(3, AA.lo)
> plot(Chop.lr, which="DecYear", set.up=FALSE)
> # Seasonality
> setGraph(5, AA.lo)
> plot(Chop.lr, which="fourier(DecYear, 2)sin(k=2)", set.up=FALSE)
> setGraph(6, AA.lo)
> plot(Chop.lr, which="fourier(DecYear, 2)cos(k=2)", set.up=FALSE)
> dev.off()

null device
          1
```

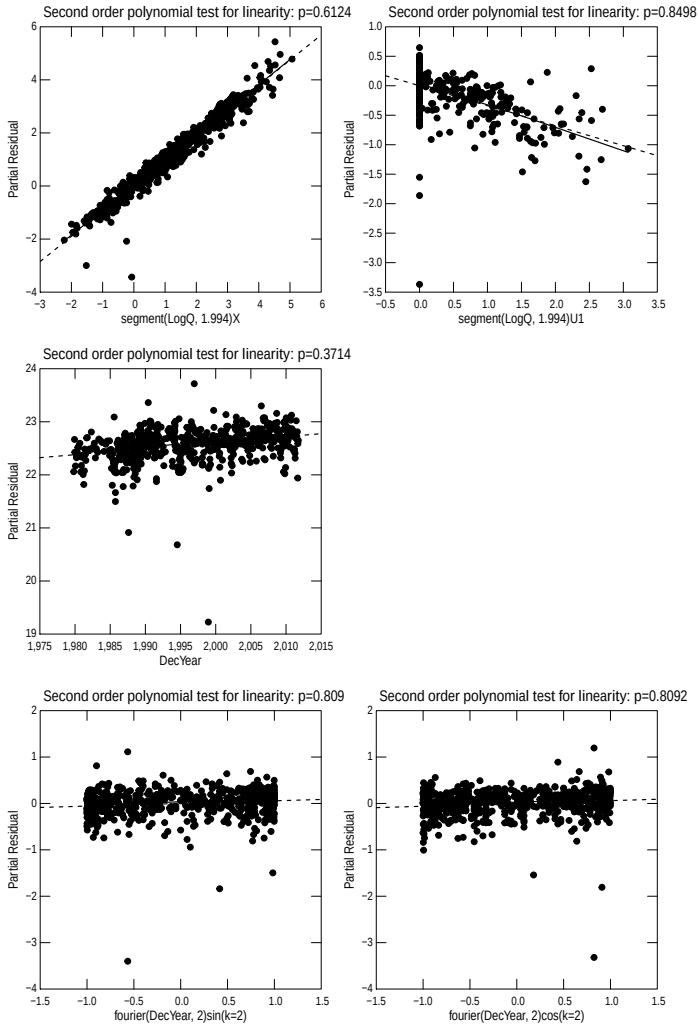

Figure 4. partial residual plots.

3 Further Considerations

To be continued.

References

[1] Hirsch, R.M. and De Cicco, L.A., 2015, User guide to Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval R for hydrologic data (version 2.0, February 2015): U.S. Geological Survey Techniques and Methods book 4, chap A10, 93 p. Available at http://dx.doi.org/10.3133/tm4A10.