

Application 6: Regression Model for Concentration

Dave Lorenz

July 26, 2017

This example illustrates how to use the regression model to estimate concentration rather than load. Concentrations can be estimated for daily or shorter time frames directly from the load regression output. Time-weighted mean concentrations for longer time periods can be computed using the method described for application 6 in Runkel and others (2004), or simply computing the mean daily if confidence intervals are not needed. Flow-weighted mean concentrations can be computed by estimating the flux for each period of time, dividing by the mean flow and dividing by the correct conversion factor; the confidence interval can be computed in the same manner.

This example uses the second revised model from example application 5. The user is directed to that example vignette for details on constructing that model. Part 2 illustrates how to estimate time-weighted mean (TWM) monthly concentrations. Part 3 illustrates how to estimate flow-weighted mean (FWM) concentrations.

```
> # Load the rloadest package and the data
> library(rloadest)
> data(app5.calib)
> head(app5.calib)
```

	DATES	TIMES	FLOW	SC	Alkalinity
1	1995-02-28	1231	10.0	1425.0	248
2	1995-03-24	1301	11.3	1010.0	205
3	1995-03-28	0801	190.0	519.0	78
4	1995-04-12	0931	13.1	784.0	204
5	1995-04-24	1301	22.4	1750.0	231
6	1995-05-08	1301	2700.0	98.9	27

1 Predict Daily Concentrations for 1999

The concentration model is computed using `loadReg`. The default print does not print the concentration model, but it can be printed by setting `load.only` to `FALSE`.

```
> # Create the and print load model with concentration.
> app6.lr <- loadReg(Alkalinity ~ quadratic(log(FLOW)) + log(SC) +
+                   fourier(DATES),
+                   data = app5.calib, subset=DATES < "1998-01-01",
+                   flow = "FLOW", dates = "DATES", conc.units="mg/L",
+                   station="Arkansas River at Halstead, Ks.")
> print(app6.lr, load.only=FALSE)
```

*** Load Estimation ***

Station: Arkansas River at Halstead, Ks.
Constituent: Alkalinity

Number of Observations: 74
Number of Uncensored Observations: 74
Center of Decimal Time: 1996.566
Center of $\ln(Q)$: 5.1571
Period of record: 1995-02-28 to 1997-12-29

Selected Load Model:

Alkalinity ~ quadratic(log(FLOW)) + log(SC) + fourier(DATES)

Model coefficients:

	Estimate	Std. Error	z-score	p-value
(Intercept)	5.95099	0.31973	18.612	0e+00
quadratic(log(FLOW))(5.12598)1	0.91116	0.01875	48.605	0e+00
quadratic(log(FLOW))(5.12598)2	0.02903	0.00516	5.626	0e+00
log(SC)	0.73392	0.04857	15.111	0e+00
fourier(DATES)sin(k=1)	-0.10864	0.02358	-4.608	0e+00
fourier(DATES)cos(k=1)	-0.09143	0.02670	-3.425	6e-04

AMLE Regression Statistics

Residual variance: 0.01496

R-squared: 99.06 percent

G-squared: 345.6 on 5 degrees of freedom

P-value: <0.0001

Prob. Plot Corr. Coeff. (PPCC):

r = 0.9789
 p-value = 0.0172
 Serial Correlation of Residuals: 0.1002

Variance Inflation Factors:

	VIF
quadratic(log(FLOW))(5.12598)1	5.856
quadratic(log(FLOW))(5.12598)2	1.181
log(SC)	7.342
fourier(DATES)sin(k=1)	1.284
fourier(DATES)cos(k=1)	1.441

Comparison of Observed and Estimated Loads

Summary Stats: Loads in kg/d

	Min	25%	50%	75%	90%	95%	Max
Est	5540	10300	16400	66400	169000	283000	865000
Obs	5290	10100	16900	58600	176000	281000	947000

Bias Diagnostics

Bp: -2.08 percent
 PLR: 0.9792
 E: 0.9867

Selected Concentration Model:

Alkalinity ~ quadratic(log(FLOW)) + log(SC) + fourier(DATES)

Model coefficients:

	Estimate	Std. Error	z-score	p-value
(Intercept)	-0.06969	0.31973	-0.218	0.8202
quadratic(log(FLOW))(5.12598)1	-0.08884	0.01875	-4.739	0.0000
quadratic(log(FLOW))(5.12598)2	0.02903	0.00516	5.626	0.0000
log(SC)	0.73392	0.04857	15.111	0.0000
fourier(DATES)sin(k=1)	-0.10864	0.02358	-4.608	0.0000
fourier(DATES)cos(k=1)	-0.09143	0.02670	-3.425	0.0006

AMLE Regression Statistics

Residual variance: 0.01496
 R-squared: 97.17 percent
 G-squared: 263.9 on 5 degrees of freedom
 P-value: <0.0001
 Prob. Plot Corr. Coeff. (PPCC):

```

r = 0.9789
p-value = 0.0172
Serial Correlation of Residuals: 0.1002

```

Comparison of Observed and Estimated Concentrations

```

-----
Summary Stats: Concentrations in mg/L
-----
Min  25% 50% 75% 90% 95% Max
Est 25.8 93.2 221 273 290 295 300
Obs 27.0 98.0 228 268 284 298 310

```

Bias Diagnostics

```

-----
Bp: 0.328 percent
PCR: 1.003
E: 0.9564

```

The `rloadest` package contains the function `predConc` that will estimate concentrations for daily or shorter time periods, depending on the time step. Note that the daily estimation data set, `app5.est`, has 7 missing days and so has only 358 observations instead of 365 for 1999.

```

> # Get the estimation data
> data(app5.est)
> # Predict daily concentrations
> app6.cd <- predConc(app6.lf, app5.est, by="day")
> head(app6.cd)

```

	Date	Flow	Conc	Std.Err	SEP	L95	U95
1	1999-01-01	44	230.4691	8.075796	29.42066	177.3864	294.6353
2	1999-01-02	40	238.0191	8.203423	30.34716	183.2577	304.1994
3	1999-01-03	41	232.3866	7.982030	29.62166	178.9330	296.9834
4	1999-01-04	38	238.2327	8.071336	30.33699	183.4825	304.3841
5	1999-01-05	41	235.7455	8.146778	30.06315	181.4977	301.3075
6	1999-01-06	38	235.2900	7.922827	29.94932	181.2371	300.5938

2 Part 2 Predict Monthly Time-weighted Mean Concentrations for 1999

The estimation of TWM concentrations requires the user to "trick" the regression model by substituting a constant synthetic value for the flow column that actually estimates concentrations in stead of load when using the `predLoad` function. The details of the development of the method are described in Runkel and others (2004). This example application only describes the implementation using `rloadest` functions.

The first step is to create a synthetic flow column in both the calibration and the estimation data sets. The value of the data is the conversion factor from load to concentration, or the reciprocal of the conversion factor from concentration to loads that can be gotten by using the `c2load` function.

```
> # Create synthetic flow values
> app6.calib <- transform(app5.calib, Sflow=1/c2load(1, 1,
+   conc.units="mg/L"))
> app6.est <- transform(app5.est, Sflow=1/c2load(1, 1,
+   conc.units="mg/L"))
> head(app6.calib)
```

	DATES	TIMES	FLOW	SC	Alkalinity	Sflow
1	1995-02-28	1231	10.0	1425.0	248	0.4087324
2	1995-03-24	1301	11.3	1010.0	205	0.4087324
3	1995-03-28	0801	190.0	519.0	78	0.4087324
4	1995-04-12	0931	13.1	784.0	204	0.4087324
5	1995-04-24	1301	22.4	1750.0	231	0.4087324
6	1995-05-08	1301	2700.0	98.9	27	0.4087324

The next step is to construct the calibrated model. This example will use the previously calibrated model, so that diagnostics plots will not be created. The only difference between this model and the previous model is that the flow column is defined as the synthetic flow column. The printed report for this load model should agree with the printed report for the concentration model in the previous section.

```
> # Create the and print load model.
> app6.lrTWM <- loadReg(Alkalinity ~ quadratic(log(FLOW)) + log(SC) +
+   fourier(DATES),
+   data = app6.calib, subset=DATES < "1998-01-01",
+   flow = "Sflow", dates = "DATES", conc.units="mg/L",
+   station="Arkansas River at Halstead, Ks.")
> print(app6.lrTWM)
```

*** Load Estimation ***

Station: Arkansas River at Halstead, Ks.
Constituent: Alkalinity

Number of Observations: 74
Number of Uncensored Observations: 74
Center of Decimal Time: 1996.566
Center of ln(Q): -0.8947
Period of record: 1995-02-28 to 1997-12-29

Selected Load Model:

Alkalinity ~ quadratic(log(FLOW)) + log(SC) + fourier(DATES)

Model coefficients:

	Estimate	Std. Error	z-score	p-value
(Intercept)	-0.06969	0.31973	-0.218	0.8202
quadratic(log(FLOW))(5.12598)1	-0.08884	0.01875	-4.739	0.0000
quadratic(log(FLOW))(5.12598)2	0.02903	0.00516	5.626	0.0000
log(SC)	0.73392	0.04857	15.111	0.0000
fourier(DATES)sin(k=1)	-0.10864	0.02358	-4.608	0.0000
fourier(DATES)cos(k=1)	-0.09143	0.02670	-3.425	0.0006

AMLE Regression Statistics

Residual variance: 0.01496
R-squared: 97.17 percent
G-squared: 263.9 on 5 degrees of freedom
P-value: <0.0001
Prob. Plot Corr. Coeff. (PPCC):
r = 0.9789
p-value = 0.0172
Serial Correlation of Residuals: 0.1002

Variance Inflation Factors:

	VIF
quadratic(log(FLOW))(5.12598)1	5.856
quadratic(log(FLOW))(5.12598)2	1.181
log(SC)	7.342
fourier(DATES)sin(k=1)	1.284
fourier(DATES)cos(k=1)	1.441

Comparison of Observed and Estimated Loads

Summary Stats: Loads in kg/d

```
-----
      Min  25% 50% 75% 90% 95% Max
Est 25.8 93.2 221 273 290 295 300
Obs 27.0 98.0 228 268 284 298 310
```

Bias Diagnostics

```
-----
Bp: 0.328 percent
PLR: 1.003
E: 0.9564
```

The user can now estimate TWM concentrations for periods longer than 1 day. Remember that the estimation dataset has 7 missing days and no estimates for those months will be made.

```
> # Predict monthly TWM concentrations using the \texttt{predLoad} function.
> app6.TWM <- predLoad(app6.lrwTWM, app6.est, by="month")
> # Change the name of the Flux column to Conc
> names(app6.TWM)[3] <- "Conc"
> app6.TWM
```

	Period	Ndays	Conc	Std.Err	SEP	L95	U95
1	January 1999	31	213.1878	7.081182	8.529423	196.95746	230.3869
2	February 1999	28	149.5486	5.216405	6.361388	137.46681	162.3985
3	March 1999	31	194.6095	5.790947	7.210804	180.85773	209.1198
4	April 1999	30	107.1358	3.026241	4.009600	99.49121	115.2064
5	May 1999	31	121.6526	3.345660	4.355846	113.33779	130.4103
6	June 1999	29	NA	NA	NA	NA	NA
7	July 1999	28	NA	NA	NA	NA	NA
8	August 1999	31	168.6656	6.141413	7.344922	154.72644	183.5125
9	September 1999	27	NA	NA	NA	NA	NA
10	October 1999	31	211.5238	6.786674	8.240603	195.83063	228.1284
11	November 1999	30	252.3945	8.046272	9.846216	233.64435	272.2351
12	December 1999	31	189.3642	6.175727	7.502260	175.08417	204.4880

A quick check of the mean daily concentrations verifies that computations. Note that there is no protection against incomplete months when the TWM concentrations are computed manually, so values are obtained for June, July and September.

```
> with(app6.cd, tapply(Conc, month(Date, label=TRUE), mean))
```

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
213.18784	149.54864	194.60951	107.13581	121.65264	106.54503	85.02916	168.66561
Sep	Oct	Nov	Dec				
130.05797	211.52384	252.39450	189.36420				

3 Part 3 Predict Monthly Flow-weighted Mean Concentrations for 1999

The estimation of FWM concentrations is a two-step process. The first step is to build the rating-curve model for loads and estimate the flux desired for each period. The second step is to compute the mean flow for each period and divide the flux by the flow and correct to concentration units. The load model for this example (app6.lr) was created in the first part of this vignette.

```
> # Compute the monthly fluxes.
> app6.FWM <- predLoad(app6.lr, app6.est, by="month")
> # Compute the mean flows
> app6.FWM$Flow <- as.vector(with(app6.est, tapply(FLOW, month(DATES), mean)))
> # Compute the FWM concentration
> app6.FWM <- transform(app6.FWM, FWMC=Flux/Flow/
+                          c2load(1, 1, conc.units="mg/L"))
> app6.FWM
```

	Period	Ndays	Flux	Std.Err	SEP	L95	U95	
1	January	1999	31	35442.25	1309.7176	2155.2335	31405.46	39850.55
2	February	1999	28	63032.12	2511.7167	3783.9322	55940.53	70767.67
3	March	1999	31	25388.38	768.4968	957.4235	23563.38	27315.89
4	April	1999	30	127971.05	4415.6512	6011.0352	116592.86	140150.41
5	May	1999	31	67572.50	1854.0435	2724.5044	62389.33	73067.49
6	June	1999	29	NA	NA	NA	NA	NA
7	July	1999	28	NA	NA	NA	NA	NA
8	August	1999	31	100304.10	3710.5034	5197.4230	90502.12	110870.01
9	September	1999	27	NA	NA	NA	NA	NA
10	October	1999	31	12434.02	397.2618	484.1962	11511.91	13409.65
11	November	1999	30	15924.63	507.6033	621.6939	14740.76	17177.40
12	December	1999	31	23377.49	805.5262	1009.0498	21461.96	25416.63

	Flow	FWMC
1	121.74194	118.99263
2	353.57143	72.86580
3	53.58065	193.67169
4	815.46667	64.14230
5	327.41935	84.35380
6	787.03448	NA
7	932.00000	NA
8	620.48387	66.07348
9	136.25926	NA
10	24.29032	209.22673
11	25.86667	251.63317
12	57.61290	165.85062

For these data, the FWM concentration is less than the TWM concentration. In general, the FWM concentration will be less than the TWM concentration when the concentration and flow are negatively correlated.